



pTabs Help

Release 1.5.4

Stefano Palma

10 March 2016

1	Installazione ed esecuzione	1
1.1	Installazione	1
1.2	Esecuzione	9
2	Aggiornamento e rinnovo	11
2.1	Aggiornamento	11
2.2	Rinnovo	12
2.3	Notifica	12
3	Parametri di configurazione	15
3.1	Parametri	15
3.2	Quartili	32
3.3	Definizione dei formati numerici	33
3.4	Temi	34
3.5	ptabs.yml	35
3.6	File di risorse	35
4	Sintassi	37
5	Scripts	39
5.1	%include	40
6	L'oggetto ptabs	41
6.1	ptabs.new	41
6.2	ptabs.csv	41
6.3	ptabs.inc	42
6.4	Variabili	42
7	Report	43
7.1	inc	43
7.2	Output	43
8	Render	45
8.1	html	45
8.2	pdf	45
8.3	Excel	45
8.4	ptweb	46
8.5	db	47
8.6	Opzioni	48
9	Fonti di dati	51
9.1	Metodi	51
9.2	Fonti di dati	53

10	Hole count	61
10.1	hc	61
10.2	hcm	62
11	Top-line	63
11.1	topline	63
12	Pannelli	65
12.1	panels.add	65
12.2	panels.drop	65
12.3	panel(n).sub	65
12.4	panel(n).add	65
12.5	panel(n).drop	65
12.6	panel(n).rowbase	66
12.7	panels.save	66
12.8	panels.switch	66
12.9	panels.restore	66
12.10	panels.clear	67
12.11	join_col_labels	67
13	Tabelle	69
13.1	autotab	70
13.2	cat	70
13.3	tbox	70
13.4	ord	71
13.5	xord	71
13.6	sord e ords	71
13.7	md	71
13.8	mc	72
13.9	grid	72
13.10	vjoin	73
13.11	hjoin	73
13.12	htjoin	73
13.13	Sottocomandi delle tabelle	74
13.14	base_pct	80
13.15	base_multi	80
13.16	Valori restituiti dalle tabelle	80
14	Net	83
14.1	net_pos	83
14.2	netonly	83
14.3	netstyle	83
14.4	Tabelle di risposte singole	83
14.5	Tabelle di risposte multiple	84
15	Selezione delle osservazioni	89
15.1	filter	89
15.2	select	90
15.3	all	90
15.4	cases	91
15.5	break	91
15.6	period e stratum	92
15.7	base_pos	92
15.8	n	93
15.9	sample	93
15.10	ddup	93
16	Operazioni sui data frame	95
16.1	Workspaces	95

16.2	Merge	95
16.3	Union	96
16.4	Aggregate	96
16.5	Stack	97
16.6	Unstack	98
17	Gestione variabili	101
17.1	rename	101
17.2	label	101
17.3	levels	101
17.4	add_levels	102
17.5	copy_label	102
17.6	copy_levels	102
17.7	lprefix	102
17.8	lsuffix	102
17.9	level_to_label	102
17.10	label_to_level	103
17.11	factor	103
17.12	numeric	104
17.13	string	104
17.14	create	105
17.15	delete	105
17.16	keep	105
17.17	copy	105
17.18	missing	105
17.19	drop_levels	105
17.20	na_if	105
17.21	value	106
17.22	value_if	106
17.23	na_to_level	106
17.24	recode	106
17.25	interaction	107
17.26	expand	107
17.27	compress	107
17.28	mc_to_md	108
17.29	md_to_panel	108
17.30	reorder	108
17.31	switch	108
17.32	move_before	109
17.33	move_after	109
17.34	if_any	109
17.35	if_all	109
17.36	if_none	109
17.37	if_only	110
17.38	split_vars	110
17.39	derotate	110
17.40	fill	111
17.41	seed	112
18	Espressioni	113
18.1	parse, parse!, parse?	114
19	Gruppi di variabili	117
19.1	Metodi dell'oggetto sets	117
19.2	Metodi degli oggetti set	118
19.3	Esempi	118
20	Date e tempo	121
20.1	chron	121

20.2	spss.time e spss.date	122
21	Funzioni di utilità	123
21.1	s	123
21.2	ss	123
21.3	rep	124
21.4	rev	125
21.5	ids	125
21.6	in_groups_of	125
21.7	pr	125
21.8	common	125
21.9	log_width	125
21.10	next	126
21.11	stop! / start!	126
21.12	r.exec o r <<	126
21.13	r.source	126
21.14	\$p	126
21.15	css	127
21.16	debug	127
21.17	Scrittura e lettura di informazioni	127
22	Funzioni informative	129
22.1	Variabili	129
22.2	Etichette e livelli	130
22.3	Dati	132
22.4	Data e ora	136
22.5	Funzioni per i titoli delle tabelle	136
23	Ponderazione	139
23.1	weight	139
23.2	wby	140
23.3	weight.action	140
23.4	rake	141
24	Test statistici e accentuazioni	145
24.1	Chi-quadrato	145
24.2	Pairwise Comparisons: medie	145
24.3	Pairwise Comparisons: proporzioni	146
24.4	Accentuazioni	147
25	Analisi delle corrispondenze	149
26	Analisi dei valori mancanti	155
26.1	show_missing	155
27	Grafici	157
27.1	Tipi di grafici	158
27.2	Opzioni	159
27.3	palette	163
27.4	colors	163
27.5	styles	164
27.6	Opzioni grafico gauge	164
28	Mappe	167
28.1	Layers e overlays	169
28.2	Markers	170
28.3	Coordinate esterne	170
28.4	Linee e forme	171

29 Immagini	173
29.1 img	173
29.2 img_foot	173
29.3 image	173
30 Testo HTML	175
30.1 doc	175
31 Finestre di dialogo	177
31.1 Messaggi	177
31.2 Input utente	178
32 Funzioni definite dall'utente	181
32.1 Funzioni e variabili interne al report	181
32.2 Funzioni e variabili globali	182
32.3 Callbacks	185
33 ptGen	191
33.1 Questionario Odin	191
33.2 File di dati SPSS	192
33.3 File Triple-S	192
33.4 Configurazione	192
33.5 Total Commander, Unreal Commander	193
34 pTabsDB	195
34.1 Selezione dei dati	196
34.2 PowerPoint	197
34.3 Grafici	197
34.4 Tabelle	198
34.5 Aggiornamento	198
34.6 Altre funzioni	199
35 pTabsWeb	201
35.1 Screenshot	202
36 Utilità da linea di comando	203
36.1 dft2pt	203
36.2 cleanQ	203
37 Files	205
37.1 File di configurazione di pTabs	205
37.2 Fogli stile	205
37.3 Header e footer PDF	205
37.4 Installazione librerie R	205
38 Risoluzione dei problemi	207
38.1 pTabs	207
38.2 ptKey	210
38.3 SciTE	210
38.4 Ottenere assistenza	211
39 Version History	213
39.1 Beta	213
39.2 Ultima	213
39.3 Precedenti	214

Installazione ed esecuzione

Il programma di installazione `pTabs_setup.#.#.#.#.exe` installa i programmi:

- **pTabs**
- **ptKey**: programma di gestione delle licenze
- **ptGen**: utility per la generazione di script *pTabs*

Per il completo funzionamento di *pTabs* è necessario installare anche i seguenti programmi:

- **R** (<http://www.r-project.org/>) e alcune sue librerie
- **wkhtmltopdf** (<http://wkhtmltopdf.org/downloads.html>)
- **Java Runtime Environment (JRE)** ¹ (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
- **Java Development Kit (JDK)** ² (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)

Suggerimento: Non installare direttamente i programmi, ma seguire la procedura descritta nella sezione seguente

1.1 Installazione

Di seguito viene illustrato un esempio completo di installazione passo per passo.

E' possibile scegliere qualsiasi cartella di installazione, ma è necessario attenersi a queste regole:

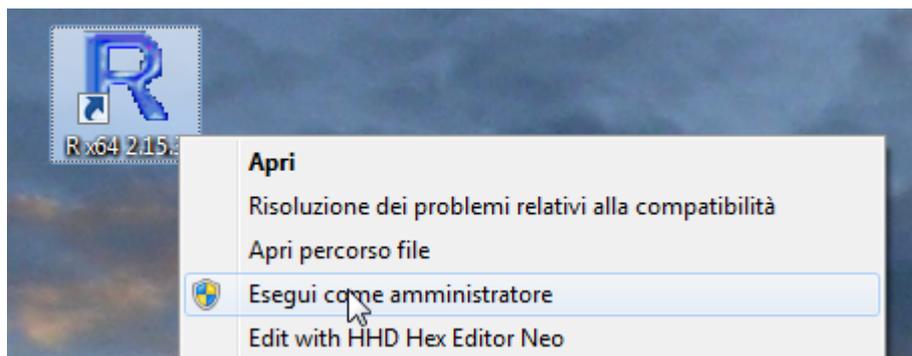
- non installare *pTabs* direttamente nella radice di un disco
- non utilizzare un percorso che comprenda degli spazi nei nomi di cartelle

Nell'esempio **pTabs** viene installato nella cartella `C:\stat\ptabs` (lo stesso percorso proposto dal programma di installazione).

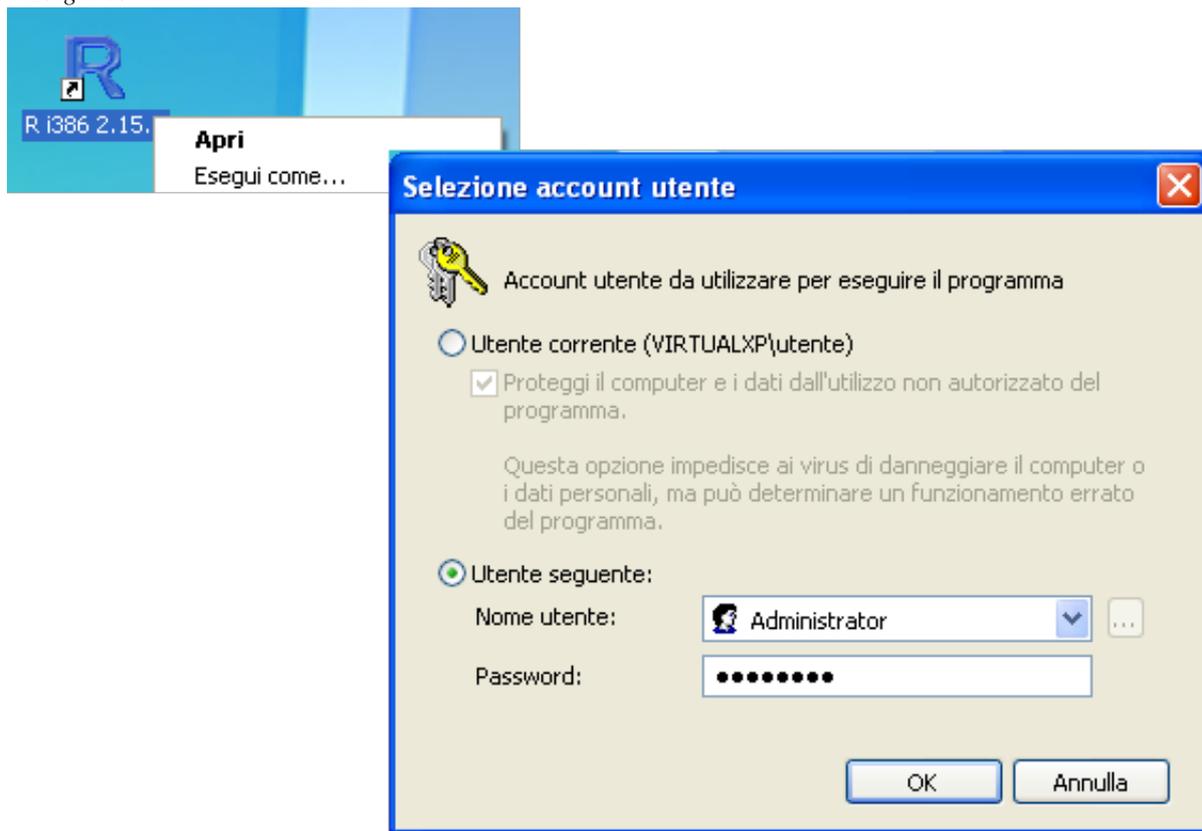
Nelle istruzioni seguenti, *eseguire/avviare come amministratore* significa cliccare con il pulsante destro del mouse sul file/icona del file e scegliere *Esegui come amministratore*

¹ Il *Java Runtime Environment (JRE)* è usato da alcune librerie **R**. Bisogna installare la versione corretta (32 o 64 bit) per la propria piattaforma.

² Il *Java Development Kit (JDK)* è utilizzato dalla libreria **Rjb** (Ruby-Java bridge). Bisogna installare la versione a 32 bit anche per i sistemi a 64 bit. Se non si utilizzano i report in formato Excel 97-2003 (xls) è possibile non installare il JDK.



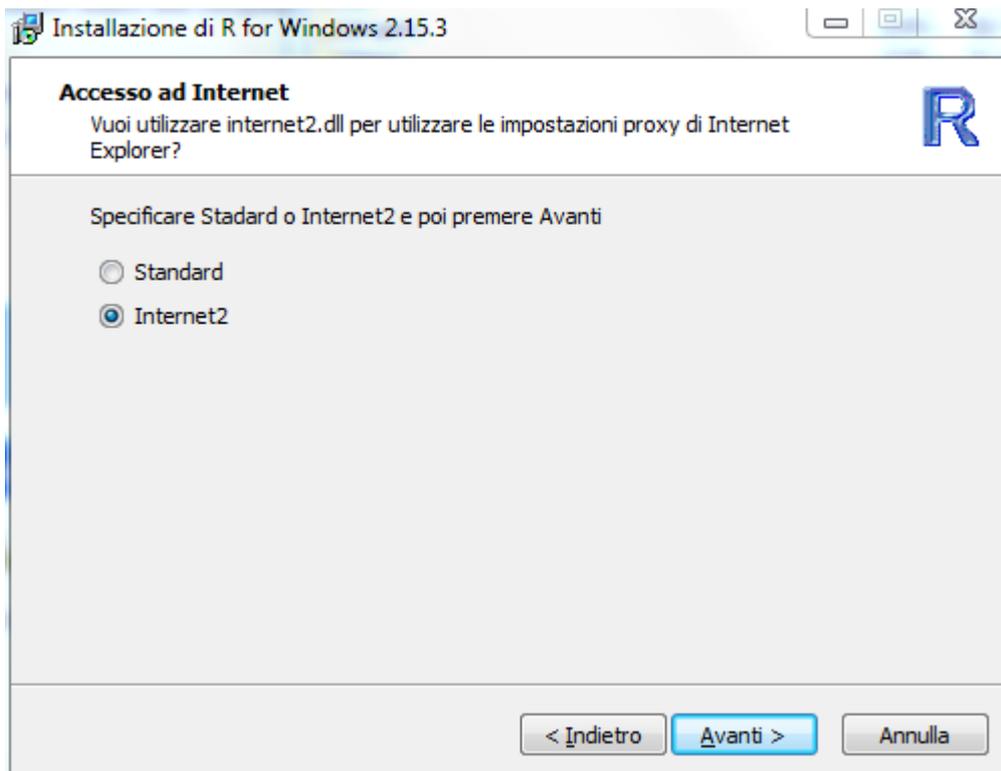
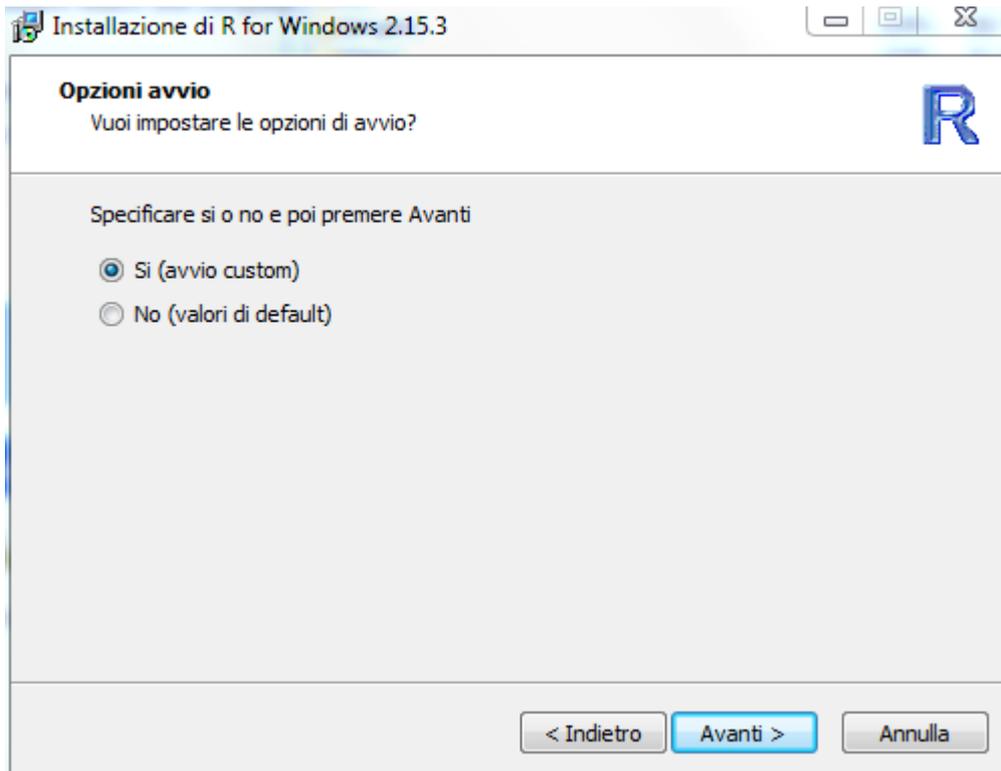
o Esegui come ...



1.1.1 1) R

- Scaricare da <http://cran.r-project.org/> il programma di installazione di R (è raccomandata la versione 2.15.3)
- Eseguire come amministratore il programma di installazione, installando la versione a 64-bit di R se il sistema operativo è a 64-bit oppure la versione a 32-bit di R se il sistema è a 32-bit. (Se il sistema è a 64-bit è possibile comunque installare *in aggiunta* anche la versione a 32-bit).

Suggerimento: Se si utilizza un proxy, scegliere **avvio custom** e più avanti **Internet2**.



- Finita l'installazione, avviare, sempre come amministratore, R
- Installare le librerie aggiuntive copiando e incollando nella console il codice seguente (assicurandosi di eseguire anche l'ultima riga):

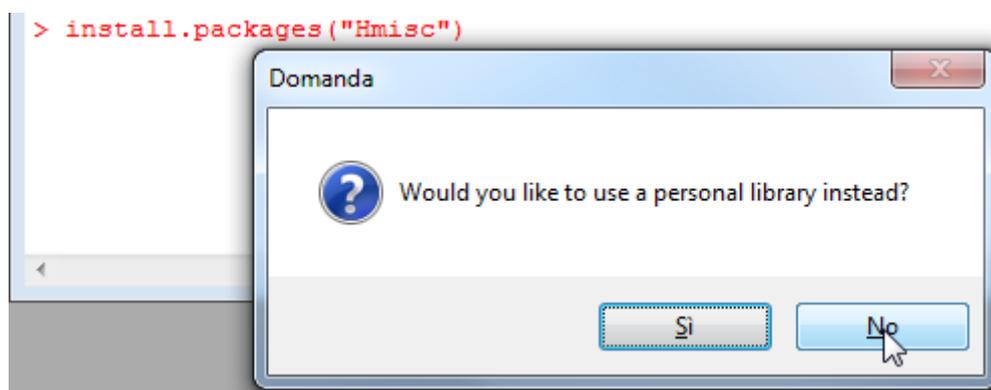
```
1 install.packages("Hmisc")
2 install.packages("car")
3 install.packages("gdata")
```

```

4 install.packages("RODBC")
5 install.packages("mice")
6 install.packages("VIM")
7 install.packages("survey")
8 install.packages("XLConnect")
9 install.packages("reshape")
10 install.packages("DBI")
11 install.packages("gsubfn")
12 install.packages("RSQLite")
13 install.packages("RSQLite.extfuns")
14 install.packages("sqldf")
15 install.packages("sss")
16 install.packages("SASxport")
17 install.packages("ffbase")
18 install.packages("ca")
19 install.packages("RColorBrewer")
20 install.packages("Rcpp")
21 source("http://bioconductor.org/biocLite.R")
22 biocLite("EBImage")

```

- Se R chiede di usare una cartella personale per le librerie, rispondere **No** in modo da utilizzare la sottocartella `library` di R.



Se la procedura restituisce un errore simile al seguente, significa che l'utente con cui è stato avviato R non ha diritti di scrittura nella cartella `library` di R.

```

> install.packages("Hmisc")
Warning in install.packages("Hmisc") :
  'lib = "C:/Program Files/R/R-2.15.3/library"' is not writable
Errore in install.packages("Hmisc") : unable to install packages
> |

```

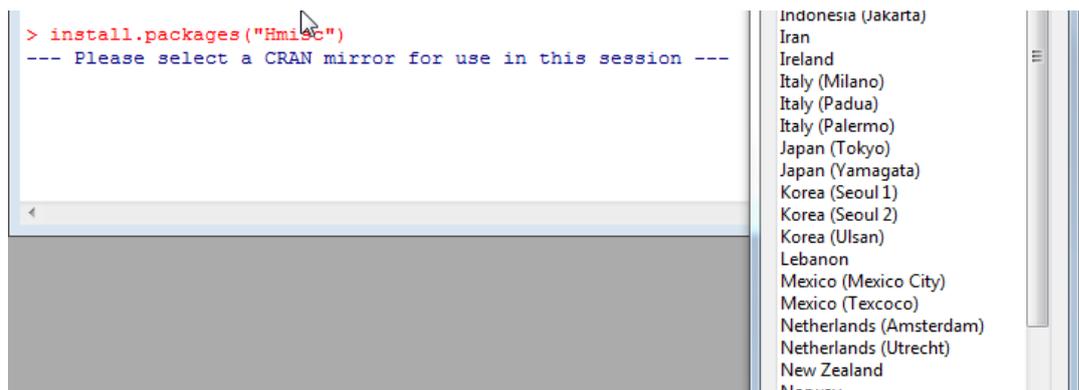
E' anche possibile verificare qual è la cartella delle librerie con l'istruzione `.libPaths()`

```

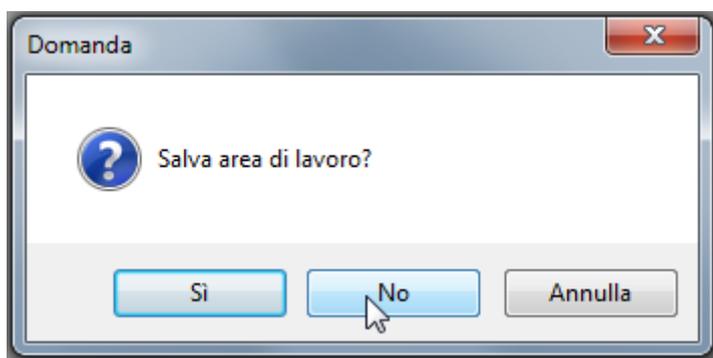
> .libPaths()
[1] "C:/Program Files/R/R-2.15.3/library"
> |

```

- Alla successiva richiesta, scegliere il mirror (per es. "Switzerland", "Italy (Milano)", ...).



- Verificare che non ci siano messaggi di errore. Nel caso rieseguire le righe interessate.
- Uscire da R senza salvare la sessione.



Nel caso siano installate più versioni di R, è possibile indicare la versione da utilizzare specificando la cartella che contiene l'eseguibile (Rterm.exe) nelle variabili d'ambiente R_HOME oppure PTABS_R_HOME.

1.1.2 2) wkhtmltopdf

- Scaricare da <http://wkhtmltopdf.org/downloads.html> il programma di installazione wkhtmltox-x.xx.x-installer.exe (versione 0.12 o superiore)
- Eseguire il programma di installazione come amministratore
- Confermare il percorso di installazione suggerito

1.1.3 3) Java

- Creare la cartella C:\stat\java
- Scaricare due tra i seguenti file³ da https://preview.cubby.com/pl/java/_ac591df3f3fa456d8e0e49df0b985ce6:
 - jre6.zip (se il sistema è a 32bit)
 - jre7.zip (se il sistema è a 64bit)
 - java.zip (per entrambi i sistemi)
- Decomprimere i due file scaricati, preservando le cartelle, in C:\stat\java in modo che si avranno le cartelle C:\stat\java\jre e C:\stat\java\jre6 per i sistemi a 32bit e C:\stat\java\jre e C:\stat\java\jre7 per i sistemi a 64bit

La struttura delle cartelle sarà simile a questa:

³ Gli archivi java vengono forniti per semplificare il processo di installazione. Se si preferisce, scaricare e installare il *Java Development Kit (JDK)* e il *Java Runtime Environment (JRE)* (vedi note precedenti).



1.1.4 4) pTabs

- Scaricare dal link fornito il programma di installazione pTabs_setup.###.exe
- Eseguire come amministratore il programma di installazione
 - Indicare come cartella d'installazione C:\stat\ptabs
 - Mettere la spunta ad *Aggiunge la cartella dell'applicazione al path di sistema*
 - Proseguire fino alla fine dell'installazione
- Aggiungere al file C:\stat\ptabs\user\ptabs.yml le seguenti righe (lasciando la prima riga ---):
 - se il sistema è a 32bit:

```
:jsdk_home_32: c:/stat/java  
:jre_home_32: c:/stat/java/jre6
```
 - se il sistema è a 64bit:

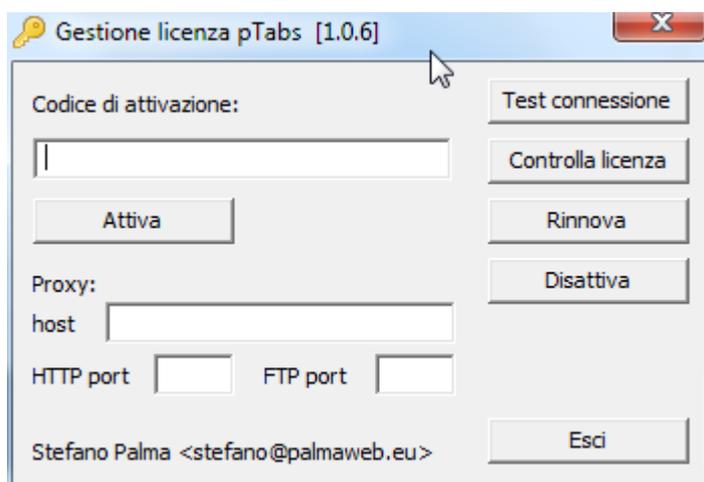
```
:jsdk_home_32: c:/stat/java  
:jre_home_64: c:/stat/java/jre7
```
- **Se** si è installato *wkhtmltopdf* in un percorso diverso da quello standard, aggiungere anche questa riga:

```
:wkhtmltopdf: M:/y/path/to/wkhtmltopdf/wkhtmltopdf.exe
```

dove al posto di M:/y/path/to/wkhtmltopdf/wkhtmltopdf.exe bisogna indicare il percorso in cui è installato l'eseguibile *wkhtmltopdf.exe*.

1.1.5 5) Attivazione

Eseguire il programma **ptKey**.



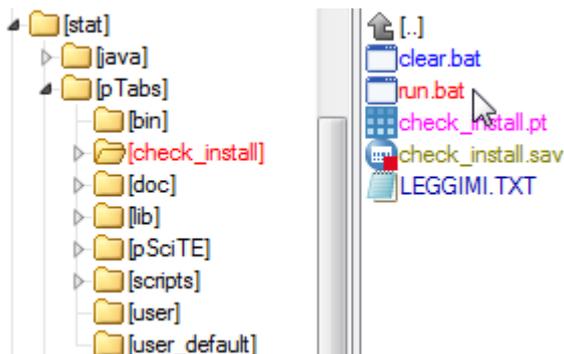
Controllare la connessione con il pulsante **Test connessione**, inserire il **codice di attivazione fornito** nel box “Codice di attivazione:” e premere **Attiva**. E’ necessaria una connessione Internet attiva che consenta una connessione **FTP**.

Suggerimento: Se il **firewall** aziendale impedisce le connessioni **FTP** connettersi temporaneamente a Internet tramite una **Internet Key**

<p>Attenzione: Qualora si desiderasse disinstallare pTabs, disattivare prima la licenza con il pulsante Disattiva</p>
--

1.1.6 6) Verifica installazione

Per verificare la corretta installazione di *pTabs*, posizionarsi nella cartella `C:\stat\ptabs\check_install` ed eseguire il batch file `run.bat`.



Controllare che non compaiano messaggi di errore e verificare che siano stati prodotti i vari output.

Il file `LEGGIMI.TXT` contiene l'elenco dei file che devono essere generati dal test.

Suggerimento: In caso di **errori** o **problemi**, consultare la sezione *Risoluzione dei problemi*

1.1.7 7) Personalizzazioni utente

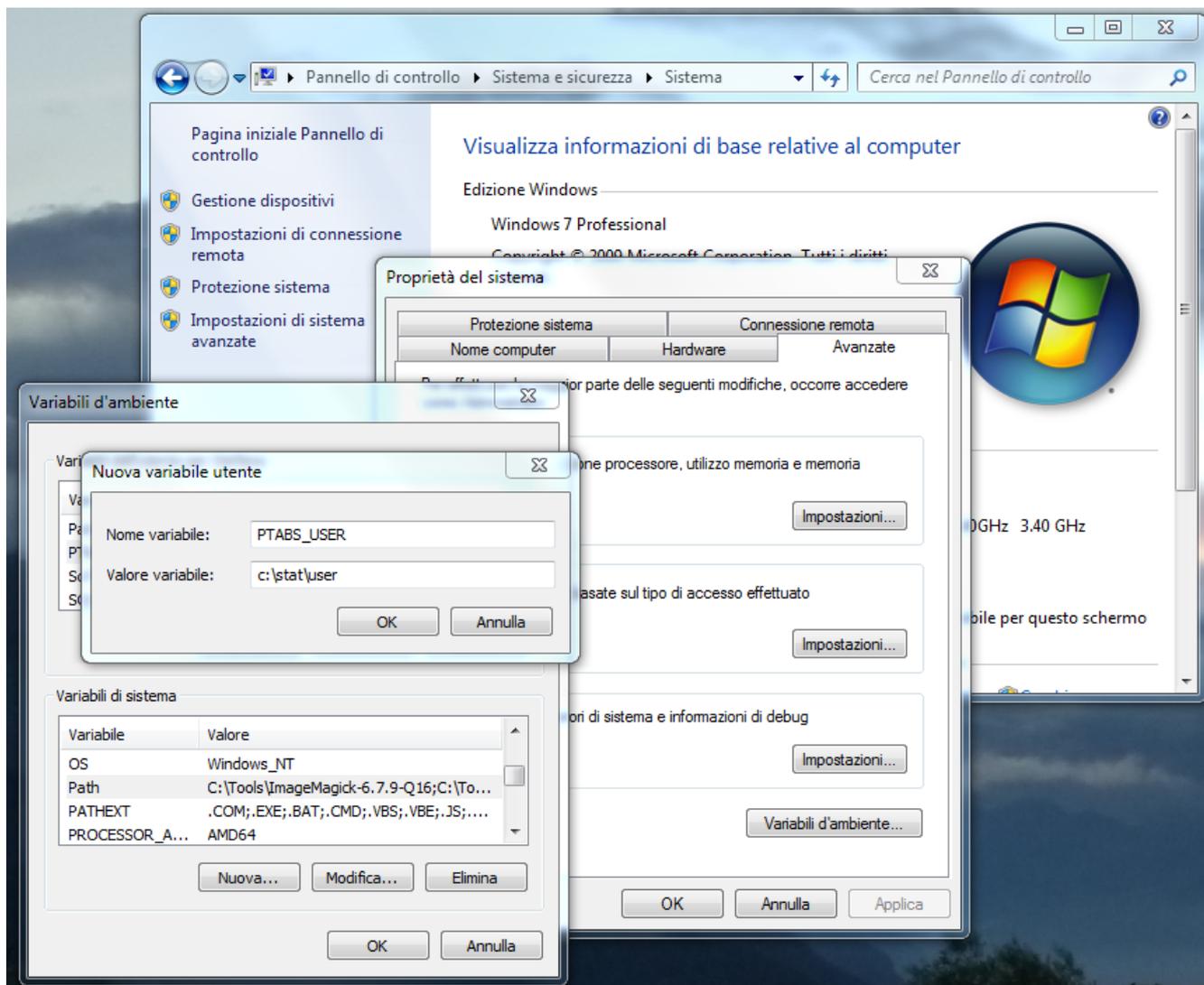
Suggerimento: Questa procedura è facoltativa e non è indispensabile per il funzionamento di *pTabs*, ma rende più pratica la gestione degli aggiornamenti

La cartella `user` all'interno del percorso di installazione di *pTabs* contiene vari file personalizzabili dall'utente.

Questa cartella non verrà sovrascritta durante gli aggiornamenti di *pTabs*, ma per una maggiore praticità⁴ si consiglia di copiare questa cartella fuori dal percorso di installazione.

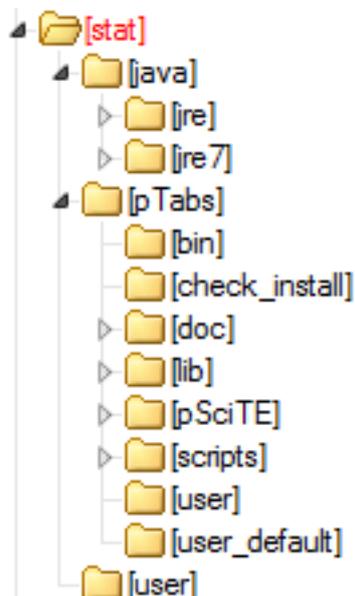
- Copiare il contenuto della cartella `C:\stat\ptabs\user` nella cartella `C:\stat\user`
- Nel *Pannello di controllo* → *Sistema Impostazioni di sistema avanzate* → *Variabili d'ambiente* inserire una nuova variabile d'ambiente:
 - nome variabile: `PTABS_USER`
 - valore variabile: `C:\stat\user`

⁴ Prima di installare una nuova versione di *pTabs* è possibile rinominare la cartella `pTabs` in modo da conservare la vecchia installazione ed eventualmente riattivarla semplicemente scambiando i nomi delle cartelle. Con la cartella `user` fuori dal percorso di installazione di *pTabs*, non è necessario preoccuparsi di sovrascrivere manualmente le proprie configurazioni.



In tal caso la cartella `C:\stat\ptabs\user` non è più attiva e bisogna utilizzare al suo posto quella definita in `PTABS_USER`.

La struttura finale delle cartelle sarà pertanto simile a questa:



La cartella `user_default` contiene la copia di default dei vari file personalizzabili dall'utente. Verrà sovrascritta ad ogni installazione con eventuali versioni aggiornate. I file contenuti in questa cartella non vengono utilizzati durante l'esecuzione di pTabs.

Attenzione: Impostare la variabile `PTABS_USER` su `C:\stat\ptabs\user` è inutile perché è già il percorso di default

1.2 Esecuzione

1.2.1 Da linea di comando

pTabs può essere eseguito da linea di comando passando il nome dello script come parametro:

```
C:\>ptabs scriptName
```

```
+ ===== pTabs [x.x.x (xxx)] ===== +
+           Palma Tabulation Program           +
+           -----                           +
+           Stefano Palma <info@ptabs.eu>      +
+ =====
```

```
Copy registered to: xxxxxx xxx
Expiration date: ####-##-##
Now: 2014-03-06 10.50.12
```

```
--- R version 2.15.3 (2013-03-01)
--- Platform: x86_64-w64-mingw32/x64 (64-bit)
--- System memory limit: 16351 Mb
--- Used memory: 23.25 Mb
```

```
... loading d:/stat/user/user_functions.rb
```

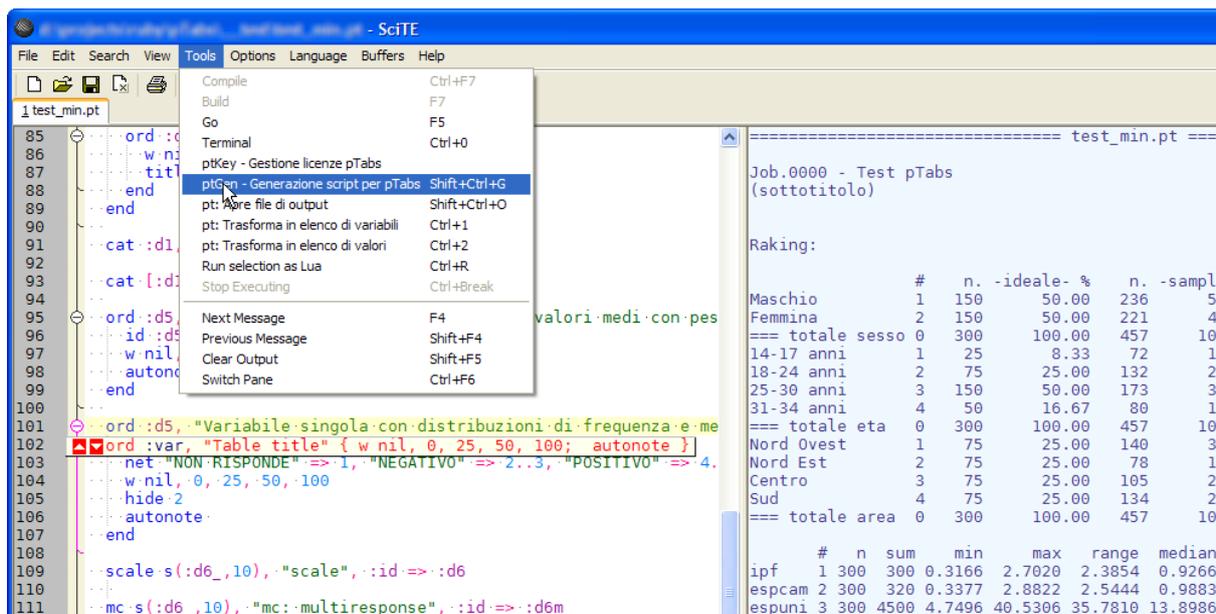
```
... loading ./resources/ptabs.yml
... loading d:/stat/user/ptabs.yml
```

```
--- Characters encoding: iso-8859-15
--- Java SE: d:/stat/java/jre7
--- Java SDK: d:/stat/java
--- xms: 64m
--- mxm: 640m
--- wkhtmltopdf: {pf32}/wkhtmltopdf/wkhtmltopdf.exe
--- User home folder: C:/Users/Stefano
--- User function and settings folder (PTABS_USER): d:/stat/user
--- Additional paths:
--- pTabs folder: d:/stat/pTabs
--- Working folder: d:/stat/pTabs/check_install
--- Script name: check_install.pt
```

```
... executing script "scriptName.pt"...
...
```

1.2.2 Con l'editor SciTE

L'installazione di *pTabs* include una versione di [SciTE](#) configurata per lavorare con *pTabs*.



Include:

- evidenziazione della sintassi
- suggerimenti per la sintassi (calltips)
- abbreviazioni
- code folding
- evidenziazione parentesi
- esecuzione dello script (F5) e cattura dell'output
- esecuzione delle sole tabelle selezionate (ALT+F5)
- esecuzione di ptKey e ptGen direttamente da SciTE
- menu con funzionalità aggiuntive
- help (F1)

SciTE command cheat sheet

Note

Aggiornamento e rinnovo

2.1 Aggiornamento

2.1.1 pTabs

Con il programma di installazione

Per aggiornare *pTabs* scaricare ed eseguire il **programma di installazione** `pTabs_setup.###.exe`.

Verrà riconosciuto il percorso scelto durante la prima installazione, pertanto normalmente si tratta di dare conferma alle scelte precedentemente effettuate. Non è necessario spuntare “*Aggiunge la cartella dell’applicazione al path di sistema*” in quanto sarà già presente nella configurazione del computer.

Con l’archivio zip

Come alternativa rispetto al programma eseguibile, è possibile scaricare **l’archivio zip** `pTabs_setup.###.zip`.

In questo caso l’utente deve aver cura di scompattare l’archivio nel corretto percorso di installazione preservando la struttura delle cartelle.

Suggerimento: L’archivio zip può essere utilizzato nel caso non si disponga di sufficienti diritti per l’esecuzione del programma di installazione.

2.1.2 Librerie R

Talvolta nella comunicazione di una nuova versione e nella pagina *Version History* viene indicato di installare delle nuove **librerie R**. In tal caso avviare **come amministratore**¹ *R* e installare le librerie aggiuntive copiando e incollando nella console il codice indicato (avendo cura di eseguire anche l’ultima riga). Per esempio:

```
1 install.packages("XLConnect")
2 install.packages("reshape")
```

2.1.3 Salvare l’installazione precedente

Se durante la configurazione di *pTabs* era stata configurata una **cartella dell’utente** esterna alla cartella di *pTabs*:

```
C:\stat\pTabs
C:\stat\user
```

¹ Per eseguire come amministratore un programma, cliccare con il pulsante destro del mouse sul file/icona del file e scegliere *Esegui come amministratore*.

è possibile, prima di installare una nuova versione, rinominare la cartella pTabs e poi procedere con l'aggiornamento:

```
mv C:\stat\pTabs C:\stat\pTabs_v1.2.3
```

In questo modo verrà conservata la vecchia installazione e, in caso di problemi, è possibile ritornare alla vecchia versione semplicemente rinominando le cartelle.

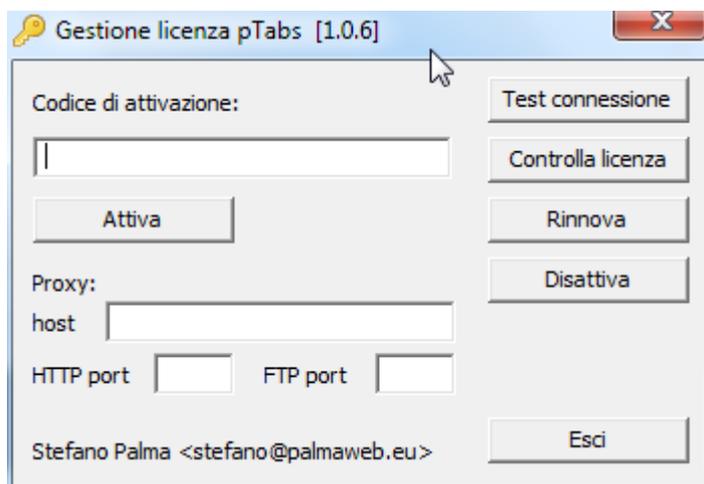
```
mv C:\stat\pTabs C:\stat\pTabs_v1.2.4
mv C:\stat\pTabs_v1.2.3 C:\stat\pTabs
```

Nel caso non si sia provveduto a configurare una cartella dell'utente esterna alla cartella di pTabs, la procedura è un po' più complessa, in quanto è necessario ricordarsi di copiare i vecchi file della cartella pTabs\user nella nuova installazione.

```
copy C:\stat\pTabs_v1.2.3\user\*.* C:\stat\pTabs\user
```

2.2 Rinnovo

Per rinnovare la licenza di *pTabs* eseguire il programma **ptKey** e cliccare sul pulsante Rinnova.



2.3 Notifica

Dalla versione 1.2.3 durante l'esecuzione, *pTabs* controlla la presenza di nuove versioni e stampa un'avviso nella finestra di output:

```
! pTabs NEW version 1.2.3 (105) available
```

Con eventuali istruzioni:

```
+ Download: https://
+ Attenzione: per installare la nuova versione e' necessario installare nuove librerie R!
+ Per dettagli e istruzioni consultare la pagina: http://help.ptabs.eu/versions.html
```

Vengono notificate anche eventuali versioni BETA.

```
- pTabs BETA version 1.3.0 (107) available
```

E' possibile disabilitare il controllo delle nuove versioni o modificarne la frequenza tramite il parametro di configurazione `check_new_version_hh`.

Suggerimento: Si consiglia di installare sempre le nuove release e di installare le versioni BETA solo quando contengono qualche nuova caratteristica che si desidera utilizzare.

Note

Parametri di configurazione

La configurazione di *pTabs* è controllata da una serie di parametri. Tutti i parametri hanno già un valore di default che può essere sufficiente per i casi più comuni. E' possibile modificare i file di configurazione oltre che modificare i vari parametri all'interno dello script.

La modifica dei parametri relativi al calcolo e al layout delle tabelle hanno effetto solo sulle tabelle successive.

I parametri logici (`true|false`), vengono attivati e disattivati negli script con la sintassi: `parametro!` e `parametro_`. Per i restanti parametri, si passa l'argomento dopo il nome del parametro.

`parametro?` stampa e restituisce il valore correntemente impostato del parametro.

```

1 rowpct!           # attiva la visualizzazione delle percentuali di riga
2 sd_              # disattiva la visualizzazione della deviazione standard
3 frm_mean {:p=>3} # imposta tre decimali nel formato delle medie
4 net_pos?        # stampa nel log il valore del parametro 'net_pos'
```

3.1 Parametri

3.1.1 Configurazione

Parametro	Tipo	Default	Descrizione
<code>add_load_path</code>	Array	<code>[]</code>	Vettore con percorsi aggiuntivi di ricerca dei file di risorse
<code>check_new_version</code>	num	<code>30</code>	Numero di ore di intervallo per controllare la presenza di una nuova versione di <i>pTabs</i> . 0 per non effettuare mai il controllo

3.1.2 Nomi variabili

Parametro	Tipo	Default	Descrizione
<code>dot_to_underscore</code>	Boolean	<code>false</code>	Converte il punto (.) nei nomi di variabile in underscore (_)
<code>underscore_to_dot</code>	Boolean	<code>false</code>	Converte l'underscore (_) nei nomi di variabile in punto (.)
<code>vars_to_lower_case</code>	Boolean	<code>true</code>	Converte in minuscolo i nomi di variabile

3.1.3 Pannelli

Parametro	Tipo	De- fault	Descrizione
join_col_labels	Boolean	true	Unisce le colonne sotto un'unica etichetta quando variabili consecutive hanno la stessa variable label
single_panel	Boolean	false	Unisce i pannelli in un pannello unico
weights_vars_action	Symbol	:message	Comportamento in caso di non corrispondenza dei pesi: :error, :message, :none

3.1.4 Fiancate

Parametro	Tipo	De- fault	Descrizione
join_row_labels	Boolean	true	Inserisce una sola volta l'etichetta della variabile quando variabili consecutive hanno la stessa variable label
varlabel	Boolean	false	Stampa le etichette delle variabili
total_after	Boolean	false	Totale in fondo alla tabella
totals	Symbol	:all	Totali di riga per tutte le variabili o una volta sola per tabella: :all, :one
rowlevel	String	nil	Stampa solo righe di livello minore o uguale (1 = righe senza rientro; 2 = righe con un rientro, ecc.)
scale_style	Symbol	:compact	Stile tabelle scale: :compact, :long

3.1.5 Net

Parametro	Tipo	Default	Descrizione
net_pos	Symbol	:before	Posizione net: :before, :after, :top, :bottom
netonly	Boolean	false	Stampa solo i net
netstyle	Boolean	true	Distingue graficamente i net dagli altri item

3.1.6 Basi

Parametro	Tipo	Default	Descrizione
base_pct	Symbol	:table	Metodo di calcolo delle percentuali di colonna: :table, :all

3.1.7 Set di variabili

Parametro	Tipo	Default	Descrizione
base_multi	Symbol	:cases	Metodo di calcolo delle percentuali di colonna nelle multiple: :cases, :resp, :all
md_true_value	Fixnum	1	Valore da contare nelle multidicotomy ("citato"). Vedere anche md_ft e md_tf
md_false_value	Fixnum	0	Valore "non citato" nelle multidicotomy. Vedere anche md_ft e md_tf
sets_display	Boolean	false	Set di variabili: stampa set trovati
sets_all	Boolean	false	Set di variabili: cerca tutti i set di variabili, non solo i set di multiple
sets_start	String	"[a-zA-Z]"	Set di variabili: caratteri di inizio (espressione regolare)
sets_sep	String	"[_.]"	Set di variabili: caratteri separatori (espressione regolare)
sets_pref	Array	["m_", "s_"]	Set di variabili: prefissi per i nomi di set: multiple e singole

3.1.8 Pesì

Parametro	Tipo	Default	Descrizione
weight_legend_label	String	"Pesi:"	Testo iniziale della legenda dei pesi
nil_weight_legend	String	"n.a"	Simbolo inserito al posto di un peso nil nella legenda in fondo alla tavola
nil_weight_label	String	"-"	Simbolo inserito al posto di un peso nil nelle etichette

3.1.9 Zero e basse numerosita'

Parametro	Tipo	Default	Descrizione
zero	String	"-"	Carattere che sostituisce lo zero nelle celle
zero-row	Boolean	true	Mostra le righe della tabella con zero casi
empty_table	Symbol	:message	Comportamento per tabelle con 0 casi: :print, :message, :none
empty_table_msg	String	"La tabella e' vuota"	Messaggio per le tabelle con 0 casi
low_values	Symbol	:weight	Considera il numero di casi ponderati (:weight) o grezzi (:raw)
low_table	Symbol	:print	Comportamento per tabelle con numero basso di casi: :print, :message, :none
low_table_msg	String	"La tabella non contiene un numero significativo di casi"	Messaggio per le tabelle con numero basso di casi
low_table_count	Count	3	Numero di casi per considerare una tabella come tabella con numero basso di casi
low_cell	Symbol	:print	Comportamento per celle con numero basso di casi: print, :hide
low_cell_label	String	"***"	Etichetta per celle con numero basso di casi
low_cell_count	Count	3	Numero di casi per considerare una cella come cella con numero basso di casi

3.1.10 Statistiche

Parametro	Tipo	Default	Descrizione
rtotal	Boolean	true	Totale di riga
tcount_u	Boolean	false	Totale valori assoluti non ponderati
tcount	Boolean	true	Totale valori assoluti
tcopct	Boolean	false	Totale percentuali riga
trowpct	Boolean	true	Totale percentuali colonna
count_u	Boolean	false	Valori assoluti non ponderati
count	Boolean	true	Valori assoluti
rowpct	Boolean	false	Percentuali riga
colpct	Boolean	true	Percentuali colonna
base_u	Boolean	false	Base non pesata
base	Boolean	true	Base
mean	Boolean	true	Media
sd	Boolean	true	Deviazione standard
se	Boolean	false	Errore standard della media
median	Boolean	false	Mediana
qrt	Boolean	false	Quartili
q0	Fixnum	0	Quartile 0
q1	Float	0.25	Quartile 1
q2	Float	0.5	Quartile 2
q3	Float	0.75	Quartile 3
q4	Fixnum	1	Quartile 4
iqr	Boolean	false	Scarto interquartile
range	Boolean	false	Range, campo di variazione
sum	Boolean	false	Somma
mresp	Boolean	false	Numero medio risposte multiple
nresp	Boolean	false	Numero totale risposte multiple
i_conc	Boolean	false	Indice concentrazione

3.1.11 Etichette

Parametro	Tipo	Default	Descrizione
title_template	String	nil	Template per titolo tabella
title_prefix	String	" "	Prefisso per titolo tabella
label_rtotal	String	"Totale"	Etichetta totale di riga
label_ctotal	String	"TOTALE"	Etichetta totale di colonna
label_campione	String	"CAMPIONE"	Etichetta tabella campione
label_freq_count	String	"N. "	Etichetta valori assoluti frequenze
label_freq_pct	String	"%"	Etichetta percentuali frequenze
label_count_u	String	"v.a (no pond.) "	Etichetta valori assoluti non ponderati
label_count	String	"v.a. "	Etichetta valori assoluti
label_rowpct	String	"% row"	Etichetta percentuali riga
label_colpct	String	"% col"	Etichetta percentuali colonna
label_base_u	String	"base (no pond.) "	Etichetta base non ponderata
label_base	String	"base"	Etichetta base
label_mean	String	"media"	Etichetta media
label_sd	String	"dev. standard"	Etichetta deviazione standard
label_se	String	"s.e. media"	Etichetta errore standard della media
label_median	String	"mediana"	Etichetta mediana
label_q0	String	"minimo"	Etichetta quartile 0
label_q1	String	"1^ quartile"	Etichetta quartile 1
label_q2	String	"mediana"	Etichetta quartile 2
label_q3	String	"3^ quartile"	Etichetta quartile 3

Contin

Tabella 3.1 – continua dalla pagina precedente

Parametro	Tipo	Default	Descrizione
label_q4	String	"massimo"	Etichetta quartile 4
label_iqr	String	"scarto interquartile"	Etichetta scarto interquartile
label_range	String	"range"	Etichetta range
label_sum	String	"somma"	Etichetta somma
label_mresp	String	"N. medio risposte"	Etichetta numero totale risposte multiple
label_nresp	String	"N. risposte"	Etichetta numero medio risposte multiple
label_i_conc	String	"Ind. conc."	Etichetta indice concentrazione
label_one_tail	String	"test a una coda"	Etichetta test una coda
label_two_tail	String	"test a due code"	Etichetta test due code
label_corr	String	"metodo di correzione"	Etichetta metodo di correzione
label_yates	String	"correzione di Yates"	Etichetta correzione di Yates
label_sig_level	String	"livello di significativita' "	Etichetta livello di significativita'
label_conf_level	String	"livello di confidenza"	Etichetta livello di confidenza
pc_test_level	Symbol	:conf	:conf => nota con livello di confidenza; :sig => nota
index_title	String	"Indice tavole"	Titolo indice
index_link_html	String	"Indice"	Link all'indice (Html)
index_link_xls	String	"(indice) "	Link all'indice (Excel)
table_continues	String	"(continua)"	Etichetta interruzione tabella
note_missed	String	"Valori mancanti: @"	Nota livelli esclusi dalla tabella (missed)
note_omitted	String	"# casi mancanti: @"	Nota livelli esclusi dalla tabella (omitted)
table_note	String	" "	Nota comune a tutte le tabelle seguenti
corner	String	" "	Etichetta nell'angolo in alto a sinistra della tabella
base_pos	Symbol	:corner	Posizione etichetta filtro: :corner, :title, :no

3.1.12 Formati

Parametro	Tipo	Default	Descrizione
encoding	String	"iso-8859-15"	Script character encoding (solo nel file yml)
dec_sep	String	", "	Separatore decimali
ths_sep	String	". "	Separatore migliaia (nil per nessun separatore)
frm_abs	Hash	{:p=>0}	Formato valori assoluti
frm_abs_n	Hash	{:p=>0}	Formato valori assoluti non ponderati
frm_pct	Hash	{:p=>1, :sx=>"%"}	Formato percentuali
frm_idx	Hash	{:p=>1}	Formato indice concentrazione
frm_base	Hash	{:p=>0}	Formato base non pesata
frm_base_n	Hash	{:p=>0}	Formato base
frm_mean	Hash	{:f=>:n, :p=>2, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato media
frm_sd	Hash	{:f=>:n, :p=>2, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato deviazione standard
frm_se	Hash	{:f=>:n, :p=>2, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato errore standard della media
frm_med	Hash	{:f=>:n, :p=>1, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato mediana
frm_q0	Hash	{:f=>:n, :p=>1, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato quartile 0
frm_q1	Hash	{:f=>:n, :p=>1, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato quartile 1
frm_q2	Hash	{:f=>:n, :p=>1, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato quartile 2
frm_q3	Hash	{:f=>:n, :p=>1, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato quartile 3
frm_q4	Hash	{:f=>:n, :p=>1, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato quartile 4
frm_iqr	Hash	{:f=>:n, :p=>1, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato scarto interquartile
frm_range	Hash	{:f=>:n, :p=>1, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato range
frm_sum	Hash	{:f=>:n, :p=>2, :t=>"%H:%M:%S", :d=>"%d-%m-%Y", :dt=>"%d-%m-%Y %H:%M:%S"}	Formato somma
frm_mresp	Hash	{:p=>1}	Formato numero medio risposte multiple
frm_nresp	Hash	{:p=>0}	Formato numero totale risposte multiple
frm_coord	Hash	{:p=>6}	Formato coordinate

3.1.13 Test

Parametro	Tipo	Default	Descrizione
chi-sq_test	Boolean	false	Attiva/disattiva il test del chi quadrato
chi-sq_labels	Array	["χ²", "df", "Sig."]	Etichette: chi-quadrato, gradi di liberta', significativita'
chi-sq_formats	Array	[:p=>3], {:p=>0}, {:p=>3}]	Formati: chi-quadrato, gradi di liberta', significativita'
chi-sq_warning_cell_freq	String	* sono presenti valori attesi f_{e}, il valore di χ² potrebbe non essere corretto"	Avvertimento celle con numerosita' bassa
pcm_test	Boolean	false	Attiva/disattiva il Pairwise Comparisons test sulle medie
pcm_test_sig	Float	0.05	Livello di significativita' t-test
pcm_test_tails	Integer	2	2=due code; 1=una coda
pcm_test_adj	Symbol	:none	Metodo di correzione p-value: :holm, :hoch, :homm, :bonf, :bh, :by, :none
label_pcm_test	String	"pairwise comparisons"	Etichetta Pairwise Comparisons
label_note_pcm	String	"pairwise comparisons medie"	Etichetta Pairwise Comparisons per le note
pcm_test_min	Integer	0	Numero minimo di casi per eseguire il test
pcm_test_note	Boolean	true	Attiva/disattiva le note
pcm_pvalue	Boolean	false	Stampa p-value nel log
pcp_test	Boolean	false	Attiva/disattiva il Pairwise Comparisons test sulle percentuali (BETA)
pcp_test_sig	Float	0.05	Livello di significativita' t-test
pcp_test_tails	Integer	2	2=due code; 1=una coda
pcp_test_adj	Symbol	:none	Metodo di correzione p-value: :holm, :hoch, :homm, :bonf, :bh, :by, :none
label_pcp_test	String	"pairwise comparisons"	Etichetta Pairwise Comparisons
label_note_pcp	String	"pairwise comparisons proporzioni"	Etichetta Pairwise Comparisons per le note
pcp_test_min	Integer	0	Numero minimo di casi per eseguire il test
pcp_test_note	Boolean	true	Attiva/disattiva le note
pcp_yates	Boolean	false	Attiva la correzione di Yates (correzione per la continuita')
pcp_pvalue	Boolean	false	Stampa p-value nel log

3.1.14 Analisi Corrispondenze

Pa- ra- me- tro	Ti- po	Default	Descrizione
ca_dims	Fi- x- num	2	Numero dimensioni grafici
ca_plots	Boo- lean	true	Genera i grafici
ca_plot_sym- bol	Sym- bol	:pdf	Formato file grafici: :pdf, :png, :wmf
ca_out	Boo- lean	true	Genera l'output testuale
ca_arrow	Ar- ray	[false, false]	Punti (false) o frecce (true). Un valore per X e uno per Y
ca_contri	Ar- ray	[:relative, :relative]	Indicazione dei contributi tramite l'intensita' dei colori: :none, :absolute, :relative. Un valore per X e uno per Y
ca_mass	Ar- ray	[true, true]	Indicazione delle masse come area dei punti. Comunque disabilitato se presenti punti supplementari. Un valore per X e uno per Y
ca_db	Boo- lean	true	Esporta coordinate nel db

3.1.15 Accentuazioni

Para- metro	Tipo	Default	Descrizione
phl	Boo- lean	false	Attiva/disattiva accentuazioni su percentuali colonna o indice di concentrazione
phl_ic	Boo- lean	false	Calcola accentuazioni in base all'indice di concentrazione e non alle percentuali di colonna
phl_lev	Ar- ray	[10, 20, 30]	Vettore con limiti intervalli accentuazioni (max 3)
phl_min	Fix- num	0	Numero minimo di casi nella cella
phl_minus	Boo- lean	true	Calcola accentuazioni anche su scostamenti negativi
mhl	Boo- lean	false	Attiva/disattiva accentuazioni medie
mhl_lev	Sym- bol	:auto	Vettore con limiti intervalli accentuazioni medie (max 3); se :auto utilizza 1/3 sdev, 1/2 sdev, 1 sdev
mhl_min	Fix- num	0	Numero minimo di casi nella cella
mhl_minus	Boo- lean	true	Calcola accentuazioni medie anche su scostamenti negativi

3.1.16 Top-line

Parametro	Tipo	Default	Descrizione
tl_index	Symbol	:long	Stampa indice: :none, :short, :long
tl_index_title	String	"Indice"	Titolo indice
tl_highlight_base	Boolean	true	Evidenzia le basi che cambiano tra una variabile e l'altra
tl_notitle	Boolean	false	Se tabulare le variabili senza titolo
tl_nodata	Boolean	true	Se tabulare le variabili vuote
tl_alpha	Boolean	false	Se tabulare le variabili alfanumeriche
tl_file_sx	String	"_top-line"	Suffisso per nome di file
tl_maxlev	Fixnum	50	Numero massimo di livelli per le distribuzioni di frequenza
tl_join	Boolean	true	Unisce le tabelle con le stesse etichette di livelli

3.1.17 Hole Count

Parametro	Tipo	Default	Descrizione
hc_maxlev	Fixnum	50	Numero massimo di livelli per le distribuzioni di frequenza

3.1.18 Grafici (HTML, PDF e pTabsWeb)

Parametro	Tipo	Default	Descrizione
chr_width	Fi- x- num	700	Larghezza grafico in pixel
chr_height	Fi- x- num	400	Altezza grafico in pixel
chr_legend	Styl- pos bol	right	Posizione legenda: :top :bottom :right :none
chr_caption	Boo- lean	true	Presenza del titolo all'interno di gruppo di grafici
chr_gauge	Hashtions	redFrom=>0, :redTo=>0, :yellowFrom=>0, :yellowTo=>0, :greenFrom=>0, :greenTo=>0, :minorTicks=>5}	Opzioni grafico gauge
chr_gauge	Fi- width x- num	100	Larghezza grafico gauge in pixel
chr_gauge	Fi- height x- num	100	Altezza grafico gauge in pixel
chr_gauge	Styl- caption bol	top	Presenza e posizione titolo grafico gauge all'interno di gruppo di grafici: :top :bottom :none
chr_gauge	Styl- label bol	inner	Presenza e posizione label grafico gauge: :inner :title :none
palette	Ar- ray	[]	Una palette di colori per i grafici

3.1.19 Mappe

Parametro	Tipo	Default	Descrizione
map_width	fix-num	800	Larghezza mappa in pixel
map_height	fix-num	600	Altezza mappa in pixel
map_search	Boolean	false	Mostra il riquadro di ricerca
map_search_text	String	stai ...	Testo riquadro di ricerca
map_zoom_control	Boolean	false	Mostra i pulsanti di controllo dello zoom
map_zoom_slider	Boolean	true	Mostra lo slider di controllo dello zoom
map_center	Array	[45, 10]	Centro geografico iniziale
map_scale	Boolean	true	Mostra scala
map_layers_control	Boolean	true	Mostra controllo selezione layers
map_markers	Symbol	mk	:mk (Maki), :fa (Font Awesome)
map_markers_size	Symbol	en	:s :m :l (solo Maki)
map_markers_icons_maki	Array	["cross", "circle", "square", "triangle", "circle-stroked", "square-stroked", "triangle-stroked"]	Nomi icone per le serie di punti (Maki)
map_markers_icons_fa	Array	["back-square-o", "star", "times", "circle", "square", "map-marker", "circle-o", "square-o", "thumb-tack"]	Nomi icone per le serie di punti (Font Awesome)
map_layers	HashMap	{ "OpenStreetMap"=>"OpenStreetMap", "OpenStreetMap B&W"=>"OpenStreetMap.BlackAndWhite", "Esri WorldStreetMap"=>"Esri.WorldStreetMap", "Outdoors"=>"Thunderforest.Outdoors", "MtbMap"=>"MtbMap", "Cycle"=>"Thunderforest.OpenCycleMap", "Trasporti"=>"Thunderforest.Transport", "Bianco e nero"=>"Stamen.Toner", "Satellite"=>"Esri.WorldImagery" }	Layers (map tiles principali)
map_overlays	HashMap	{ "Confini"=>"OpenMapSurfer.AdminBoundaries", "Strade"=>"Stamen.TonerLines", "Nomi"=>"Stamen.TonerLabels" }	Overlays (map tiles livelli sovrapponibili)
map_active_layer	fix-num	1	Numero layer principale preselezionato
map_data_series	String	"Interviste"	Nome serie dati principale
map_bound	fix-num	1	Serie su cui centrare la mappa
map_clusters	fix-num	1	Mostra i punti della serie aggregati: # (una specifica serie); :all (tutte le serie); :none (nessuna serie)
map_show_position	Boolean	true	Stampa le coordinate geografiche nel fumetto del punto

3.1.20 SPSS

Parametro	Tipo	Default	Descrizione
spss_use_value_labels	Boolean	true	Converte le variabili con etichette dei valori in factor R
spss_max_levels	Fix-num	100	Se :spss_use_value_labels=true : massimo numero di livelli di una variabile categoriale. Le variabili con un numero maggiore di livelli vengono considerate numeriche
spss_trim_value_labels	Boolean	true	Se :spss_use_value_labels=true : ignora gli spazi attorno ai valori e alle etichette dei valori quando vengono convertiti in factor
spss_trim_factor_names	Boolean	true	Elimina gli spazi attorno ai nomi dei livelli
spss_use_missing	Boolean	true	Casi con valori definiti come user-defined missing vengono impostati a valore mancante
spss_string_as_factor	Boolean	false	Converte le variabili alfanumeriche in factor

3.1.21 R

Parametro	Tipo	Default	Descrizione
r_log_width	Fix-num	150	Impostazione iniziale del numero colonne del log di R. Nello script usare log_width
total_varname	String	"._tot"	Nome della variabile R del totale casi
fake_weight_varname	String	"._fake.weight"	Nome della variabile R del peso uguale a 1
delete_table_objects	Boolean	false	Elimina precedenti tabelle

3.1.22 Java

Parametro	Tipo	Default	Descrizione
java_home	String	nil	Percorso Java SDK (deprecato, usare :jsdk_home_32)
jsdk_home_32	String	nil	Percorso Java SDK 32 bit
java_heap_xms	String	"64m"	Initial java heap size
java_heap_xmx	String	"640m"	Maximum java heap size
jre_home_32	String	nil	Percorso Java SE 32 bit
jre_home_64	String	nil	Percorso Java SE 64 bit

3.1.23 Logo e footer

Parametro	Tipo	Default	Descrizione
img	String	nil	Nome o indirizzo file immagine per prima pagina; formati: gif, jpeg, png, jpg
img_link	String	nil	Indirizzo link immagine
img_foot	String	nil	Nome o indirizzo file immagine per footer; formati: gif, jpeg, png, jpg
img_foot_link	String	nil	Indirizzo link immagine per footer
footer	String	nil	Testo footer

3.1.24 HTML

Parametro	Tipo	Default	Descrizione
html_print_index	Boolean	true	Stampa indice
html_print_nav	Boolean	true	Stampa pulsanti navigazione
html_print_footer	Boolean	true	Stampa footer
css_html	String	"ptabs.css"	Nome file foglio stile CSS per tavole
css_tl	String	"ptabs_tl.css"	Nome file foglio stile CSS per top-line
theme	Fix-num	1	Schema di colori attivo (numero, simbolo o stringa)

3.1.25 PDF

Parametro	Tipo	Default	Descrizione
wkhtmltopdf	String	"{pf32}/wkhtmltopdf/wkhtmltopdf.exe"	Percorso eseguibile wkhtmltopdf.exe
pdf_encoding	String	"iso-8859-15"	Character encoding file PDF
css_pdf	String	"ptabs_pdf.css"	Nome file foglio stile PDF
pdf_header	String	"pdf_header.html"	Nome file header
pdf_footer	String	"pdf_footer.html"	Nome file footer
pdf_page_row_number	Fix-num	49	Massimo numero di righe per pagina
pdf_zoom	Float	0.9	Percentuale ingrandimento/riduzione

3.1.26 Excel

Parametro	Tipo	Default	Descrizione
multiple_sheets	Boolean	false	Un foglio per ciascuna tabella
sheet_index_name	String	"indice"	Nome foglio indice
zero_xls	Boolean	false	Carattere che sostituisce lo zero nelle celle (false per lasciare 0)
xls_round	Boolean	true	Valori arrotondati o con tutti i decimali
xls_width_data	Float	8.43	Ampiezza delle colonne di dati
xhp1	String	nil	Altezza (in punti) prima riga pannello
xhp2	String	nil	Altezza (in punti) seconda riga pannello
xhnote	String	nil	Altezza (in punti) riga note

3.1.27 TXT

Parametro	Tipo	Default	Descrizione
txt_page_width	Fix-num	178	Ampiezza della pagina in caratteri
txt_page_height	Fix-num	64	Ampiezza della pagina in linee
txt_side_size	Fix-num	35	Ampiezza della fiancata in caratteri
txt_cell_size	Fix-num	8	Ampiezza delle celle in caratteri
txt_net_id	String	" (NET) "	Testo da aggiungere ai net
txt_net_indent	Fix-num	2	Numero spazi rientro net
min_split_chars	Fix-num	3	Numero minimo di caratteri in cui spezzare le etichette del pannello
hyphen_char	String	"\$"	Carattere per inserire un'interruzione nelle etichette del pannello (non inserisce il trattino)
break_char	String	" "	Carattere per attivare la sillabazione nelle parole delle etichette del pannello (inserisce il trattino)

3.1.28 Matrix

Parametro	Tipo	Default	Descrizione
matrix_round	Boolean	true	Valori arrotondati o con tutti i decimali
matrix	Boolean	true	Attiva/disattiva esportazione delle tabelle seguenti
mx	Boolean	false	Attiva/disattiva esportazione della prima tabella seguente

3.1.29 CSV

Parametro	Tipo	Default	Descrizione
csv_dec_sep	String	","	Separatore decimale
csv_col_sep	String	";"	Separatore campi

3.1.30 DB2ppt

Parametro	Tipo	Default	Descrizione
cchart	Symbol	:colclu	Tipo di grafico per statistiche count
schart	Symbol	:radar	Tipo di grafico per statistiche summary

3.2 Quartili

I valori che definiscono i **quartili** devono essere specificate con `probs_qrt` indicando congiuntamente i cinque valori. Impostare a `nil` i quartili che non si desidera mostrare.

`label_qrt` modifica in un'unica istruzione le etichette dei quartili.

`frm_qrt` imposta lo stesso formato per tutti i quartili.

```

1  qrt!
2  probs_qrt [0, 0.25, 0.5, 0.75, 1]
3  label_qrt ['Q0', 'Q1', 'Q2', 'Q3', 'Q4']
4  frm_qrt p:0

```

E' possibile calcolare il valore **minimo** e il valore **massimo** di una variabile impostando solo due valori:

```
1 probs_qrt [0, nil, nil, nil, 1]
2 label_q0 'MIN'
3 label_q4 'MAX'
```

Nota: La modifica dei valori che definiscono i *quartili* non influisce sul calcolo di **mediana** (`median`), **scarto interquartile** (`iqr`) e **range** (`range`).

3.3 Definizione dei formati numerici

I formati sono definiti attraverso un hash dove possono essere specificati i seguenti elementi:

- `:f`: formato attivo: `:n` (numerico), `:d` (data), `:t` (tempo), `:dt` (data/tempo)
- `:px`: prefisso (per esempio: \$)
- `:sx`: suffisso (per esempio: %)
- `:h`: carattere di separazione delle migliaia (per nessun separatore usare `nil`). Se presente, sovrascrive `ths_sep`
- `:s` carattere di separazione dei decimali (per esempio: ,). Se presente, sovrascrive `dec_sep`
- `:p` numero di decimali
- `:d` formato data
- `:t` formato tempo
- `:dt` formato data/tempo

Per una statistica possono essere definiti contemporaneamente i formati numerici e quelli di data e tempo. Il parametro `:f` definisce qual è il formato attivo.

frm_descr imposta contemporaneamente i formati per *media*, *somma*, *deviazione standard* e *errore standard della media*.

```
1 frm_pct :p => 1, :sx => '%'
2 frm_pct p:1, sx:'%'
3 frm_count p:0
4 frm_mean p:2
5
6 frm_mean t: '%H:%M' # cambia il formato tempo della media
7 frm_mean f: :t      # attiva il formato tempo per la media
8
9 frm_descr :f => :d # attiva il formato data per media, somma,
10                # deviazione standard e errore standard della media
```

Le sintassi `f: :t` e `f => :t` sono equivalenti. Attenzione, in `p: 2` lo spazio è facoltativo (`p:2`), ma in `f: :t` è obbligatorio.

Formati tempo: ¹

- `%Y` - anno a 4 cifre
- `%y` - anno a 2 cifre
- `%m` - mese in numero
- `%B` - mese in lettere
- `%b` - mese 3 lettere

¹ Le variabili di data e tempo devono avere formato numerico ed essere espresse in numero di secondi.

- %d - giorno del mese
- %A - giorno della settimana in lettere
- %a - giorno della settimana 3 lettere
- %H - ora
- %M - minuti
- %S - secondi

3.4 Temi

I colori e le dimensioni dei caratteri dei report HTML, PDF e Excel sono specificati nel file `theme.yml`. Ulteriori file `theme_<nome>.yml` (dove `nome` è il numero o il nome specificato nel parametro di configurazione `theme`) permettono di modificare anche solo parzialmente i colori. Con l'installazione di *pTabs* sono inclusi vari temi nella cartella `...\pTabs\lib\pt\resources`. `theme_1.yml` è vuoto e serve semplicemente per richiamare i colori di default, `theme_old.yml` è il vecchio schema di colori. Sono inoltre disponibili i file `theme_sea.yml`, `theme_sky.yml` e `theme_pink.yml`.

I file dei temi contengono cinque sezioni:

- parametri iniziati per `tl_`: colori top-line in formato HTML
- parametri iniziati per `rp_`: colori report in formato HTML, PDF e XLSX (Excel Open XML Format Spreadsheet)
- parametri iniziati per `xl_`: colori report in formato XLS (Excel 97-2003)
- parametri iniziati per `h_`: dimensioni caratteri formato HTML e PDF
- parametri iniziati per `x_`: dimensioni caratteri formato XLS e XLSX

Per i parametri `tl_` e `rp_` è possibile indicare il valore esadecimale del colore oppure il nome HTML/CSS. Per i parametri `xl_` (Excel 97-2003) è invece disponibile solo un numero limitato di colori.

Per utilizzare un tema, modificare il parametro `theme` nel file di configurazione, oppure specificare direttamente nello script:

```
1 report = ptabs.new do
2
3   theme :sea
```

`theme 1` richiama invece il tema di default.

L'istruzione `theme` accetta anche un blocco in cui è possibile specificare dei parametri direttamente nello script:

```
1 # viene caricato il tema sky variando i valori di alcuni parametri
2 theme :sky do |t|
3   t.h_report_title_sz 30
4   t.rp_report_title  "orange"
5 end
6
7 # i valori di alcuni parametri vengono modificati rispetto al tema di default
8 theme do |t|
9   t.h_table_caption_sz 16
10  t.x_index_caption_sz 14
11  t.rp_table_caption   :purple
12 end
```

E' possibile creare dei propri temi creando dei nuovi file, per esempio `theme_cliente.yml`, e specificando `:cliente` o `"cliente"` nel parametro di configurazione `theme` oppure `theme :cliente` direttamente nello script.

I nuovi file possono essere collocati o nella cartella di lavoro o nella *cartella dell'utente*, per es. `c:\stat\user`.

3.5 ptabs.yml

Tutti i parametri hanno dei valori di default specificati nel file `ptabs.yml`².

E' possibile utilizzare un proprio file `ptabs.yml` anche con parametri parziali. Più file di configurazione vengono letti in sequenza dalle seguenti posizioni:

- dal file eseguibile
- dalla cartella che contiene l'eseguibile
- dalla cartella `user`
- dalla cartella di sistema dell'utente³
- dalla cartella di lavoro

Le configurazioni presenti nei file nelle posizioni più basse della lista precedente hanno il sopravvento.

Suggerimento: La cartella `user` si trova nel percorso di installazione di *pTabs*. E' però possibile definire un percorso diverso impostando la variabile d'ambiente `PTABS_USER`. In tal caso la cartella `user` non è più attiva e bisogna utilizzare al suo posto quella definita in `PTABS_USER`.

```

1  # ptabs.yml
2
3  :footer: &copy; 2011 - Company Name
4  :img: my_company_logo_big.jpg
5  :img_foot: my_company_logo_small.jpg
6  :theme: 3
7  :dot_to_underscore: false
8  :underscore_to_dot: false
9  :vars_to_lower_case: false
10 :label_rowpct: "% r"
11 :label_colpct: "% c"
12 :frm_count:
13   :p: 0
14 :frm_pct:
15   :p: 2
16   :sx: "%"
17 :chisq_formats:
18 - :p: 3
19 - :p: 0
20 - :p: 3
21 :pdf_page_row_number: 40

```

3.6 File di risorse

I file di risorse di default possono essere estratti da pTabs e copiati nella cartella corrente tramite lo switch `rs`:

```
ptabs -rs
```

Note

² Il file è in formato [YAML](#).

³ La cartella definita dalla variabile d'ambiente `HOME` oppure dalle variabili `HOMEDRIVE` e `HOMEPATH`.

Sintassi

Gli script di *pTabs* seguono le regole sintattiche del linguaggio Ruby ¹.

- I comandi sono **case sensitive**, quindi vanno scritti con i corretti caratteri maiuscoli e minuscoli. ²
- La **virgola** separa gli argomenti di un metodo o funzione.
- Le **stringhe** di testo sono definite racchiudendo il testo tra virgolette singole o doppie.
- All'interno di una stringa delimitata da doppie virgolette, `#{nome_variabile}` inserisce il contenuto della variabile nel testo (*string interpolation*).

```
1 title "Tab. 25.#{index}."
```

- All'interno di una stringa `\n` manda a capo il testo seguente.
- Un'istruzione può essere mandata a capo purchè sia chiaro dal contesto che l'istruzione non è terminata. In genere viene mandata a capo dopo una virgola. Una stringa di testo non può essere semplicemente interrotta e mandata a capo, bisogna dividere la stringa e concatenarla con il carattere `+`.

```
1 ord :d20,
2     "Tab. 20. Dia un giudizio complessivo del prodotto testato. "+
3     "Utilizzi un voto da 1 a 10 per esprimere il suo giudizio."
```

- Due numeri uniti da due punti `..` rappresentano un **range**, una sequenza numerica: `1..10`.
- I caratteri preceduti da due punti `:` sono chiamati **simboli** (`:q1`). Ciascun simbolo viene istanziato una volta sola indipendentemente dal numero di chiamate. In *pTabs* si usiamo soprattutto per i nomi di parametri e per i nomi di variabili. Se però il nome della variabile contiene un punto, bisogna usare una stringa.

```
1 md s(:d7_, 2..12)
2 md s('d7.', 2..12)
```

- Le **parentesi tonde** racchiudono gli argomenti di un metodo o funzione. Se il contesto non è ambiguo o se non si concatena direttamente un metodo sulla stessa funzione, si possono tralasciare.

```
1 vars s :d6_, 10
2 vars s(:d7_, 4), s(:d7_, 7..11)
3
4 report.save :r
5 report.save (:r).print
```

- Le **parentesi quadre** racchiudono gli **array** (vettori). In *pTabs* quando un array costituisce l'unico argomento di una funzione, si possono tralasciare.

```
1 ord [d5, :r5], "Titolo tabella" # gli argomenti di ord sono due, l'array delle variabili e
2                                # senza parentesi diventerebbero tre argomenti distinti
3 ord :d5 do
```

¹ Lo script è effettivamente codice Ruby che viene interpretato da `ruby.exe`.

² Comunque, al momento, tutti i comandi sono in minuscolo.

```

4     w nil, -10, -5, 5, 10           # la sequenza di valori è un array, ma essendo l'unico argo
5     end                             # la funzione w li interpreta correttamente anche senza pa

```

- L'operatore => viene usato per associare la chiave di un **hash** (array associativo) al suo valore. In pTabs vengono utilizzati per specificare delle opzioni oppure per associare dei valori alle variabili:

```

1   show_missing :out => "report_na"
2
3   select :d1 => [1,2], :sesso => 1

```

Una sintassi alternativa più semplice, ma che può essere utilizzata solo quando la chiave dell'hash è un simbolo, consiste nell'utilizzare il carattere di duepunti dopo il simbolo, seguito dal valore:

```

1   show_missing out:"report_na"
2
3   select d1: [1,2], sesso: 1

```

- A un metodo o funzione, oltre agli argomenti può essere passato un **blocco**. Un blocco è una porzione di codice (una *funzione anonima*, una *closure*) che viene eseguito nel contesto definito dall'oggetto a cui viene passato il blocco. In pTabs tutto il report è definito come blocco e anche le tabelle utilizzano un blocco quando occorre specificare maggiori parametri oltre alle variabili e al titolo. Un blocco utilizza la sintassi `do ... end`.³ Lo stesso blocco può avere degli argomenti: `do |args| ... end`.

```

1   # i metodi sort e first agiscono sulla tabella che si sta definendo
2   md s(:d7_, 10) do
3     sort :d
4     first 3
5   end
6
7   # rep cicla su ciascuna variabile della sequenza e nel contempo rende disponibile a ogni ciclo
8   # il nome e l'indice della variabile
9   # (var e index sono nomi arbitrari, potremmo per esempio chiamarli v e i)
10  s(:q25_, 1..10).rep do |var, index|
11    ord var, title "25.#{index}. Giudizio riguardo:\n$var_label()"
12  end

```

Note

³ I blocchi in Ruby possono utilizzare anche la sintassi `{ |args| ... }`. Per convenzione questa sintassi è utilizzata quando il blocco è di una sola riga.

Scripts

Uno **script** di *pTabs* è un file di testo con estensione `.pt`. Uno script contiene la definizione di uno o più report.

Un **report** è un blocco di codice passato al metodo `ptabs.new`. L'oggetto Report generato normalmente viene per comodità assegnato a una variabile.

```
1 report = ptabs.new do
2   ...
3 end
```

All'interno del blocco di codice vengono definiti gli elementi del report: titolo, sottotitolo e piè di pagina del report, file di dati, pannelli, tabelle, ecc.

```
1 report = ptabs.new("nome_outputs") do
2   title      "Titolo"
3   subtitle   "Sottotitolo"
4   footer     "Piè di pagina"
5
6   spss.open 'esempi_c tables'
7   weight.by :peso
8
9   panels.add :_tot, :sesso, :eta, :titolo
10  panels.add :_tot, :area, :ampc
11
12  cat :q1
13  ord :q2
14  md s(:d3,10)
15  ...
16 end
```

Infine una serie di metodi di Report sono disponibili per elaborare gli output.

```
1 report = ptabs.new("nome_outputs") do
2   ...
3 end
4
5 report.render :html
6 report.render :pdf
7 report.render :xls
8 report.render :text
```

E' possibile generare una bozza dello script tramite lo switch `sc`:

```
ptabs -sc scriptname
```

5.1 %include

L'istruzione `%include nomefile` permette di includere il contenuto di un file esterno. Questa istruzione viene eseguita prima di qualsiasi interpretazione del contenuto dello script, pertanto il nome del file non può essere generato dinamicamente, ma deve essere una costante. Si vedano anche le istruzioni *ptabs.inc* e *inc*.

L'oggetto ptabs

L'oggetto `ptabs` è l'oggetto base del programma **pTabs**. Permette la definizione dei report e la gestione di altri file. E' anche possibile usare l'abbreviazione `pt`).

6.1 ptabs.new

Inizializza e definisce il contenuto di un report. Un report è un blocco di codice passato al metodo `ptabs.new()`, un helper che istanzia un nuovo oggetto `PTables::Report`. Assegnando l'oggetto `Report` a una variabile è poi possibile invocare i metodi per la generazione dei vari formati di output. In uno script possono essere definiti più report.

```
1 # crea gli output nella stessa cartella in cui risiede lo script.
2 # i nomi degli output vengono generati a partire dal nome dello script
3 report = ptabs.new do
4   ...
5 end
6
7 # crea gli output nella stessa cartella in cui risiede lo script
8 report = ptabs.new("nome_outputs") do
9   ...
10 end
11
12 # specifica una cartella di default differente
13 report = ptabs.new("percorso/cartella/lavoro", "nome_outputs") do
14   ...
15 end
```

6.2 ptabs.csv

Apri un file `csv` e cicla sulle sue righe. Passa nel blocco due argomenti: la riga e l'indice della riga. L'indice parte da 1 e non include la riga delle intestazioni dei campi. Il separatore di campo di default è il punto e virgola (;).

```
1 ptabs.csv('nome_file_csv', separatore_di_campo) do |row, index|
2   ...
3 end
```

In questo esempio la funzione `myrep` racchiude la definizione di un report e permette la sua parametrizzazione: nell'esempio viene parametrizzato il nome del report, il sottotitolo e il codice dell'area usato per generare il nome del file di dati. `ptabs.csv` cicla sulle righe del file `aree.csv` e utilizza i dati contenuti per eseguire più volte il report.

```
1 def (User::Ptabs).myrep(row)
2
3   reptime, replabel, id_area = row.field('sigla'), row.field('label'), row.field('code').to_i
4
5   ptabs.new(reptime) do
6     ...
7     subtitle replabel
8     ...
9     spss.open 'area_'+id_area
10    ...
11  end
12  report.render :html
13
14 end
15
16 ptabs.csv('aree') do |row, index|
17   ptabs.my_report row
18 end
```

Il contenuto del file *aree.csv*:

```
sigla;label;code
NO;Nord Ovest;1
NE;Nord Est;2
C;Centro;3
SI;Sud e Isole;4
```

6.3 ptabs.inc

Include un file di istruzioni nello script che si sta eseguendo. Questo comando va utilizzato all'esterno della definizione del report. Il file deve contenere un blocco di codice sintatticamente corretto. L'estensione predefinita è `.pt`. Si vedano anche le istruzioni `%include` e `inc`.

6.4 Variabili

Una variabile definita all'interno del *blocco del report* non sarebbe visibile all'esterno del blocco. Per impostare variabili a livello dell'oggetto *ptabs* usare i metodi `.varname(value)` e `.varname()` dell'oggetto *ptabs*.

```
1 report = ptabs.new do
2
3   ...
4   ptabs.pdf = true if ...
5   ...
6
7 end
8
9 report.render :pdf if ptabs.pdf
```

Report

Il report è definito dal *blocco* di codice passato al metodo `ptabs.new("nome")`. All'interno del blocco è possibile definire il titolo, il sottotitolo e il piè di pagina del report. Il nome passato al metodo `new` viene usato come nome di file per i vari output, altrimenti viene utilizzato lo stesso nome dello script (senza estensione). Il metodo `name` permette di ridefinire il nome dei file di output.

```

1  report = ptabs.new("nome") do
2    title   "Titolo"
3    subtitle "Sottotitolo"
4    footer  "Piè di pagina"
5    name    "nuovo_nome"
6    ...
7  end

```

7.1 inc

All'interno della definizione di un report è possibile utilizzare il comando `inc` per includere un file di istruzioni *pTabs* nello script che si sta eseguendo. Il file deve contenere un blocco di codice sintatticamente corretto.¹ L'estensione predefinita è `.pt`.² Si vedano anche le istruzioni `%include` e `ptabs.inc`.

```

1  # File d6.pt
2  se!
3  scale s(:d6_, 10), "D6. Valori medi (anche standard error)" do
4    weights nil, 25, 50, 75, 100
5    autonote
6    sort :d, :mean
7  end
8  se_
9
10 # Script principale
11 ...
12 inc 'd6'

```

7.2 Output

Una serie di metodi di Report sono disponibili per elaborare l'output. I metodi ritornano sempre l'oggetto Report e quindi possono essere concatenati.

```

1  report = ptabs.new("nome") do
2    ...
3  end.print.save(:r)

```

¹ Eventuali definizioni di variabili saranno visibili solo all'interno del file che si include.

² Per includere un file con istruzioni Ruby generiche (non *pTabs*) usare i metodi `load` o `require`.

```
4  
5 report.render(:html).render(:pdf)
```

7.2.1 print

Stampa nel log le istruzioni **R**.

7.2.2 save

`save(:r)` salva la sintassi **R** in un file.

7.2.3 render

`render` genera i file con le tavole del report nei vari formati disponibili: HTML, Pdf, Excel, Testo e come matrice di dati.

Note

Genera gli output in vari formati.

E' possibile generare nello stesso batch più file di formato diverso o anche dello stesso formato.

8.1 html

Genera il report in formato **HTML**.

- il layout è modificabile attraverso fogli stili CSS
- indice con link alle tabelle

```
1 report.render :html
```

8.2 pdf

Genera il report in formato **PDF**.¹

- il layout è modificabile attraverso fogli stili CSS
- intestazione e piè di pagina modificabile
- numerazione pagine e pagine totali
- impaginazione tabelle lunghe su più pagine
- indice delle tabelle nei segnalibri
- logo nella prima pagina e nel piè di pagina

```
1 report.render :pdf
2 report.render :pdf, :file => "altro_nome"
```

8.3 Excel

8.3.1 xls

Genera il report in formato **Excel 97-2003**.

Parametri:

- **single_panel**: pannelli originari (`false`) o unione in un pannello unico (`true`)

¹ I file PDF sono generati attraverso `wkhtmltopdf`.

- **multiple_sheets**: tabelle in un unico foglio (`false`) o un foglio per tabella (`true`)
- **file**: nome del file di output (se diverso dal nome del report)
- **stat_col***: permette di aggiungere (`true`) una colonna con le etichette delle statistiche
- **stats**: array delle statistiche da includere nelle tabelle. `:all` per includerle tutte. Se il valore è `:all` o il parametro non è specificato verranno incluse tutte le statistiche.
- **sheet**: nome del foglio. Se non specificato è uguale al nome del file.
- **sheets**: permette di specificare più fogli in uno stesso file. Hash con i nomi dei fogli e gli array delle statistiche da includere nelle tabelle
- **xhp1**: (in punti) forza l'altezza della prima riga del pannello (etichette delle variabili)
- **xhp2**: (in punti) forza l'altezza della seconda riga del pannello (etichette dei valori)
- **xhnote**: (in punti) forza l'altezza della riga delle note

```

1 report.render :xls, :single_panel => true, :stats => [:colpct, :base, :mean, :sd]
2 report.render :xls, :single_panel => true, :multiple_sheets => true, :stats => :all, :file => "a
3 report.render :xls, :single_panel => true, :xhp2 => 60,
4       :sheets => { "VA %" => [:all],
5                   "%"   => [:tcount, :trowcol, :rowpct, :colpct, :mresp, :base
6                   "VA"  => [:tcount, :trowcol, :count, :mresp, :base

```

8.3.2 xlsx

Genera il report in formato **Excel Open XML Format Spreadsheet**.

Accetta gli stessi parametri di `xls`.

```
1 report.render :xlsx
```

- **xls**
 - generazione del report più veloce
 - fino a 256 colonne
 - fino a 65.536 righe
 - file di dimensioni più grandi
 - numero limitato di colori
 - richiede l'installazione del JDK
- **xlsx**
 - generazione del report più lenta
 - fino a 16.384 colonne
 - fino a 1.048.576 righe
 - file di dimensioni più piccole
 - consente di utilizzare gli stessi colori utilizzati nell'output HTML e PDF
 - non richiede l'installazione del JDK

8.4 ptweb

Esporta le tabelle nel formato utilizzato da *pTabsWeb*.

```

1 # genera il file <nome_report.db>
2 report.render :ptweb, :file => "nome_report"

```

All'interno dello script sono disponibili le istruzioni `period` e `stratum` per etichettare e raggruppare le tabelle (vedi *Selezione delle osservazioni*).

8.5 db

Esporta le tabelle come **matrice di dati**. I formati di dati disponibili sono: **Access**, **csv**, **xlsx** e **xls**.

Utilizzando il formato *Access*, viene creato un **database** con interfaccia grafica (*pTabsDB*) per la navigazione delle tabelle, la produzione e l'aggiornamento di grafici e tabelle in *Power Point*.

Utilizzando i formati *csv*, *xlsx* e *xls* viene generata solo la matrice di dati.

Parametri:

- **id**: una stringa di testo che identifica i dati, se non specificato utilizza l'id del report (corrisponde al campo `matrix`)
- **period**: una stringa di testo che identifica il periodo (corrisponde al campo `period`)
- **db** o **file**: il nome del file o del database
- **append**: accoda i dati in una tabella esistente altrimenti sostituisce la tabella; default: `false`
- **clear**: prima di caricare i nuovi dati, elimina dal database i dati con stesso *id* e stesso *period*. Permette di ricaricare un set di dati. Solo in unione con *append*. Default: `false`
- **company**: nome dell'azienda che deve comparire nell'interfaccia
- **version** o **v**: genera un db nella versione dell'interfaccia specificata
- **format**: il formato del file di output: `:access` | `:csv` | `:xlsx` | `:xls`. Il default è `:access`.

L'estensione per il formato *Access* è `accdb`. E' necessario *Access 2010* per utilizzare l'interfaccia del database. Il file *Access* se non esiste viene creato.

```

1 report.render :db, :id => "Main sample", :period => "2012 trim 1", :append => true

```

La struttura della tabella è fissa, non dipende dai pannelli o dal tipo di tabelle prodotte. I dati sono identificati attraverso una serie di campi che costituiscono una chiave univoca. Pertanto, oltre all'utilizzo con interfaccia grafica, è possibile utilizzare programmaticamente il database estraendo i dati attraverso delle query SQL. Questo formato può costituire una solida base per report e presentazioni.

Struttura della tabella:

Campo	Chiave	Tipo	Descrizione
matrix	x	string	identificativo della matrice di dati
period	(x)	string	identificativo del periodo (default vuoto)
stratum	x	string	identificativo dello strato/selezione (default vuoto) ⁴
table	x	string	identificativo della tabella ⁵
subtable	x	integer	indice sottotabella
rvar	x	string	nome variabile di riga
rid	x	integer	indice riga
rlabel		string	etichetta riga
net		integer	distingue gli item normali (0) dai net (1)
cvarid	x	integer	identificativo progressivo della variabile di colonna
cvar	x	string	nome variabile di colonna
cvarlabel		string	etichetta variabile riga
cid	x	integer	indice colonna
clabel		string	etichetta colonna
stat	x	string	tipo statistica: count o summary
count		float	conteggio casi
rpct		float	percentuale di riga
cpct		float	percentuale di colonna
base		float	base casi
base_u		float	base casi non ponderati
mean		float	media
sd		float	deviazione standard
se		float	errore standard della media
nresp		float	numero totale risposte multiple
mresp		float	numero medio risposte multiple
coord		float	coordinate
i_conc		float	indice concentrazione
accent		integer	flag accentuazioni

Il campo periodo quando usato, diventa parte della chiave primaria della tabella Access.

8.6 Opzioni

Alcune opzioni disponibili nella generazione degli output:

²Viene utilizzato il valore di `id` impostato dalle struzioni `filter`, `select` e `all`.

³pTabs genera un identificativo automatico per ciascuna tabella che dipende dall'ordine di generazione. E' possibile fornire un identificativo esplicito nella definizione della tabella tramite il sottocomando `id`.

⁴Viene utilizzato il valore di `id` impostato dalle struzioni `filter`, `select` e `all`.

⁵pTabs genera un identificativo automatico per ciascuna tabella che dipende dall'ordine di generazione. E' possibile fornire un identificativo esplicito nella definizione della tabella tramite il sottocomando `id`.

Parametro	T/F	Descrizione
file		nome del file di output (senza estensione)
single_panel	x	genera un unico pannello
multiple_sheets	x	mette ogni tabella in un foglio distinto (Excel)
matrix_id	x	nell'output Excel, aggiunge gli identificativi della matrice
rtotal	x	attiva/disattiva totale di riga
tcount	x	attiva/disattiva valori assoluti totale
trowpct	x	percentuali riga totale
tcolpct	x	percentuali colonna totale
count	x	attiva/disattiva valori assoluti
rowpct	x	attiva/disattiva percentuali riga
colpct	x	attiva/disattiva percentuali colonna
base_u	x	attiva/disattiva base non pesata
base	x	attiva/disattiva base
mean	x	attiva/disattiva media
sd	x	attiva/disattiva deviazione standard
se	x	attiva/disattiva errore standard della media

Note

Fonti di dati

pTabs può gestire dati in formato **R**, **SPSS**, **PSPP**, **XPORT (SAS)**, **CSV**, **Testo**, **Excel**, **Access**, **SQL Server** e **ODBC**.

9.1 Metodi

Alcuni metodi (funzioni) sono comuni a più fonti di dati. Lo stesso metodo può accettare parametri differenti in relazione alla specifica fonte di dati.

Fonte dati	Driver	Estensione	open	save	labels	query
R	r	rData	X	X		X
SPSS	spss	sav	X	X	X	X
PSPP	spss	sav	X	X	X	X
SAS	xport	xpt	X	X		X
ODBC	odbc	—	X		X	X
SQL Server	sqlserver	—	X		X	X
Excel 97-2003	excel	xls	X	X	X	X
Excel 2007-2010	excelx	xlsx	X	X	X	X
Access fino 2003	jet	mdb	X		X	X
Access 2007-2010	ace	accdb	X		X	X
Triple-S	sss	asc / sss	X			
Testo	fwf	dat	X	X		
Nipo Odin	odin	dat / var	X			
CSV	csv	csv	X	X		X

Altri metodi specifici di alcuni formati sono specificati nella sezione relativa alla fonte di dati.

9.1.1 open

Carica una fonte di dati in un *data frame* e lo rende disponibile a *pTabs*.

Non è necessario specificare le estensioni dei file se si utilizzano quelle standard.

I dati vengono caricati in un *data frame* chiamato `dfr`.

Argomenti:

- `filename`: il nome del file da caricare
- `:labels => name` il nome del foglio/tabella contenente le etichette

E' possibile controllare il modo in cui i nomi di variabili vengono caricati attraverso i parametri `dot_to_underscore`, `underscore_to_dot` e `vars_to_lower_case`.

.open!

Il metodo `.open!` seguito da un blocco permette di aprire il file, eseguire il blocco di istruzioni e salvare il file in formato *R* (`nomefile._pt_.RData`). Il file conterrà eventuali trasformazioni (variabili, etichette, ecc.) presenti nel blocco di istruzioni. Successive esecuzioni dello script caricheranno direttamente il file *RData* (se esiste) senza rieseguire il blocco di istruzioni.

Con il metodo `.open?` si può forzare il caricamento del file originario, la riesecuzione del blocco e il salvataggio del file *RData* anche se già presente.

Con il metodo `.open` si ha invece un comportamento normale con caricamento del file originario ed esecuzione del blocco senza salvare il file *RData*.

```
1 spss.open! 'dati' do
2   debug "----- inizio blocco eseguito solo se viene ricaricato il .SAV e non l'.RData"
3   excelx.labels 'labels2'
4   levels :sesso => ["M", "F"]
5   debug "----- fine blocco"
6 end
```

9.1.2 save

Salva i dati del *data frame* attivo.

Argomenti:

- `filename`: il nome del file in cui salvare i dati
- `:vars => [varlist]`: l'elenco di variabili da salvare (`:all` esporta tutte le variabili)
- `:id => :varname`: ordina i record sulla variabile indicata

9.1.3 query

Interroga la fonte di dati e restituisce il risultato. Non carica la fonte di dati nel *data frame* attivo.

Argomenti:

- `filename`: il nome del file da interrogare
- `:table => tablename`: il nome della tabella o del foglio di cui restituire i dati
- `:var => [varlist]`: l'elenco di variabili da restituire
- `:sql => string`: la query SQL con cui interrogare i dati. E' disponibile per tutte le fonti dati che supportano *query*, non solo per i database.

Specificando solo il nome del file e/o la tabella (per i database) viene restituito l'intero contenuto del file/tabella. `:var` permette di limitare il risultato a un sottoinsieme di variabili. Per query più complesse specificare una query SQL nel parametro `:sql`.

query restituisce un vettore se il risultato comprende una sola variabile oppure un vettore di vettori se sono state richieste più variabili.

```
1 # fuori dal blocco del report
2 debug ptabs.new.excelx.query('aree')
3 #<Array> [{"NO", "Nord Ovest", "1"}, {"NE", "Nord Est", "2"}, {"C", "Centro", "3"}, {"SI", "Sud e
4
5 # nel report
6 debug spss.query('labels', :sql => "select name from labels where code is null")
7 #<Array> ["Sesso", "area", "d1", "d5"]
```

<<

9.2 Fonti di dati

9.2.1 R Data Frame

r gestisce dati in formato R.

Se il file contiene un *data frame* con un nome differente da `dfr`, è necessario specificarlo nel parametro `:dframe`.

```
1 # apre un data frame R (.RData)
2 r.open "nome_file", :dframe => 'dati'
3
4 # salva il file aperto in formato R (.RData)
5 r.save "nome_file"
```

Metodi aggiuntivi:

- `r.dump` salva tutti i workspace.
- `r.restore` ripristina tutti i workspace.

<<

9.2.2 SPSS System File

spss gestisce dati in formato SPSS.

Caricando un file di dati *SPSS*, le variabili con almeno una *value label* vengono trasformate in *factor*. I livelli delle variabili *factor* vengono creati rispettando l'ordine dei codici della variabile SPSS, ma i codici originali vengono persi. Internamente, i livelli sono memorizzati con numeri interi progressivi che partono da 1.

Avvertimento: I codici delle variabili SPSS con *value label* devono essere numeri interi e non possono esserci etichette uguali.
 Il codice 0 verrà convertito in 1 e i restanti codici verranno riscaldati in avanti.
 Sequenze di codici con salti verranno compressi in una sequenza di numeri interi progressivi e continui.
 I valori mancanti definiti dall'utente (*missing values*) vengono trasformati in effettivi valori mancanti (*system missing*).

```
1 # apre un file SPSS (.sav)
2 spss.open "nome_file"
3
4 # salva i dati in un file SPSS (.sav) ordinato per 'ser_no'
5 spss.save "nome_file", :id => :ser_no, :vars => [:sesso, :eta, :area, :ipf, :espcam, :espuni]
```

Suggerimento: [ViewSav, a viewer for SPSS data files](#) è un programma free particolarmente utile per *visualizzare* velocemente il contenuto dei file di dati SPSS.

Suggerimento: [PSPP](#) è un programma free che permette di *creare* e *gestire* file di dati SPSS.

<<

9.2.3 GNU PSPP

PSPP (GNU PSPP) è un clone open-source di SPSS. Sebbene non supporti tutte le caratteristiche di SPSS, ha la stessa sintassi e scrive i dati nello stesso formato, pertanto valgono tutte le osservazioni e le istruzioni fornite per SPSS.

<<

9.2.4 SAS Transport File

xport gestisce dati SAS esportati in formato **Transport File** (*XPORT*). Il file XPORT deve contenere un solo dataset.

I nomi delle variabili vengono convertiti in minuscolo.

E' necessario aver installato la libreria R `SASxport`.

```
1 # apre un file xport (.xpt)
2 xport.open "nome_file"
3
4 # salva i dati in un file XPORT (.xpt)
5 xport.save "nome_file", :vars => [:sesso, :eta, :area, :ipf, :espcam, :espuni]
```

<<

9.2.5 Triple-S

sss gestisce dati in formato **Triple-S**.

Oltre al file di dati, è necessario specificare anche il file contenente i *metadati Triple-S*. Le estensioni predefinite sono `.asc` per i dati e `.sss` per i metadati.

E' possibile specificare il carattere di separazione che viene utilizzato per generare le variabili delle domande a risposta multipla. Il carattere predefinito è `_`.

E' necessario aver installato la libreria R `sss`.

Argomenti aggiuntivi:

- `:sep => "_"` carattere di separazione per nomi variabili multiple

```
1 # legge i file survey.asc e survey.sss
2 sss.open "survey"
3
4 # specifica i file con le estensioni e il carattere di separazione per le multiple
5 sss.open "survey2.dat", "survey2.xml", sep="_"
```

<<

9.2.6 Testo

fwf gestisce dati in file ASCII in formato fisso (*Fixed Width Format*). L'elenco delle variabili e delle posizioni viene passato al parametro `:data` come vettore di stringhe.¹ Le variabili alfanumeriche vengono convertite di default in *factor*. Per caricarle come caratteri (per esempio per i dati anagrafici) specificare il formato `:a`. Gli allineamenti non sono rilevanti.

Argomenti aggiuntivi:

- `:nrows => #` massimo numero di righe da leggere (default `-1`, tutte).
- `:buffer => #` massimo blocco di righe da leggere (default `2000`). Un numero basso diminuisce l'uso di memoria, ma aumenta il tempo di lettura.
- `:as_num` determina se le variabili *factor* devono essere convertite in variabili numeriche (`true`, il default), oppure se devono essere salvate le etichette al posto dei codici (`false`).
- `:vars => [varlist]` specifica l'elenco delle variabili (`:all` esporta tutte le variabili).
- `:id => :varname` specifica una variabile su cui ordinare i dati.

E' necessario aver installato la libreria R `ffbase`.

¹ `%w(...)` crea un vettore di stringhe senza la necessità di usare le virgolette e separare gli elementi con la virgola.

```

1  # apre un file di testo (.dat)
2  fwf.open "nome_file", :data => %w(
3    ser_no    1-4
4    area      5
5    sesso     6
6    eta       9-10
7    titolo    11
8    prof      12-13
9    d1        15
10   d2        16
11   d5        17
12   nome      50-59 :a
13   cognome   60-69 :a
14   indirizzo 70-99 :a
15 )
16
17 fwf.open "nome_file", :data => %w(ser_no 1-4 area 5 d1 15 d5 17 cognome 60-69 :a)
18
19 fwf.save "nome_file", :id => :ser_no, :vars => [:sesso, :eta, :area, :ipf, :espcam, :espuni], :a

```

<<

9.2.7 File delimitati: CSV

csv gestisce dati in formato delimitato.

La prima riga del file deve contenere i nomi delle variabili.

Con `csv.save`, oltre al file dei dati, viene salvato anche un secondo file (viene aggiunto il suffisso `_labels`) con le etichette delle variabili.

Argomenti aggiuntivi:

- `:as_num` determina se le variabili factor devono essere convertite in variabili numeriche (`true`, il default), oppure se devono essere salvate le etichette al posto dei codici (`false`).
- `:vars => [varlist]` specifica l'elenco delle variabili (`:all` esporta tutte le variabili).
- `:id => :varname` specifica una variabile su cui ordinare i dati.
- `:sep => "c" o :s => "c"` specifica il separatore di campo (default `,`).
- `:dec => "c" o :d => "c"` specifica il separatore dei decimali (default `.`).

Le varianti `csv1` e `csv2`, impostano automaticamente i separatori come `:s=>' , '`, `:d=>' . '` (`csv1`) e `:s=>' ; '`, `:d=>' , '` (`csv2`).

```

1  csv.open "nome_file", :sep => ';', :dec => '.'
2  csv1.open "nome_file"
3  csv2.save "nome_file", :id => :ser_no, :vars => [:sesso, :eta, :area, :ipf, :espcam, :espuni]

```

<<

9.2.8 Excel 97-2003

excel gestisce file *Excel* nel formato 97-2003 (`.xls`).

Il file deve contenere i nomi delle variabili nella prima riga del foglio di lavoro. Se non viene specificato il nome del foglio di lavoro, si assume che il nome sia uguale al nome del file.

Con `excel.save`, oltre al foglio dei dati, viene salvato anche un secondo foglio con le etichette delle variabili.

Argomenti aggiuntivi:

- `:as_num` determina se le variabili `factor` devono essere convertite in variabili numeriche (`true`, il default), oppure se devono essere salvate le etichette al posto dei codici (`false`).
- `:codebook` determina se esportare anche un foglio con le etichette delle variabili (`true`, il default).
- `:names => :name|:label|:both` determina l'intestazione delle colonne: `:name` il nome della variabile (il default); `:label` l'etichetta della variabile; `:both` entrambe.
- `:vars => [varlist]` specifica l'elenco delle variabili (`:all` esporta tutte le variabili).
- `:id => :varname` specifica una variabile su cui ordinare i dati.

```
1 # apre un file Excel (.xls)
2 excel.open "nome_file", :sheet => "nome_foglio"
3
4 excel.save "nome_file", :id => :ser_no, :vars => [:sesso, :eta, :area, :ipf, :espcam, :espuni],
5           :sheet => "nome_foglio", :as_num => false
```

<<

9.2.9 Excel 2007

`excelx` gestisce file *Excel* nel formato 2007-2010 (`.xlsx`).

Ha la stessa sintassi di `excel`.

```
1 excelx.open "nome_file", :sheet => "nome_foglio"
2 excelx.save "nome_file", :id => :ser_no, :vars => :all
```

<<

9.2.10 Access: jet

`jet` gestisce file *Access* nel formato 2003. ²

E' possibile specificare la tabella da aprire oppure la query SQL.

Argomenti aggiuntivi:

- `:uid => login` il nome dell'account per l'accesso al database
- `:pwd => password` la password per l'accesso al database
- `:table => table_name` il nome della tabella (se non usato `sql`)
- `:sql => sql_string` la stringa di istruzioni in linguaggio SQL (se non usato `table`)

```
1 # apre un file Access formato 2003 (.mdb)
2 jet.open "nome_file", :table => "nome_tabella"
3 jet.open "nome_file", :table => "nome_tabella", :uid => "login", :pwd => "password"
4 jet.open "nome_file", :sql => "SELECT * FROM nome_tabella"
```

<<

9.2.11 Access: ace

`ace` gestisce file *Access* nel formato 2007-2010. ³

E' possibile anche aprire file *Access* nel formato 2003 specificando l'estensione nel nome del file.

Ha la stessa sintassi di `jet`.

² Solo con R a 32 bit, ma sia con la versione a 32 bit di Microsoft Office, sia con la versione a 64 bit.

³ Solo se la versione di R e di Microsoft Office corrisponde, cioè se sono entrambi a 32 bit o entrambi a 64 bit.

```

1 # apre un file Access formato 2007-2010 (.accdb)
2 ace.open "nome_file", :table => "nome_tabella"
3 ace.open "nome_file", :table => "nome_tabella", :uid => "login", :pwd => "password"
4 ace.open "nome_file", :sql => "SELECT * FROM nome_tabella"
5
6 # apre un file Access formato 2003 (.mdb)
7 ace.open "nome_file.mdb", :table => "nome_tabella"

```

<<

9.2.12 Origine dati ODBC

odbc gestisce origini dati *ODBC*.

E' possibile specificare la tabella da aprire oppure la query SQL.

Argomenti aggiuntivi:

- `dsname` il *Database Source Name (DSN)* registrato sul computer
- `:uid => login` il nome dell'account per l'accesso al database
- `:pwd => password` la password per l'accesso al database
- `:table => table_name` il nome della tabella (se non usato *sql*)
- `:sql => sql_string` la stringa di istruzioni in linguaggio SQL (se non usato *table*)

E' possibile specificare la tabella da aprire oppure la query SQL.

```

1 # apre un'origine dati ODBC
2 odbc.open "nome_DSN", :uid => "login", :pwd => "password", :table => "nome_tabella"
3 odbc.open "nome_DSN", :uid => "login", :pwd => "password", :sql => "SELECT * FROM nome_tabella"

```

<<

9.2.13 SQL Server

sqlserver gestisce database *Microsoft SQL Server*.

E' possibile accedere al database tramite un *DSN* registrato oppure direttamente fornendo il nome del server e il nome del database.

E' possibile specificare la tabella da aprire oppure la query SQL.

Argomenti aggiuntivi:

- `dsname` il *Database Source Name (DSN)* registrato sul computer
- `:uid => login` il nome dell'account per l'accesso al database
- `:pwd => password` la password per l'accesso al database
- `:server => server_name` il nome del server (se non usato *dsname*)
- `:db => database_name` il nome del database (se non usato *dsname*)
- `:table => table_name` il nome della tabella (se non usato *sql*)
- `:sql => sql_string` la stringa di istruzioni in linguaggio SQL (se non usato *table*)

Metodi aggiuntivi:

- `sqlserver.config` permette di configurare i parametri di accesso al database per non ripeterli successivamente

```

1  # apre una tabella di un database SQLServer
2  sqlserver.open "nome_DSN", :uid => "login", :pwd => "password", :table => "nome_tabella"
3  sqlserver.open "nome_DSN", :uid => "login", :pwd => "password", :sql => "SELECT * FROM nome_tabe
4
5  # configura l'accesso a un database SQLServer
6  sqlserver.config(
7    :server => 'PCNAME\SQLSERVER',
8    :db     => 'DATABASE',
9    :uid    => 'login',
10   :pwd    => 'password'
11 )
12
13 # esegue una query su una tabella di un database SQLServer
14 debug sqlserver.query(:sql => "select InterviewNumber ResponseCode Status DisplayField1 from Sam

```

<<

9.2.14 Nipo ODIN

IN TEST (Attenzione: la procedura è in fase di test)

odin gestisce dati in file ASCII in formato fisso generato da un questionario NIPO ODIN.

Oltre al file di dati, deve essere presente anche il file contenente la descrizione delle variabili in formato *Diana*. Le estensioni predefinite sono `.dat` per i dati e `.var` per i metadati.

Le variabili multiple (*MV) vengono definite come *set* di variabili multiple, complete di titolo, durante la lettura del file di dati, pertanto si sconsiglia l'utilizzo dell'istruzione `sets.find`.

Argomenti aggiuntivi:

- `:nrows =>` # massimo numero di righe da leggere (default -1, tutte).
- `:buffer =>` # massimo blocco di righe da leggere (default 2000). Un numero basso diminuisce l'uso di memoria, ma aumenta il tempo di lettura.
- `:maxcols =>` # limita le variabili caricate alle prime N colonne specificate
- `:nolabels => true` non carica le etichette delle variabili (default false)
- `:skiptek => true` non legge le variabili alfanumeriche (default true)
- `:var => 'filename'` nome del file VAR. Se non specificato viene usato lo stesso nome del file di dati

E' necessario aver installato la libreria R `ffbase`.

```

1  odin.open "survey.dat"
2  odin.open "survey.dat", :var => "survey01" :buffer => 50

```

<<

9.2.15 Etichette

Importando dati *Excel*, *Access*, *ODBC*, *csv* o di *testo*, le variabili sono prive di etichette, pertanto dopo l'importazione devono essere etichettate. Le variabili categoriali rappresentate da codici numeriche⁴ devono essere convertite in *factor*.

```

1  excel.open "nome_file_dati", :sheet => "nome_foglio_dati"
2
3  label :d6_1 => "Appuntamento irrinunciabile",
4        :d6_2 => "Evadere dalla routine quotidiana",
5        :d6_3 => "Imparare nuove cose"

```

⁴ Non è opportuno utilizzare dati contenenti le etichette al posto dei codici, in quanto i livelli delle variabili *factor* create automaticamente dalla procedura di importazione verrebbero ordinati alfabeticamente.

```

6
7 factor :sesso => ["SESSO", {1 => "M", 2 => "F"}],
8         :area => ["AREA GEOGRAFICA", ["NO", "NE", "C", "S"]],
9         :d1 => ["Dom 1", ["A", "B", "C", "D"]],
10        :d5 => ["Dom 5", ["A", "B", "C", "D", "E"]]

```

In alternativa è possibile utilizzare una tabella di etichette che può essere contenuta nello stesso file Excel o nello stesso database che contiene i dati oppure in un file o database differente.

La tabella delle etichette deve essere nel seguente formato:

name	code	label
d1		Etichetta variabile d1
d1	1	Etichetta codice 1 variabile d1
d2	2	Etichetta codice 2 variabile d1
d2	0	Etichetta variabile d2
d2	1	Etichetta dice 1 variabile d2
...

In corrispondenza dell'etichetta della variabile, il campo `code` può essere lasciato vuoto o posto uguale a zero.

La tabella può contenere anche solo le etichette di un sottoinsieme di variabili o specificare le etichette della variabile e non quella dei suoi livelli o viceversa. I livelli di una singola variabile devono essere invece specificati in modo completo. Un eventuale livello mancante determina lo sfasamento delle etichette dei livelli successivi.

```

1 excel.open "nome_file", :sheet => "nome_foglio_dati", :labels => "nome_foglio_etichette"
2 access.open "nome_file", :table => "nome_tabella_dati", :labels => "nome_tabella_etichette"
3 odbc.open "nome_DSN", :uid => "login", :pwd => "password", :sql => "SELECT * FROM nome_tabella",

```

Se le etichette sono in un file differente, utilizzare il metodo `labels` dell'oggetto sorgente di dati.

```

1 excel.open "nome_file_dati", :sheet => "nome_foglio_dati"
2 excel.labels "nome_file", :sheet => "nome_foglio_etichette"
3
4 access.open "nome_file_dati", :table => "nome_tabella_dati"
5 access.labels "nome_file_dati", :table => "nome_tabella_etichette"

```

Usando file separati, le etichette possono risiedere anche in un file di formato diverso.⁵ `text` e `sss` non hanno un metodo `labels`.

```

1 fwf.open "nome_file", :data => %w(ser_no 1-4 area 5 d1 15 d5 17 cognome 60-69 :a)
2 csv.labels "nome_file"
3
4 excel.open "nome_file_dati", :sheet => "nome_foglio_dati"
5 spss.labels "nome_file"

```

<<

Note

⁵ Una tabella di etichette può essere utilizzata anche per un file *SPSS* in sostituzione oppure a integrazione o correzione delle etichette del system file *SPSS*.

Hole count

10.1 hc

Produce nel log le distribuzioni di frequenza e le medie delle variabili. E' anche possibile listare le prime e le ultime n osservazioni.

Se è attiva una variabile di ponderazione (`weight.by`) vengono calcolate sia le frequenze non pesate che pesate. Per utilizzare una variabile di ponderazione diversa da quella attiva, passare esplicitamente la variabile peso con `:w => :nomevar`

Per le variabili numeriche (non factor) vengono prodotte le distribuzioni di frequenza oltre ai valori riassuntivi, solo se il numero di valori è inferiore o uguale al parametro di configurazione `:hc_maxlev`. Oltre a modificare globalmente il parametro `:hc_maxlev` è possibile anche specificare un diverso numero di valori con `:maxlev => 100`.

L'output finisce nel log di pTabs, non nel report.

```

1  hc :area, :sesso, :eta, :titolo, :prof
2  hc :d1, :d2
3  hc 5, :d1, :d2           # => lista anche le prime 5 e le ultime 5 osservazioni
4  hc 5, :sesso, :area, :w => :espuni # => frequenze ponderate su espuni
5  hc :etaq, :maxlev => 10    # => limita a 10 la soglia per la generazione delle
6                             # distribuzioni di frequenza
7  hc allvars              # hole count per tutte le variabili del dataframe

```

Output:

Sesso	Sesso	#	< N.	% >	< W.	% >
Maschio	1	154	51.33	236	51.68	
Femmina	2	146	48.67	221	48.32	
N.	0	300	100.00	457	100.00	

	n missing	unique	Mean
Sample	300.0000	0	2 1.487
Weighted	457.0659	0	2 1.483

Area geografica	Area geografica	#	< N.	% >	< W.	% >
Nord Ovest	1	93	31.00	140	30.69	
Nord Est	2	50	16.67	78	16.97	
Centro	3	70	23.33	105	23.06	
Sud	4	87	29.00	134	29.29	
N.	0	300	100.00	457	100.00	

	n missing	unique	Mean
Sample	300.0000	0	4 2.503
Weighted	457.0659	0	4 2.509

```

      peso sesso area
1  1.675156      1      1
2  1.751057      2      1
3  1.168366      1      1
4  1.761970      2      1
5  1.823958      2      1
296 1.869170      1      2
297 1.797027      1      2
298 1.981697      1      2
299 1.206531      1      2
300 1.712199      1      2

```

etaq Età

	n	missing	unique	Mean	X.05	X.10	X.25	X.50	X.75	X.90	X.95
Sample	300	0	21	24.78	14.95	15	21	25	29	32.1	34
Weighted	300	0	21	25.81	15.00	18	23	26	29	32.0	3

10.2 hcm

Produce nel log le distribuzioni di frequenza di un set di variabili multiple. E' anche possibile listare le prime e le ultime *n* osservazioni. Se è attiva una variabile di ponderazione (*weight.by*) vengono calcolate sia le frequenze non pesate che pesate. Per utilizzare una variabile di ponderazione diversa da quella attiva, passare esplicitamente la variabile peso con *:w => :nomevar* L'output finisce nel log di pTabs, non nel report.

```
1 hcm 5, s(:d7_,10)
```

Output:

```

d7_1 ... d7_10
# < N. % > < W. % >
Vivere intense emozioni 1 220 78.57 331 77.44
Qualcosa che non si puo' rimandare 2 129 46.07 197 46.12
Occasione per conoscere altre realta' 3 178 63.57 270 63.18
Momento piacevole con la famiglia 4 127 45.36 196 45.74
Puro divertimento, svago 5 222 79.29 339 79.16
Entrare in una dimensione di finzione e irrealta' 6 207 73.93 317 74.05
Stare in compagnia degli amici 7 226 80.71 344 80.37
Argomento di cui parlare con gli altri 8 222 79.29 336 78.57
Modo per rilassarmi 9 230 82.14 349 81.46
Dimenticarmi del resto del mondo 10 180 64.29 271 63.36
N. 0 280 100.00 428 100.00

```

	N.	Responses	Mean responses
Sample	257	1941	7.55
Weighted	391	2950	7.55

```

      peso d7_1 d7_2 d7_3 d7_4 d7_5 d7_6 d7_7 d7_8 d7_9 d7_10
1  1.675156      1      1      1      0      1      1      1      1      1      1
2  1.751057      0      0      1      0      1      1      0      1      0      1
3  1.168366      1      0      0      1      1      1      1      1      1      1
4  1.761970      1      1      0      0      1      1      1      1      1      0
5  1.823958      1      1      1      0      1      0      1      1      1      1
296 1.869170      1      1      1      1      1      1      1      1      1      1
297 1.797027      1      1      1      1      1      0      0      0      1      1
298 1.981697      1      1      1      1      1      1      1      1      1      0
299 1.206531      1      0      1      0      0      0      0      1      1      0
300 1.712199      1      0      1      0      1      1      1      0      1      1

```

11.1 topline

Produce un report HTML con le distribuzioni di frequenza e le principali statistiche delle variabili del dataframe. Se è attiva una variabile di ponderazione (`weight.by`) vengono calcolate le statistiche sia sui dati non pesati che pesati.

« 10	d2		N.	no vado come prima	si vado più di prima	si vado meno di prima	non indica	Missing					
		#		1	2	3	4						
		N.	300	180	23	97	0	0					
		%	100.0	60.0	7.67	32.33	0.0						
« 11	d5		N.	Non risponde	Per niente d'accordo	Poco d'accordo	Abbastanza d'accordo	Molto d'accordo	Missing				
		#		1	2	3	4	5					
		N.	270	20	16	83	95	56	30				
		%	100.0	7.41	5.93	30.74	35.19	20.74					
« 12	d6_1		N.	Molto	Abbastanza	Poco	Per nulla	Missing					
		#		1	2	3	4						
		N.	300	22	104	114	60	0					
		%	100.0	7.33	34.67	38.0	20.0						
« 13	d6_2		N.	300	6	50	173	71	0				
		%	100.0	2.0	16.67	57.67	23.67						
« 14	d6_3		N.	300	17	82	159	42	0				
		%	100.0	5.67	27.33	53.0	14.0						
« 15	d6_4		N.	300	17	74	141	68	0				
		%	100.0	5.67	24.67	47.0	22.67						
« 16	d6_5		N.	300	58	87	124	31	0				
		%	100.0	19.33	29.0	41.33	10.33						
« 17	d6_6		N.	300	40	82	145	33	0				
		%	100.0	13.33	27.33	48.33	11.0						
« 18	d6_7		N.	300	62	71	117	50	0				
		%	100.0	20.67	23.67	39.0	16.67						
« 19	d6_8		N.	300	6	52	151	91	0				
		%	100.0	2.0	17.33	50.33	30.33						
« 20	d6_9		N.	300	30	96	129	45	0				
		%	100.0	10.0	32.0	43.0	15.0						
« 21	d6_10		N.	300	46	71	131	52	0				
		%	100.0	15.33	23.67	43.67	17.33						
« 22	d7		N.	Vivere intense emozioni	Qualcosa che non si può rimandare	Occasione per conoscere altre realtà	Momento piacevole con la famiglia	Puro divertimento, svago	Entrare in una dimensione di finzione e irrealtà	Stare in compagnia degli amici	Argomento di cui parlare con gli altri	Modo per rilassarmi	Dimenticar del resto del mondo
		#		d7_1	d7_2	d7_3	d7_4	d7_5	d7_6	d7_7	d7_8	d7_9	d7_10
		N.	280	220	129	178	127	222	207	226	222	230	180
		%	100.0	78.57	46.57	63.57	45.36	79.29	73.93	80.71	79.29	82.14	64.29

Parametri:

- `varlist`: l'elenco di variabili da tabulare. Se non viene specificata nessuna variabile, vengono prodotti i top-line di tutte le variabili
- `:drop`: `varlist` esclude le variabili specificate
- `:index`: `:none` | `:short` | `:long` l'indice delle tabelle
- `:maxlev`: il numero massimo di livelli di una variabile per produrre le distribuzioni di frequenza
- `:file`: specifica il nome del file di output
- `:hlbase`: `true` | `false` evidenzia i cambiamenti di base

- **:notitle**: true|false include/esclude le variabili senza etichetta
- **:nodata**: true|false include/esclude le variabili senza dati
- **:empty**: true|false stampa le tabelle senza dati

Le impostazione di default dei parametri sono definite nei *Parametri di configurazione*.

Prima del comando *topline*, può essere utile eseguire la procedura di riconoscimento dei gruppi di variabili: *sets.find*. In questo modo *topline* tabulerà automaticamente le variabili multiple in un'unica tabella.

```
1 topline
2
3 topline :drop => [:intnr, :inttime, :srcnt, "v999.1"]
4
5 sets.find
6 topline :v1, :v2, :v3, :v4, :v5, :v6, m_v7, :index => :long, :file => "top_line"
```

Sono disponibili due *callback*, *skip_variable* e *skip_summary*, per valutare se escludere o meno una variabile/set di variabili dal *topline*.

Pannelli

I pannelli sono attivi per tutte le tabelle successive, finché non vengono modificati. `_t` è una variabile gestita dal programma per generare la colonna del totale. E' il totale della variabile posta in fiancata.

Se non è stato definito nessun pannello, `pTabs` ne genera uno automaticamente con il solo totale.

12.1 panels.add

Aggiunge un nuovo pannello al set di pannelli esistenti.

```
panels.add :_t, :area, :ampc           # pannello 1
panels.add :_t, :sesso :eta, :titolo   # pannello 2
panels.add :_t, 'p1.1', 'p2.1', 'p3.1' # pannello 3
```

12.2 panels.drop

Elimina un pannello.

```
panels.drop 1           # i pannelli ora sono due numerati 1 e 2
```

12.3 panel(n).sub

Sostituisce un pannello esistente.

```
panel(2).sub :_t, :x1, :x2
```

12.4 panel(n).add

Aggiunge una o più variabili a un pannello.

```
panel(2).add :q100, :q200
```

12.5 panel(n).drop

Elimina una o più variabili da un pannello.

```
panel(2).drop :x1
```

12.6 panel(n).rowbase

Imposta la base per il calcolo delle percentuali di riga. Per ciascuna variabile del pannello è possibile impostare la variabile e la colonna che deve fare da base per il calcolo della percentuale di riga.

E' possibile indicare come base solo variabili che nel pannello precedono la variabile di cui si vuole specificare la base.

```
# pannello con totale, due colonne per ciascun anno, i mesi dell'ultimo anno
panels.add :_t, :years, :months_last_year
panel[1].rowbase :months_last_year => {:years => 2}
# oppure, in modo più generico:
panel[1].rowbase :months_last_year => {:years => get_max_level(:years)}

# :eta e :prof vengono sdoppiate in base al sesso
split_vars :sesso => [:eta, :prof], :names => "##_s#"
label :eta_s1 => "Età maschi", :prof_s1 => "Professione maschi", :eta_s2 => "Età femmine", :prof_

# si memorizza un riferimento al pannello nella variabile 'pa'
pa = panels.add :_t, :sesso, :eta_s1, :prof_s1, :eta_s2, :prof_s2
pa.rowbase [:eta_s1, :prof_s1] => {:sesso => 1}
pa.rowbase [:eta_s2, :prof_s2] => {:sesso => 2}
# oppure, in una sola riga
pa.rowbase [:eta_s1, :prof_s1] => {:sesso => 1}, [:eta_s2, :prof_s2] => {:sesso => 2}

# annulla la specifica delle basi per le variabili indicate
panel[1].rowbase [:eta_s1, :prof_s1] => nil

# annulla tutte le specifiche delle basi del pannello
p.rowbase
```

12.7 panels.save

Salva una copia dei pannelli per usarli successivamente. Utile quando è necessario modificare o sostituire temporaneamente i pannelli.

```
panels.save
```

12.8 panels.switch

Salva una copia dei pannelli e li sostituisce con un pannello differente.

```
panels.switch :_t, :t100, :t200, :t300
```

12.9 panels.restore

Ripristina i pannelli precedentemente salvati.

```
panels.restore
```

12.10 panels.clear

Elimina i pannelli attivi.

```
panels.clear
```

12.11 join_col_labels

Se il parametro `join_col_labels` è impostato a `true`, quando variabili consecutive del pannello hanno la stessa *variable label*, l'etichetta della variabile viene inserita una sola volta come se le diverse modalità delle variabili appartenessero a un'unica variabile.

Tabelle

Le tabelle vengono generate attraverso funzioni differenti, ciascuna specializzata in un tipo specifico di tabella.

Funzione	Tipo di tabella
cat (vars, title, options, &block)	Distribuzione di frequenze variabile categoriale
xcat (vars, title, options, &block)	Tabelle <i>cat</i> di un set di variabili
md (vars, title, options, &block)	Distibuzioni di frequenze di un set di variabili multidicotomy
mc (vars, title, options &block)	Distibuzioni di frequenze di un set di variabili multicategory
scale (vars, title, options, &block)	Statistiche descrittive di un set di variabili
ord (vars, title, options, &block)	Distribuzione di frequenze + statistiche descrittive (media, deviazione standard, ecc.)
xord (vars, title, options, &block)	Tabelle <i>ord</i> di un set di variabili
sord (vars, title, options)	Tabella <i>scale</i> di un set di variabili + singole tabelle <i>ord</i>
ords (vars, title, options)	Tabelle <i>ord</i> di un set di variabili + <i>scale</i>
tbox (vars, title, options, &block)	Top-box di un set di variabili
grid (vars, title, options, &block)	Distibuzioni di frequenze a totale di variabili singole o multiple a confronto. Una colonnna per variabile
camp (title, options)	Distibuzioni di frequenze a totale delle variabili dei pannelli
cbyc (title, options)	Distibuzioni di frequenze variabili dei pannelli incrociate per le stesse variabili
vjoin (title, options)	Unisce verticalmente più tabelle
hjoin (title, options)	Unisce orizzontalmente più tabelle
htjoin (title, header, options)	Unisce orizzontalmente più tabelle impostando solo il totale e un'etichetta per le colonne

Se si omette il titolo in `cat` e `ord`, viene utilizzata la label della variabile, altrimenti viene lasciato vuoto.

I nomi delle variabili possono essere passati come stringhe (racchiusi da virgolette singole o doppie) o come simboli ¹. Liste di più variabili devono essere racchiuse tra parentesi quadre o devono comunque essere dei vettori ². La funzione `s` è utile per generare sequenze di variabili anche complesse.

```

1  cat 'd1'
2  ord :d5, "D5. Titolo"
3  md [:d7_1, :d7_2, :d7_3, :d7_4, :d7_5, :d7_6, :d7_7, :d7_8, :d7_9, :d7_10]
4  md s(:d7_, 10)           # => equivale all'istruzione della riga precedente
5  scale s([:v, '.1'], 101..104, 199) # => scale [v101.1, v102.1, v103.1, v104.1, v199.1]
```

¹ Un **simbolo** rappresenta semplicemente una stringa ed è costituito dalla stringa di caratteri (il primo non numerico) preceduto dai due punti: `:area`. Il punto `''` non può essere utilizzato all'interno di un simbolo.

² Le parentesi quadre sono la sintassi base per costruire un **Array** (vettore), ovvero: `[elemento1, elemento2, ...]`. In generale può essere usata qualsiasi funzione che restituisce un *array*. La funzione `s` o `seq` di *pTabs* restituisce un *array*.

Le variabili utilizzate in `cat`, `ord` e nelle `grid` (con variabili singole), per le ditribuzioni di frequenze, devono essere *factor*.

Ulteriori opzioni possono essere passate alla funzione all'interno di un blocco, secondo la sintassi:

```
1 cat ... do
2   # istruzioni interne al blocco
3   ...
4 end
```

Varie altre personalizzazioni delle tabelle si ottengono modificando i *Parametri di configurazione*.

Un underscore (`_`) davanti al nome dell'istruzione commenta l'istruzione e tutto il blocco corrispondente.

```
1 # questa tabella non viene generata
2 _vjoin "D1 + D5" do
3   cat :d1 do
4     titles "D1"
5   end
6   ord :d5 do
7     w nil, -10, -5, 5, 10
8     titles "D5"
9   end
10 end
```

Un punto interrogativo (?) dopo il nome dell'istruzione calcola la tavola senza stamparla nel report.

```
1 cat? :d1
```

13.1 autotab

`autotab` produce automaticamente tutte le tabelle per un elenco di variabili o, se non specificato, per tutte le variabili del file di dati.

`autotab` accetta un elenco di variabili e il parametro `:drop` con l'elenco di variabili da escludere.

Se definiti, `autotab` riconosce i *set di variabili* sia di variabili multiple che di variabili singole.

La parte principale della procedura è contenuta nel *callback auto_tab* che può essere ridefinito per personalizzare l'implementazione.

```
1 # produce le tabelle con il pannello per tutte le variabili del dataset
2 panels.add :_t, :sesso, :eta, :area
3 sets.find_all # i set trovati verranno usati da autotab
4 m_d7.title "Titolo per il set delle d7_#"
5 autotab
```

13.2 cat

E' possibile ottenere una tabella con le distribuzioni di frequenza di più variabili. In questo caso è utile attivare la stampa delle etichette delle variabili (`varlabel!`).

```
1 varlabel!
2 cat [:d1, :d2, :d3]
```

13.3 tbox

Nelle tabelle `tbox` le percentuali di colonna vengono calcolate sulla base di ciascuna variabile. Se presenti, i totali di riga si riferiscono alla prima variabile. Può essere opportuno disabilitare i totali di riga con `rtotal_`.

13.4 ord

Per le tabelle `ord` è possibile specificare una o due variabili. Specificando due variabili, sulla prima variabile vengono calcolate le distribuzioni di frequenze, mentre sulla seconda le statistiche descrittive. E' possibile specificare i pesi dei valori della variabile con `weights (o w)`.

```
1 ord :d5
2 ord [:d5, :r5]
3 ord :d5 do
4   w 7, 2.5, 1, 0.5, 0.25
5 end
```

13.5 xord

`xord` produce le singole tabelle con le distribuzioni di frequenza e le statistiche riassuntive (`ord`) per un set di variabili. La sintassi è identica all'istruzione `ord`.

Utilizzare nel titolo i caratteri `##` che verranno sostituito con il *numero dell'item*. Al titolo verrà aggiunta una riga ulteriore con l'etichetta della variabile.

```
1 xord s(:d6_,10), "Tabella 6.## - Quanto sono importanti per Lei questi aspetti?" do
2   w :legend, 4, 3, 2, 1
3   net "POSITIVO" => 1..2, "NEGATIVO" => 3..4
4 end
```

13.6 sord e ords

`sord` e `ords` producono la tabella delle statistiche riassuntive (*scale*) più le singole tabelle con le distribuzioni di frequenza e le statistiche riassuntive (*ord*). `sord` metta la tabella *scale* prima, mentre `ords` la mette dopo *ord*. Nel parametro `options` è possibile specificare:

- `:w`: i pesi da applicare alle variabili (*array*)
- `:sort`: il metodo di ordinamento da applicare alla tabella *scale* (*array*)
- `:net`: i net per le tabelle *ord* (*hash*)
- `:sep`: i separatore del progressivo delle tavole nel titolo; if default è `.`

Utilizzare nel titolo i caratteri `##` che verranno sostituito con il *separator* e il *numero dell'item*. Al titolo verrà aggiunta una riga ulteriore con l'etichetta della variabile.

`sord` non accetta blocchi e sottocomandi.

```
1 sord s(:d6_,10), "Tabella 6## - Quanto sono importanti per Lei questi aspetti?"
2
3 ords s(:d6_,10), "Tabella 6## - Quanto sono importanti per Lei questi aspetti?",
4   :w => [:legend, 4, 3, 2, 1], :sort => [:d, :mean]
5   :net => {"POSITIVO" => 1..2, "NEGATIVO" => 3..4}
```

13.7 md

Nelle tabelle `md` (multidicotomy) viene conteggiato il valore indicato nel parametro di configurazione `md_true_value` (il valore predefinito è 1).

La base dei casi validi è data dai valori validi della prima variabile: tipicamente i casi con valori 0 e 1. Pertanto è necessario assicurarsi che la prima variabile abbia la base corretta, senza badare alla presenza degli zero nelle altre variabili e al fatto che la somma logica di tutte le citazioni corrisponda alla base delle osservazioni.

Se i parametri `nresp` e/o `mresp` sono attivi, in fondo alla tavola viene calcolato anche il numero delle risposte e/o il numero medio delle risposte.

Avvertimento: Se i valori delle variabili SPSS sono etichettati, allora nell'importazione verranno convertite in *factor* e pertanto i livelli partiranno da 1 e non più da 0. Se le variabili non sono numeriche o non sono codificate con 0 e 1, utilizzare le istruzioni `md_tf` o `md_ft` per modificare i parametri `md_true_value` e `md_false_value`.

```
1 md s(:d7_, 10), "md"
2
3 # entrambe le istruzioni:
4 md_ft 1, 2
5 md_tf 2, 1
6 # impostano i valori md_false_value a 1 e md_true_value a 2
```

13.8 mc

Nelle tabelle `mc` (multicategory) la base dei casi validi è data dai valori validi della prima variabile. Pertanto è necessario assicurarsi che tutte le osservazioni che costituiscono la base della tabella, abbiano un valore valido sulla prima variabile.

13.9 grid

Le tabelle `grid` sono di due tipi: *singole* e *multiple*. Il risultato è l'affiancamento delle distribuzioni di frequenza a totale delle varie variabili singole o dei set di variabili multiple.

Nelle `grid` *singole* ogni colonna corrisponde a una variabile e ogni riga a uno dei livelli delle variabili. Le variabili devono avere gli stessi livelli.

Nelle `grid` *multiple* ogni colonna corrisponde a un set di variabili multidicotomy e ogni riga ai diversi item. I set di variabili multiple devono avere la stessa numerosità e avere item corrispondenti.

Le tabelle `grid` hanno bisogno di ulteriori etichette. Con `label` si specifica l'etichetta del gruppo di colonne e con `labels` le etichette delle singole colonne. Nelle `grid` *singole*, se `labels` non è specificato vengono usate le etichette delle variabili. Nelle `grid` *multiple*, le variabili vanno specificate in più gruppi (un vettore di vettori).

Se necessario, impostare nel modo opportuno le percentuali della riga del totale con `trowpct` e `tcolpct`.

```
1 trowpct_
2 tcolpct!
3
4 grid "Test GRID\n(Variabili singole)" do
5   vars s(:d6_,10)
6   label "Domanda 6"
7 end
8
9 grid "Test GRID\n(Variabili multiple)" do
10  vars ss("d9_#1_#2",4,8)
11  label "MARCHE"
12  labels "Marca 1", "Marca 2", "Marca 3", "Marca 4"
13 end
```

E' possibile inserire esplicitamente la matrice di variabili, oppure combinare più funzioni `s`, oppure utilizzare la funzione `ss`, oppure utilizzare `in_groups_of` per suddividere un unico elenco:

```
1 varlist = ['d9_1_1', 'd9_1_2', 'd9_1_3', 'd9_2_1', 'd9_2_2', 'd9_2_3', 'd9_3_1', 'd9_3_2', 'd9_3_3']
2
3 vars ss("d9_#1_#2",4,3)
4 # ...equivale a:
```

```

5 vars s(:d9_1_,1..3), s(:d9_2_,1..3), s(:d9_3_,1..3), s(:d9_4_,1..3)
6 # ...equivale a:
7 vars varlist.in_groups_of(3)
8 # ...equivale a:
9 vars [['d9_1_1', 'd9_1_2', 'd9_1_3'], ['d9_2_1', 'd9_2_2', 'd9_2_3'],
10      ['d9_3_1', 'd9_3_2', 'd9_3_3'], ['d9_4_1', 'd9_4_2', 'd9_4_3']]

```

Schema dell'impiego dei sottocomanti di una grid con variabili multiple:

	MARCHE				
	Marca 1	Marca 2	Marca 3	Marca 4	
Etichetta item 1	d9_1_1	d9_2_1	d9_3_1	d9_4_1	← label "MARCHE"
Etichetta item 2	d9_1_2	d9_2_2	d9_3_2	d9_4_2	← labels "Marca 1", "Marca 2", "Marca 3", "Marca 4"
Etichetta item 3	d9_1_3	d9_2_3	d9_3_3	d9_4_3	← vars ss("d9_#1_#2",4,8)
Etichetta item 4	d9_1_4	d9_2_4	d9_3_4	d9_4_4	ovvero:
Etichetta item 5	d9_1_5	d9_2_5	d9_3_5	d9_4_5	vars s(:d9_1_,1..8), s(:d9_2_,1..8), s(:d9_3_,1..8), s(:d9_4_,1..8)
Etichetta item 6	d9_1_6	d9_2_6	d9_3_6	d9_4_6	ovvero:
Etichetta item 7	d9_1_7	d9_2_7	d9_3_7	d9_4_7	vars [:d9_1_1,:d9_1_2,:d9_1_3,:d9_1_4,:d9_1_5,:d9_1_6,:d9_1_7,:d9_1_8],
Etichetta item 8	d9_1_8	d9_2_8	d9_3_8	d9_4_8	[:d9_2_1,:d9_2_2,:d9_2_3,:d9_2_4,:d9_2_5,:d9_2_6,:d9_2_7,:d9_2_8],
					[:d9_3_1,:d9_3_2,:d9_3_3,:d9_3_4,:d9_3_5,:d9_3_6,:d9_3_7,:d9_3_8],
					[:d9_4_1,:d9_4_2,:d9_4_3,:d9_4_4,:d9_4_5,:d9_4_6,:d9_4_7,:d9_4_8]

13.10 vjoin

Unisce verticalmente (accoda) più tabelle che presentano lo stesso pannello. Eventuali parametri modificati all'interno del corpo di `vjoin` hanno validità locale.

```

1 vjoin "D5.2. Domanda 5 con medie più aggregazione" do
2   total_after!
3   ord :d5 do
4     w nil, -10, -5, 5, 10
5   end
6   cat :r5 do
7     titles "D5.2 aggregata"
8   end
9 end

```

13.11 hjoin

Unisce orizzontalmente (affianca) più tabelle che presentano la stessa fiancata. Eventuali parametri modificati all'interno del corpo di `hjoin` hanno validità locale.

```

1 hjoin "D6_1 vs D6_2 vs D6_3" do
2   ord :d6_1
3   ord :d6_2
4   ord :d6_3
5 end

```

13.12 htjoin

Unisce orizzontalmente (affianca) più tabelle che presentano la stessa fiancata, imposta il pannello al solo totale e aggiunge un'etichetta alle colonne. Genera tabelle grid per qualsiasi tipo di tabella.

```

1 htjoin "D6. Domanda 6 con item affiancati in colonna", "AFFERMAZIONI" do
2   s(:d6_,10).rep do |v|
3     ord v
4   end
5 end

```

13.13 Sottocomandi delle tabelle

13.13.1 id

E' possibile assegnare alla tavola un *identificativo* univoco all'interno del report.³ Deve essere una stringa o un simbolo, deve iniziare con una lettera e può contenere solo lettere, numeri e underscore. Deve avere un formato diverso da `t+numero`.

```
1  cat :d1, "Titolo" do
2    id :dom_1
3  end
```

E' possibile specificare l'id anche nell'argomento *options* del metodo:

```
1  cat :d1, "Titolo", :id => :dom_1
```

13.13.2 vars e title

I parametri `vars` e `title` possono anche essere sostituiti dalle funzioni con lo stesso nome all'interno del blocco di codice della tabella. In tal caso, in `vars`, non è necessario racchiudere le liste di variabili tra parentesi quadre. Vedi anche `$var_label()`.

```
1  md do
2    vars :d7_1, :d7_2, :d7_3, :d7_4, :d7_5
3    title "D7. Titolo"
4  end
```

13.13.3 header

Inserisce un'etichetta immediatamente sotto il pannello. Nel caso di unioni di tabelle, le istruzioni `header` delle sottotabelle vengono ignorate.

```
1  cat :d1, "Tab. 1"
2    header "Domanda D1"
3  end
4
5  vjoin "Tab. 2" do
6    header "Domanda D1+D5"
7    cat :d1
8    cat :d5
9  end
```

13.13.4 titles

Inserisce delle etichette esplicite per le variabili in fiancata. Usare una stringa vuota o `nil` se non si vuole etichettare una variabile.

```
1  cat [:d1, :d2] do
2    titles "Etichetta D1", "Etichetta D2"
3  end
4  cat [:d1, :d2] do
5    titles nil, "Etichetta D2"
6  end
```

³ Gli **ID** delle tavole vengono utilizzati in alcuni formati di output (cfr. *render.matrix*).

13.13.5 note, footnote, corner

Aggiungono delle note di testo alla tabella:

- `note`: nota della sezione
- `footnote`: nota della tabella
- `corner`: nota nell'angolo in alto a sinistra della tabella

```
1 md s(:d7_, 10), "D7. Multipla + sort" do
2   sort :d
3   note "Dati ordinati per frequenze decrescenti"
4 end
```

13.13.6 w

Per `ord` o `scale`: assegna dei pesi ai valori delle variabili. Indicando `nil`, `:na` oppure `NA`, la modalità viene esclusa dal calcolo.

I pesi vengono assegnati dal primo all'ultimo ai livelli della variabile partendo dal livello 1.

Inserendo l'argomento `:legend`, verrà generata una nota automatica che descrive i pesi. Il testo della legenda è determinato dal parametro di configurazione `weight_legend_label`. Il simbolo inserito al posto di un peso `nil` è determinato dal parametro di configurazione `nil_weight_note`.

Nelle tabelle `ord`, per aggiungere alle etichette delle righe l'indicazione del peso, inserire una stringa che funge da template, dove il carattere `#` inserisce il peso, mentre il carattere `*` inserisce l'etichetta. Il simbolo inserito al posto di un peso `nil` è determinato dal parametro di configurazione `nil_weight_label`.

```
1 ord :d5, "D5. Distribuzione di frequenza più valori medi con pesi" do
2   w 7, 2.5, 1, 0.5, 0.25
3 end
4
5 scale s(:x9,10), "Tabella 9." do
6   w (1..10).to_a << :na # <= ovvero: 1,2,3,4,5,6,7,8,9,10,nil
7 end
8
9 # aggiunge una legenda nel piè di pagina
10 ord :d1 do
11   w :legend, :na, 5, 4, 3, 2, 1
12 end
13
14 # aggiunge l'indicazione del peso nell'etichetta:
15 # ( 5 ) Molto
16 ord :d1 do
17   w "( # ) *", nil, 5, 4, 3, 2, 1
18 end
19
20 # aggiunge l'indicazione del peso nell'etichetta:
21 # Molto [w:5]
22 # aggiunge anche la legenda
23 ord :d1 do
24   w :na, 5, 4, 3, 2, 1, "* [w:#]", :legend
25 end
```

13.13.7 wby

`wby` permette di specificare la/le variabile/i di ponderazione per una specifica tabella.

```
1 cat :d1 do
2   wby :peso2
3 end
```

13.13.8 codes

Nelle tabelle `tbox` specifica i codici che devono essere raggruppati e conteggiati. Può essere specificato come sottocomando o nelle opzioni.

```
1  tbox s(:d6_,10), "Tabella 6 - Quanto sono importanti per Lei questi aspetti? (TOP BOX)" do
2    codes 1,2
3    sort :d
4  end
5
6  tbox s(:d6_,10), "Tabella 6 - Quanto sono importanti per Lei questi aspetti? (TOP BOX)", :codes :
```

13.13.9 missed

Nelle tabelle `cat`, `ord` e `scale`, specifica i livelli delle variabili che devono essere considerati mancanti. Agisce escludendo i livelli specificati per ciascuna variabile della tabella. Attenzione: i livelli delle variabili vengono riscalati partendo da 1.

Inserendo l'argomento `:note`, verrà generata una nota automatica che riporta le etichette degli item esclusi (utilizza le etichette della prima variabile). Il testo della nota è determinato dal parametro di configurazione `note_missed`.

Attenzione: i livelli vengono esclusi dalla tabella prima di qualsiasi altra operazione, pertanto nelle istruzioni `w`, `net`, `hide`, ecc. bisogna fare riferimento ai livelli restanti.

```
1  cat :v10 do
2    missed 8, 9
3  end
4
5  # escludo il valore centrale di ciascuna variabile e assegno i pesi ai restanti valori
6  scale s(:v26_,10) do
7    missed 3, :note
8    w :legend, 4, 3, 2, 1
9  end
```

13.13.10 omitted

Nelle tabelle `cat` e `ord`, specifica i livelli della variabile che devono essere esclusi dalla tabella. Agisce come un filtro dei dati sui livelli specificati della prima variabile della tabella.

Inserendo l'argomento `:note`, verrà generata una nota automatica che riporta il numero di casi mancanti e le etichette degli item. Il testo della nota è determinato dal parametro di configurazione `note_omitted`.

Attenzione: i livelli vengono esclusi dalla tabella prima di qualsiasi altra operazione, pertanto nelle istruzioni `w`, `net`, `hide`, ecc. bisogna fare riferimento ai livelli restanti.

```
1  cat :v10 do
2    omitted 8, 9, :note
3  end
4
5  # il filtro è applicato sui dati in base alla prima variabile, quindi anche la seconda variabile
6  ord [:eta_cla, :eta_real] do
7    omitted 1, :note
8  end
9
10 ord :d5 do
11   omitted 3
12   w :legend, 4, 3, 2, 1
13 end
```

13.13.11 sort

Ordina le righe secondo le numerosità delle frequenze. `:a` ascendente, `:d` discendente. Come secondo parametro si può specificare la statistica su cui calcolare l'ordinamento oppure `:label` per avere l'ordinamento sulle etichette dei livelli: il default è `:count`. Specificando `:rev`, le righe vengono ordinate in senso inverso.

```

1 md s(:d7_, 10), "D7. Multipla + sort" do
2   sort :d
3   note "Item ordinati per frequenze decrescenti"
4 end
5 md s(:d7_, 10), "D7. Multipla + sort" do
6   sort :a, :label
7   note "Item ordinati alfabeticamente"
8 end
9 scale s(:d6_, 10) do
10  sort :a, :mean
11  note "Item ordinati sui valori medi"
12 end
13 cat d1, "D1" do
14  sort :rev
15  note "Item ordinati in senso inverso"
16 end

```

13.13.12 fixed

In caso di ordinamento, specifica gli item che devono rimanere in fondo. Specificare *un numero negativo* indicante il numero di item a partire dall'ultimo che devono essere tenuti fissi. Tenere eventualmente conto dell'item aggiunto dal sottocomando `none`.

```

1 md s(:d7_, 1..10, 98, 99), "D7. Multipla" do
2   sort :a
3   fixed -2
4   noresp -2           # gli item 11 e 12: d7_98 e d7_99
5 end
6
7 md s(:d7_, 10), "D7. Multipla" do
8   none "Nessuna citazione"
9   sort :a
10  fixed -1           # l'item 11, quello generato da 'none'
11 end

```

13.13.13 down, up

In caso di ordinamento, specifica gli item che devono rimanere in fondo (`down`) o in cima (`up`). Specificare *l'elenco degli item* che devono essere tenuti in fondo/in cima. Nel caso di una variabile singola indicare i codici dei livelli. Nel caso di variabili multidicotomy indicare il numero d'ordine della variabile rispetto all'elenco di variabili passato al comando `md`. L'item generato da una eventuale istruzione `none` va considerato come una variabile aggiuntiva.

```

1 md s(:d7_, 10, 20), "D7. Multipla" do
2   sort :a
3   up 11
4 end
5
6 md s(:d7_, 10, 20), "D7. Multipla" do
7   none "Nessuna citazione"
8   sort :a
9   down 11, 12
10 end

```

13.13.14 keep e drop

Seleziona le righe da mostrare in una tabella, elencando le righe da tenere oppure quelle da eliminare. *keep* e *drop* agiscono dopo il calcolo della tabella, quindi la base e il calcolo delle percentuali rimane invariato. Agiscono però prima di *sort*, quindi si riferiscono sempre all'ordine originario dei livelli.

Passando a *keep* gli indici in un ordine non sequenziale, le righe vengono riordinate secondo quella sequenza. Omettendo o duplicando degli indici le relative righe vengono omesse o duplicate.

```
1 md s(:d7_, 10) do
2   keep 4..6, 10
3 end
4
5 cat :d1 do
6   drop 6
7 end
```

13.13.15 first e last

Mostra solo le prime o le ultime *n* righe della tabella. Per esempio in una tabella con variabili multiple, in combinazione con *sort :d*, è possibile mostrare solo le prime citazioni. *first* e *last* agiscono dopo il calcolo della tabella, quindi la base e il calcolo delle percentuali rimane invariato. Inoltre agiscono dopo eventuali ordinamenti (*sort*).

```
1 md s(:d7_, 10), "D7. Multipla, prime 3 citazioni" do
2   sort :d
3   first 3
4 end
5
6 md s(:d7_, 10) do
7   last 4
8 end
```

13.13.16 hide

Elimina le righe della tabella finale. *hide* agisce al momento della stampa della tabella, dopo che sono state applicate tutte le precedenti trasformazioni (*net*, *keep*, *drop*, *sort*, *first*, *last*, ecc.).

```
1 cat :v200 do
2   hide 1..4
3 end
4
5 ord :v101 do
6   net "GIUDIZIO NEGATIVO" => 1..5, "GIUDIZIO POSITIVO" => 6..10, "NON INDICA" => 11
7   hide 14
8 end
```

13.13.17 no_panel

Elimina il pannello dalle *hjoin* con solo il totale e aggiunge un'etichetta alle colonne.

```
1 panels.switch :_t
2 hjoin "D6. Domanda 6 con item affiancati in colonna" do
3   no_panel "AFFERMAZIONI"
4   s(:d6_, 10).rep do |v|
5     ord v
6   end
7 end
8 panels.restore
```

13.13.18 base

Nelle md permette di specificare esplicitamente i casi su cui calcolare le percentuali. *base* utilizza la stessa sintassi delle espressioni specificabili in *filter*.

```
1 md s(:d7_, 10), "Base delle percentuali calcolata sul primo item"
2 md s(:d7_, 10), "Base delle percentuali calcolata sul secondo item" do
3   base :d7_2
4 end
```

13.13.19 noresp

Nelle md dichiara le variabili che non devono essere conteggiate nel numero medio delle risposte. E' possibile indicare i nomi delle variabili oppure un numero positivo (numero di variabili a partire dalla prima) o negativo (numero di variabili a partire dall'ultima).

```
1 md s(:d7_, 10, 98, 99), "D7. Multipla" do
2   noresp :d7_98, :d7_99
3 end
4
5 md s(:d7_, 10, 98, 99), "D7. Multipla" do
6   noresp -2
7 end
```

13.13.20 none

Nelle tabelle *cat* e *ord*, quando è attivato *base_pct :all*, aggiunge il conteggio delle osservazioni che non hanno risposta sulla variabile.

```
1 base_pct :all
2 ord :d5, "D5. Titolo" do
3   w :legend, 5, 4, 3, 2, 1
4   none "(non pertinente)"
5 end
```

Nelle md aggiunge il conteggio delle osservazioni che non hanno attivato nessun item. Questa voce non viene conteggiata nel numero medio delle risposte.

none conteggia i casi che non hanno attivato nessuna citazione nel set di variabili. Vengono conteggiati solo i record che hanno valore valido sulla prima variabile o quelli della base definita con il sottocomando **base**.

Il valore utilizzato per considerare un item attivato e il valore generato nella nuova variabile è quello definito dalle istruzioni *md_ft* o *md_tf* (default 1).

```
1 md s(:d7_, 10), "D7. Multipla" do
2   none "Nessuna citazione"
3 end
```

13.13.21 all

Nelle md aggiunge il conteggio delle osservazioni che hanno attivato almeno un item. Questa voce non viene conteggiata nel numero medio delle risposte.

Il valore contato e quello generato nella nuova variabile è quello definito dalle istruzioni *md_ft* o *md_tf* (default 1).

```
1 md s(:d7_, 10), "D7. Multipla" do
2   all "TOTALE CITAZIONI"
3 end
```

13.13.22 nresp e mresp

Nelle md, i parametri `nresp` e `mresp` attivano rispettivamente il calcolo del numero delle risposte e del numero medio delle risposte. Per default è attivo `mresp`.

```
1 mresp!
2 nresp!
```

Avvertimento: Eventuali item che conteggiano le osservazioni che non hanno citazioni (per esempio la voce “Nessuna di queste” nella domanda sul ricordo delle marche), se inseriti come semplici variabili verranno conteggiati nel numero delle risposte. Pertanto, se si utilizza `nresp!` e/o `mresp!`, dichiararle tramite il sottocomando `noresp`, oppure usare al loro posto `none` o `net . none`, oppure utilizzare dei nomi di variabile che terminano per `_NA_` o `_na_`: per esempio la variabile `d7_99_na_` verrà automaticamente esclusa dal conteggio.

13.14 base_pct

Il parametro `base_pct` permette di impostare la base per il calcolo delle percentuali di colonna.

- `:table` (il default) calcola le percentuali sui casi validi della variabile di fianco
- `:all` calcola le percentuali sul totale casi o, nel caso sia impostato un filtro, sul totale casi selezionati

```
1 base_pct :all
2 cat :dl
```

13.15 base_multi

Il parametro `base_multi` permette di impostare il metodo di calcolo delle percentuali di colonna nelle multidicotomy.

- `:cases` (il default) calcola le percentuali rispetto ai casi validi
- `:resp` calcola le percentuali sul totale delle risposte
- `:all` calcola le percentuali sul totale casi

13.16 Valori restituiti dalle tabelle

I seguenti metodi restituiscono dati e metadati da una tabella.

Metodo	Restituisce
title	il titolo della tabella
footnote	la nota della tabella
cvars	i nomi delle variabili di pannello di una tabella
rvars (ntable=1)	i nomi delle variabili della fiancata di una tabella
labels (ntable=1)	le etichette della fiancata
header (varname, code, ntable=1)	l'etichetta di una colonna
headers (varname=nil, ntable=1)	le etichette di tutto il pannello o di una variabile di colonna
column (varname, code, stat=:count, ntable=1)	una colonna di dati
columns (varname, stat=:count, ntable=1)	le colonne di dati di una variabile di pannello
row (stat=:count, varname=nil, code=1, ntable=1)	una riga di dati
rows (stat=:count, varname=nil, ntable=1)	le righe di dati della tabella
coord (series=[1,2])	le coordinate (per ca)

Questi metodi vengono prevalentemente usate su una variabile a cui è stata assegnata la tabella.

```
1  dom = ord :d28 do
2    w :legend, 15, 45, 90, 150, 210, 300, 0
3  end
4
5  dom.row(:trowpct, :area)
6  dom.labels
7  dom.headers(:eta)
8  dom.column(:eta, 1, :count)
9  dom.columns(:sesso, :colpct)
10 dom.columns(:_t, [:count, :colpct])
```

Note

14.1 net_pos

Il parametro di configurazione `net_pos` determina la posizione dei net: `:before`, `:after`, `:top`, `:bottom`. Il default è `:before`.

```
1 net_pos :top
```

14.2 netonly

Il parametro di configurazione `netonly` permette di mostrare nella tabella solo i net. Il default è `false`.

```
1 netonly!
```

14.3 netstyle

Di default i net vengono differenziati graficamente dagli altri item. `netstyle` permette di disattivare gli stili attribuiti ai net e il rientro delle voci appartenenti a un net.

```
1 # Gli item 1 e 6 vengono raggruppati in un net, ma poi vengono nascosti e viene
2 # visualizzato solo il net come se fosse un normale item
3 net_pos :bottom
4 netstyle_
5 cat :d5 do
6   net "Non indica" => [1,6]
7   hide [1,6]
8 end
9 netstyle!
```

14.4 Tabelle di risposte singole

Nelle tabelle di variabili singole (`ord` e `cat`) è possibile generare aggregazioni di item su un solo livello.

14.4.1 net

Il sottocomando `net` aggiunge delle aggregazioni delle righe della tabella. I livelli da aggregare vanno indicati utilizzando l'indice della posizione.

L'elenco dei livelli può essere indicato come vettore `[2, 3, 4]` o come range `(2..4)`.

E' possibile anche utilizzare una stringa contenente un'espressione da calcolare sulle righe della tabella. L'espressione verrà calcolata sui valori assoluti delle righe della tabella e sui risultati saranno calcolate le restanti statistiche. Eventuali costanti numeriche devono essere precedute dal carattere cancelletto #. L'espressione può contenere operatori aritmetici "4+5", parentesi "(4+5)-(2+3)", costanti "(4+5)*#2" e funzioni R "sqrt(5+4)".

Il parametro opzionale `:stats` permette di limitare le statistiche da mostrare per i net.

E' possibile specificare più sottocomandi `net` per tabella.

Se i livelli dei net non sono in sequenza o se si utilizzano espressioni, il parametro `net_pos` può essere impostato solo con `:top` o `:bottom`.

```
1 net_pos :after
2 cat :d5, "D5. Singola + NET" do
3   net "NON D'ACCORDO" => (1..3), "D'ACCORDO" => [4,5], :stats => [:rowpct, :colpct]
4   net "D'ACCORDO - NON D'ACCORDO" => "(4+5)-(1+2+3)", :stats => :colpct
5 end
```

14.4.2 net_from_groups

Genera automaticamente i net di una variabile desumendoli dai dati in base a una variabile di raggruppamento.

```
1 cat :modelli do
2   net net_from_groups(:modelli, :marche)
3 end
```

Di default i net vengono costruiti in base all'elenco dei valori (`:as_list`), per utilizzare invece i range, indicare l'opzione `:as_range`.

```
1 debug net_from_groups(:modelli, :marche, :as_list)
2 # <Hash> {"MARCA 1"=>[1, 2, 3], "MARCA 2"=>[4, 5, 6, 7, 8], ...
3
4 debug net_from_groups(:modelli, :marche, :as_range)
5 # <Hash> {"MARCA 1"=>1..3, "MARCA 2"=>4..8, ...
```

Avvertimento: Eventuali livelli senza casi della variabile verranno rimossi (*drop_levels*).

14.5 Tabelle di risposte multiple

Nelle tabelle di variabili multidicotomy (`md`) è possibile generare aggregazioni di item su più livelli (*net* e *subnet*).

14.5.1 nets

nets costruisce un oggetto con la struttura di net desiderata. Va definito prima della tabella. Un net può essere formato sia da semplici variabili, sia da altri net (*subnet*). I net sono a tutti gli effetti delle variabili aggiunte al data frame, pertanto sono disponibili per più tabelle e anche per essere utilizzate singolarmente.

Il valore *contato* è quello definito dalle istruzioni `md_ft` o `md_tf` (default 1).

Il valore *generato* nel net è quello definito dalle istruzioni `md_ft` o `md_tf` (default 1).

Parametri:

- `:name`: la radice del nome da utilizzare per generare le variabili net. Se non specificato viene venerata automaticamente.
- `:code`: il valore da *contare* (`code: codice`) o il valore da *contare* e il valore da *imputare* nel net `code: {codice => nuovo_valore}`.
- `:nocode`: il valore da *contare* (`code: codice`).

Sottocomandi:

- **net**: genera un nuovo net. Deve specificare l'etichetta del net e opzionalmente l'elenco delle variabili. E' possibile specificare un blocco (do) in cui specificare ulteriori subnet e/o variabili
- **var**: inserisce una o più variabili nella posizione indicata
- **none**: un net che conteggia le osservazioni che non hanno attivato nessuna variabile della lista. Deve specificare l'etichetta del net e opzionalmente l'elenco di variabili. Se le variabili non vengono specificate, *none* utilizza tutte le variabili indicate nelle precedenti istruzioni *net* e *var*. Questo net non viene conteggiato nel numero medio delle risposte.

E' possibile indicare il valore da *contare* anche direttamente nei sottocomandi *net* e *none*, indicando il valore prima dell'elenco delle variabili.

E' possibile indicare sia il valore da *contare* che il valore da imputare nel net anche direttamente nel sottocomando *net*, indicando {codice => nuovo_valore} prima dell'elenco delle variabili.

```

1 nome_net = nets(options={}) do
2   net "Etichetta" [, codice][, elenco_variabili] do
3     net "Etichetta", [, codice]elenco_variabili
4     var elenco_variabili
5     net "Etichetta" [, codice][, elenco_variabili] do
6       net ...
7       var ...
8     end
9   end
10  none "NESSUNA CITAZIONE"
11 end
12
13 md nome_net, "Multipla con net"

1 n7 = nets(name: "net_d7", code: 2, nocode: 1) do
2   net "TOT. CITAZIONI", s(:d7_, 3) do
3     net "AMICI/FAMIGLIA", :d7_4, :d7_7, :d7_8
4     net "SVAGO" do
5       net "SVAGO 1", :d7_5, :d7_6
6       net "SVAGO 2", :d7_9, :d7_10
7     end
8     var :d7_11, :d7_12
9   end
10  none "NESSUNA CITAZIONE"
11 end
12 md n7, "Multipla con net"

```

L'oggetto restituito dall'istruzione *nets* (*NetTree*) dispone di metodi che possono essere utilizzati a scopo informativo o in altre istruzioni.

Metodi dell'oggetto NetTree:

- **info**: stampa nel log la struttura dei net
- **netvarlist**: restituisce l'elenco delle nuove variabili create (net)
- **varlist**: restituisce l'elenco delle variabili utilizzate nelle istruzioni *net* e *var*
- **varlist(:all)**: restituisce l'elenco delle variabili utilizzate nelle istruzioni *net*, *var* e *none*
- **to_var**: restituisce l'elenco dei nomi di variabili/net di primo livello. E' il metodo chiamato da **md** per generare la tavola

```

1 n7.info
2 #< (1) [any] level: 1, parent: 0, node: 1, name: n_bffbg9_1, label: "TOT. CITAZIONI", vars: [d7_1
3 #< (2) [any] level: 2, parent: 1, node: 2, name: n_bffbg9_2, label: "AMICI/FAMIGLIA", vars: [d7_4
4 #< (3) [any] level: 2, parent: 1, node: 3, name: n_bffbg9_3, label: "SVAGO", vars: [d7_5, d7_6, d
5 #< (4) [not] level: 1, parent: 0, node: 4, name: n_bffbg9_4, label: "NESSUNA CITAZIONE", vars: [d
6 #< (5) [var] level: 1, parent: 0, vars: [d7_99] >

```

```

7
8 debug n7.netvarlist
9 #<Array> ["n_bffbg9_1", "n_bffbg9_2", "n_bffbg9_3", "n_bffbg9_4"]
10
11 hcm n7.netvarlist

```

14.5.2 Sintassi alternativa

Attenzione: A `net.new` e `net.none` va preferita la nuova sintassi `nets`.

net.new

I net devono essere definiti prima della tabella tramite l'istruzione `net.new`. Un net può essere formato sia da semplici variabili, sia da altri net (*subnet*): l'elenco di variabili di `net.new` può contenere nomi di variabili e nomi di net. I net sono a tutti gli effetti delle variabili aggiunte al data frame, pertanto sono disponibili anche per più tabelle.

Il valore *contato* è quello definito dalle istruzioni `md_ft` o `md_tf` (default 1).

Il valore *generato* nel net è quello definito dalle istruzioni `md_ft` o `md_tf` (default 1).

E' possibile indicare esplicitamente il valore da *contare* direttamente nell'istruzione `net.new`, indicando il valore prima dell'elenco delle variabili.

E' possibile indicare esplicitamente sia il valore da *contare* che il valore da imputare nel net direttamente nell'istruzione `net.new` indicando `{codice => nuovo_valore}` prima dell'elenco delle variabili.

Nell'istruzione `md` dovranno comparire solo i nomi di variabili non appartenenti ai net e i net di più alto livello (non i subnet).

```

1 net.new nome_net, "Etichetta", elenco_variabili/subnet
2 net.new nome_net, "Etichetta", codice, elenco_variabili/subnet
3 net.new nome_net, "Etichetta", {codice => nuovo_codice}, elenco_variabili/subnet
4
5 net.new :n7_11, "AMICI/FAMIGLIA", :d7_4, :d7_7, :d7_8
6 net.new :n7_12, "SVAGO", :d7_5, :d7_6, :d7_9, :d7_10
7 net.new :n7_1, "TOT. CITAZIONI", s(:d7_, 1..3, :n7_11, :n7_12)
8
9 md [:n7_1, :d7_98], "Domanda D7"
10
11 # conta il codice 2
12 net.new :n1, "Etichetta", 2, s(:x7_, 10)
13 # conta il codice 1 e crea un net con il codice 2
14 net.new :n2, "Etichetta", {1 => 2}, s(:x7_, 10)

```

net.none

Genera un net che conteggia le osservazioni che non hanno attivato nessuna variabile/subnet della lista. Questo net non viene conteggiato nel numero medio delle risposte.

Il valore *generato* nel net è quello definito dalle istruzioni `md_ft` o `md_tf` (default 1).

E' possibile indicare esplicitamente il valore da *contare* direttamente nell'istruzione `net.none`, indicando il valore prima dell'elenco delle variabili.

```

1 ...
2 net.none :n7_0, "NESSUNA CITAZIONE", s(:d7_, 1..10)
3
4 md [:n7_1, :n7_0], "Domanda D7"

```

5
6 net.none :n7, "NESSUNA CITAZIONE", 0, s(:d7_,1..10)

Selezione delle osservazioni

15.1 filter

Seleziona un sottoinsieme di casi in modo **temporaneo** (può essere modificato o disattivato).

Una nuova istruzione `filter` sostituisce quella precedente.

- `:na` e `:valid` rappresentano, rispettivamente, i valori mancanti e i valori validi. Per una sola variabile è possibile indicare più sinteticamente `!v1` e `v1`.
- `:label` definisce un'etichetta che verrà stampata in ogni tavola.
- `:id` definisce un identificativo del sottocampione; viene utilizzato in alcuni formati di output.
- `#` il cancelletto datavanti alla variabile nelle espressioni R, converte la variabile in numerica

```

1 # con variabile / valori
2 filter :sesso => 1,      :label => "Intervistati di sesso maschile"
3 filter :d1 => 1
4 filter :d1 => '<3'
5 filter :d1 => :valid, :label => "Osservazioni con valori validi sulla D1"
6 filter :d1,          :label => "Osservazioni con valori validi sulla D1" # equivalente alla pre
7 filter :d1 => :na,   :label => "Osservazioni con valori mancanti sulla D1"
8 filter "!d1",      :label => "Osservazioni con valori mancanti sulla D1" # equivalente alla pre
9 # Con condizioni in OR per vettori e range
10 filter :d1 => [1,2], :d2 => 3..5 # ((d1=1 OR d1=2) AND (d2=3 OR d2=4 OR d2=5))
11
12 # con espressione R
13 filter "sex == 'Maschio'"
14 filter "#sex==1 & etanum>=18", :label => "Intervistati maschi maggiorenni"
15 filter "(#d1==1 | #d1==3) & #sex==1"

```

Suggerimento: Per le espressioni vedi anche *Espressioni*

15.1.1 filter_not

Il suffisso `_not` permette di invertire l'espressione.

```

1 filter_not :sesso => 2, :label => "Intervistati di sesso maschile"

```

15.1.2 and_filter e or_filter

Sono inoltre disponibili i prefissi `and_` e `or_` che permettono di concatenare più filtri sia per specificare un filtro complesso, sia per modificare il filtro tra diverse tavole.

```
1 filter :d1 => [1,2], :label => "D1=1 OR D1=2"
2 and_filter_not :sesso => 1, :label => "(D1=1 OR D1=2) AND NOT SESSO=1"
3 or_filter :d5 => 4, :label => "((D1=1 OR D1=2) AND NOT SESSO=1) OR D5=4"
```

and_filter e *or_filter* non possono essere usati se prima non è stato definito un filtro con l'istruzione *filter*.

Il programma tiene memoria dei filtri aggiunti con *and_* e *or_* e delle relative etichette, pertanto è possibile eliminare gli ultimi filtri e ristabilire i precedenti.

```
1 filter -1 # elimina l'ultimo filtro aggiunto
2 filter 2 # elimina tutti i filtri tranne quelli di livello 1 e 2
3 filter 0 # elimina il filtro, come se si usasse 'all'
```

15.2 select

Selezione un sottoinsieme di casi in modo **definitivo**. I record vengono eliminati dal data frame caricato in memoria (non dal file su disco).

Utilizza la stessa sintassi di *filter*.¹

15.2.1 select_not

Il suffisso *_not* inverte l'espressione.

```
1 select :sesso => 1, :label => "Intervistati di sesso maschile"
2 select_not :sesso => 2, :label => "Intervistati di sesso maschile"
3 select :d1 => '<3'
4 select :d1 => [1,2], :d2 => 3..5 # ((d1=1 OR d1=2) AND (d2=3 OR d2=4 OR d2=5))
5 select "#sex==1 & etanum>=18", :label => "Intervistati maschi maggiorenni"
```

Non sono disponibili i prefissi *and_* e *or_*.

15.2.2 filter + select

Se usato senza argomenti, trasforma il filtro temporaneo (*filter*) attivo in una selezione definitiva.

```
1 # espressione composta: filter + select
2 filter :d1 => [1,2]
3 or_filter :d5 => 4
4 select
```

15.3 all

Se è stato selezionato un sottoinsieme di casi con il comando *filter*², *all* elimina il filtro riattivando tutti i record. E' possibile utilizzare i parametri *:id* e *:label* come in *filter*. L'etichetta può anche essere specificata come primo argomento senza la chiave *:label*.

```
1 all
2 all "Base: totale intervistati", :id => "tutti"
3 all :label => "Base: totale intervistati", :id => "tutti"
```

¹ Quando la selezione dei casi riguarda tutto il report (o tutta la parte restante), benché il risultato sia identico, **select** è da preferire a **filter** in quanto, eliminando le osservazioni, riduce i tempi di elaborazione. I dati non vengono eliminati dal file di dati, ma solo dal data frame in memoria.

² **all** non ha efficacia se è stato precedentemente utilizzato **select**.

15.4 cases

`cases` modifica i parametri `:id` e/o `:label` della selezione corrente (anche per il totale casi), senza modificare la selezione dei casi. L'etichetta può anche essere specificata come primo argomento senza la chiave `:label`. Può essere utilizzato anche se non è stato utilizzato nessuno dei comandi `filter`, `select` e `all`.

```
1 cases "BASE: totale casi"
2 cases :label => "BASE: totale casi"
3 cases :id => :male, :label => "BASE: maschi"
```

Suggerimento: Se si vuole semplicemente modificare l'etichetta stampata nelle tavole, utilizzare `cases` non `all`.

15.5 break

`break` esegue il codice contenuto nel blocco impostando un nuovo filtro a ogni ripetizione. Eventuali filtri attivi vengono eliminati.

`break` passa nel blocco due argomenti: l'etichetta e l'indice (con base 1).

`break` può essere eseguito come metodo di una variabile (indicata come stringa o come simbolo), ciclando così per ciascun livello della variabile:

```
1 :area.break do |lab, i|
2   cat :d1
3   ord :d5
4 end
```

`break` può essere eseguito come metodo di un *Hash* in cui viene indicato esplicitamente l'etichetta del filtro (le chiavi dell'hash) e l'espressione filtro (i valori dell'hash):

```
1 { "NORD"           => {:area => [1, 2]},
2   "CENTRO"        => {:area => 3},
3   "SUD - MASCHI"  => "#area == 4 & #sesso==1",
4   "SUD - DONNE"  => "#area == 4 & #sesso==2" }.break do |label, index|
5   scale s(:d6_, 10)
6   md s(:d7_, 10)
7 end
```

`break` può essere eseguito come metodo di un *Array* p di un *Range*, in tal caso si limita a iterare senza attivare nessun filtro.

`break` accetta i seguenti argomenti:

- `:total`: esegue il blocco di istruzioni anche senza filtro ("a totale") usando come etichetta la stringa indicata
- `:all`: attiva tutti i casi eliminando qualsiasi filtro precedente (default `true`)
- `:mode`: `:list` o `:page` definisce la modalità in cui visualizzare i gruppi di tabelle in *pTabsWeb* (default `:list`)
- `:label`: l'etichetta del controllo `select` di *pTabsWeb*

```
1 'sesso'.break(total: "Totale casi") do
2   cat :d1
3 end
4
5 {...}.break(total: "TOTALE") do ...
6
7 filter :area
8 :sesso.break(total: "", all: false) do |label, i|
9   and_filter :area => 1, :label => "Nord Ovest: #{label}"
```

```
10   cat :dl
11 end
12
13 (1..3).break(mode: :page) do ...
```

E' possibile generare le tavole *a totale* in modo esplicito:

```
1 { "TOTALE" => :area,
2   "NORD"   => {:area => [1,2]}, ... }.break ...
```

break può essere utilizzato non soltanto per le tabelle, ma anche con altri elementi come grafici e mappe:

```
1 :area.break("Totale Italia") do |lab, i|
2   map "Mappa", :lat => :latitudine, :lng => :longitudine, :caption => :name
3 end
```

Importante: `break` genera dei gruppi di tavole che vengono riconosciuti da *pTabsWeb* e trattati come un unico oggetto con un filtro selezionabile.

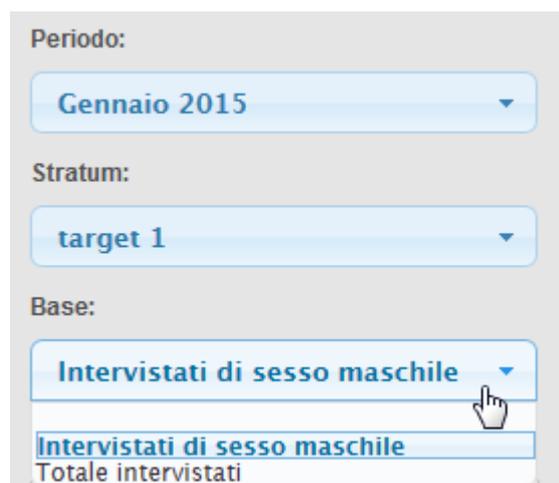
Consiglio: Per realizzare blocchi ripetuti di tavole e/o di altre istruzioni, vedere anche il metodo `rep` nella sezione *Funzioni di utilità*.

15.6 period e stratum

Nel formato per *pTabsWeb* sono disponibili le istruzioni `period` e `stratum` per etichettare e raggruppare le tabelle. Il comando sarà attivo fino alla successiva istruzione *period/stratum*. Le etichette verranno visualizzate insieme alla base come campi di ricerca in *pTabsWeb*.

```
1 period "Gennaio-giugno 2015"
2 stratum "Acquirenti"

1 # disabilita period e stratum
2 period
3 stratum
```



The screenshot shows a web interface with three sections: 'Periodo:', 'Stratum:', and 'Base:'. Each section has a dropdown menu. The 'Periodo:' dropdown is set to 'Gennaio 2015'. The 'Stratum:' dropdown is set to 'target 1'. The 'Base:' dropdown is set to 'Intervistati di sesso maschile', and a mouse cursor is clicking on it. Below the 'Base:' dropdown, there is a list of options: 'Intervistati di sesso maschile' and 'Totale intervistati'.

15.7 base_pos

Definisce dove deve essere inserita l'etichetta del filtro:

- `:corner`: l'angolo in alto a sinistra della tabella

- `:title`: nel titolo
- `:note`: nelle note della sottotabella
- `:tnote`: nelle note della tabella

15.8 n

Seleziona le prime n osservazioni.

```
1 n 100
```

15.9 sample

Estrae un sample delle osservazioni.

```
1 sample 100
```

15.10 ddup

Elimina i record duplicati. Restituisce il numero di record eliminati. E' possibile indicare le variabili su cui effettuare il controllo dei record duplicati oppure utilizzare tutte le variabili.

```
1 # elimina i record che risultano doppi rispetto a tutte le variabili
2 ddup
3
4 # elimina i record che risultano doppi per le variabili id_fam e id_pers
5 ddup :id_fam, :id_pers
6
7 # elimina i record e assegna il numero di record eliminati a num_deleted
8 num_deleted = ddup(:ser_no)
```

Note

Operazioni sui data frame

16.1 Workspaces

E' possibile caricare più di un file di dati, allo scopo, per esempio, di alternare la fonte di dati delle tavole, oppure per unire due file. In tali casi, i dati devono essere caricati in un'area di lavoro differente, utilizzando il parametro `:workspace => # o :ws => #`, dove `#` è un numero superiore a 1. Il data frame aggiuntivo si chiamerà `dfr#`. Non utilizzando il parametro `:workspace` la nuova matrice di dati prenderebbe il posto del data frame precedente. Il *workspace* di default è 1 corrispondente al data frame `dfr`.

```
1  spss.open "dati" # carica i dati in dfr
2  spss.open "altri_dati", :workspace => 2 # carica i dati in dfr2
3  r.open "altri_dati", :ws => 3 # carica i dati in dfr3
```

L'istruzione `workspace # o ws #` attiva un data frame differente e lo rende disponibile per la tabulazione. Il nome del data frame attivo è comunque sempre `dfr`.

```
1  cat :d1 # tavola sul data frame 1
2
3  workspace 2 # attiva il data frame 2 rendendolo disponibile come
4  r << "dfr$espcam <- dfr$peso * 1.327" # istruzione R che agisce sul data frame attivo (dfr)
5  weight! :espcam
6  cat :x1 # tavola sul data frame 2
7
8  ws 1 # riattiva il data frame 1
9  cat :d5 # tavola sul data frame 1
```

16.2 Merge

merge unisce le variabili di due data frame precedentemente caricati nelle rispettive aree di lavoro (*workspace*).

Parametri:

- **:x**: primo data frame; default: `dfr`
- **:y**: secondo data frame; default: `dfr2`
- **:z**: data frame destinazione; default: primo data frame
- **:by_x**: chiave di unione primo data frame; default: prima colonna
- **:by_y**: chiave di unione secondo data frame; default: prima colonna
- **:all_x**: tiene tutte le righe di x; default: `true`
- **:all_y**: tiene tutte le righe di y; default: `false`
- **:sort**: ordina data frame risultante sulla chiave; default: `true`

`merge` senza alcun parametro unisce in secondo data frame (*dfr2*) al primo (*dfr*) usando la prima colonna come chiave e tenendo tutti i casi del primo data frame (*dfr*).

I dati dei due data frame non devono essere precedentemente ordinati.

Il tipo di dato/formato delle chiavi può essere diverso: per esempio interi in un data frame e caratteri nell'altro. I dati verranno confrontati convertendo il dato più specifico verso quello più generico (in questo caso gli interi vengono convertiti in stringhe `123 => "123"`), pertanto `123` corrisponderà a `"123"`, ma non a `"123.00"` o a `" 123"`.

Attenzione, in caso di presenza di variabili con lo stesso nome, verranno aggiunti i suffissi `.x` e `.y` ai nomi delle variabili.

```
1  spss.open "dati" # => dfr
2  spss.open "comuni", :workspace => 2 # => dfr2
3  merge :by_x => :id_comune, :by_y => :id
```

16.3 Union

`union` unisce i casi di due data frame precedentemente caricati nelle rispettive aree di lavoro (*workspace*).

Parametri:

- **:x**: primo data frame; default: `dfr`
- **:y**: secondo data frame; default: `dfr2`
- **:z**: data frame destinazione; default: primo data frame

`union` senza alcun parametro unisce il secondo data frame (*dfr2*) al primo (*dfr*).

```
1  spss.open "maschi" # => dfr
2  spss.open "femmine", :workspace => 2 # => dfr2
3  union
```

16.4 Aggregate

`aggregate` genera un nuovo data frame applicando le funzioni specificate alle variabili dei record che hanno la stessa chiave.

Parametri:

- **:key**: un nome di variabile o un *array* con un elenco di variabili che costituiscono la chiave su cui aggregare i dati
- **:keep**: un *array* con un elenco di variabili che devono essere riportate come sono nei nuovi record: il loro valore viene assunto come costante rispetto alla chiave di aggregazione
- **:vars**: un *hash* con le nuove variabili da calcolare durante l'aggregazione
- **:workspace** o **:ws** (*opzionale*): il workspace di destinazione. Se omissso, utilizza il workspace successivo rispetto a quello attivo

16.4.1 Funzioni di aggregazione

Funzioni che agiscono su una sola variabile. Possono essere applicate una variabile per volta: `:newvarname => { :funzione => :varname }`, oppure su un elenco di variabili: `[newvarlist] => { :funzione => [varlist] }`. Il numero delle variabili generate deve essere uguale al numero di variabili di partenza:

- **:count**: senza argomenti conteggia il numero di record che vengono aggregati. Con il nome di una variabile, conteggia il numero di record con un valore valido rispetto alla variabile

- **:avg**: la media dei valori validi di una variabile
- **:min**: il valore minimo di una variabile
- **:max**: il valore massimo di una variabile
- **:sum**: la somma dei valori di una variabile

Alle funzioni è possibile aggiungere il suffisso **_distinct**: il risultato verrà calcolato escludendo i record con valori duplicati.

Funzioni che agiscono su un elenco di variabili; hanno la forma: `:newvarname => { :funzione => [varlist] }`. Viene generata una sola variabile a partire da un elenco di più variabili. L'operazione di aggregazione viene fatta in due diversi passi: prima si effettua un calcolo su ogni record del data frame e poi si aggrega il risultato:

- **:vcount**: conteggia il numero di valori validi in un set di variabili e poi somma il valore ottenuto per ciascun record
- **:vavg**: calcola la media dei valori validi di un set di variabili e poi calcola la media del valore ottenuto tra i record
- **:vmin**: calcola il valore minimo di un set di variabili e poi calcola il valore minimo tra i record
- **:vmax**: calcola il valore massimo di un set di variabili e poi calcola il valore massimo tra i record
- **:vsum**: calcola la somma dei valori di un set di variabili e poi calcola la somma dei valori tra i record

```

1 aggregate :key => :id_fam,
2           :keep => [:mcr, :ampc, :regio],
3           :vars => {
4             :nrecords => :count,
5             :nvalue_d5 => { :count_distinct => :d5 },
6             :totwgt => { :sum => :peso },
7             :meanwgt => { :avg => :peso },
8             s(:q6_, 10) => { :max => s(:d6_, 10) },
9             s(:d7_, 10) => { :sum => s(:d7_, 10) },
10            :nlinee => { :vcount => s(:linea_, 5) }
11            :toteuro => { :vsum => s(:linea_, 5) },
12          },
13          :ws => 2

```

Attenzione: Tutte le variabili utilizzate nel comando *aggregate* non devono contenere il punto nel nome.

16.5 Stack

stack genera un nuovo data frame concatenando differenti set di variabili di un data frame in singole variabili.

Converte un data frame dal formato *wide* al formato *long*.

Parametri:

- **:keep**: una stringa o simbolo oppure un array con le variabili che devono essere riportate come sono nei nuovi record
- **:vars**: un hash con i set di variabili che devono essere concatenati. Le chiavi dell'hash sono i nomi delle nuove variabili, mentre i valori sono gli elenchi delle variabili. Il tipo di variabile (numerica o factor) viene mantenuto
- **:group** (*opzionale*): il nome della variabili che distingue i gruppi di record. Può essere: una stringa o un simbolo, oppure un hash dove la chiave è il nome della variabile mentre i valori sono i valori da attribuire ai diversi gruppi. Se i valori sono stringhe (etichette) verrà generata una variabile factor.
- **:workspace** o **:ws** (*opzionale*): il workspace di destinazione. Se omissso, utilizza il workspace successivo rispetto a quello attivo

```

1 stack :keep => :peso, :vars => {:d1 => s(:v1_,10)}
2
3 stack :keep => [:ser_no, :area, :sesso, :eta, :peso],
4       :vars => {:d6 => s(:d6_,6), :d7 => s(:d7_,6)},
5       :group => {:bimestre => ["Bimestri 2013", ["Gen-Feb", "Mar-Apr", "Mag-Giu", "Lug-Ago", "Set-
6       :ws => 3

```

Possibili forme di `:group`:

```

1 :group => :x
2 :group => {:x => "VarLabel"}
3 :group => {:x => ["A", "B", "C", "D"]}
4 :group => {:x => ["VarLabel", ["A", "B", "C", "D"]]}
5 :group => {:x => 101..106}

```

Attenzione: Tutte le variabili utilizzate nel comando *aggregate* non devono contenere il punto nel nome.

16.6 Unstack

unstack genera un nuovo data frame contenente un solo record per ogni valore della chiave e affiancando gruppi ripetuti di variabili.

Converte un data frame dal formato *long* al formato *wide*.

Parametri:

- **:key**: un nome di variabile o un *array* con un elenco di variabili che identifica i record del nuovo data frame
- **:keep**: un *array* con un elenco di variabili che devono essere riportate come sono nei nuovi record: il loro valore viene assunto come costante rispetto alla chiave
- **:group**: il nome della variabile che identifica i gruppi da ripetere nel caso si voglia generare tutti i gruppi presenti nel file di dati oppure un *hash* con il nome della variabile e l'elenco dei valori dei gruppi da generare nel caso si intenda indicare gruppi in difetto o in eccesso rispetto a quelli effettivamente presenti nei dati
- **:vars**: le variabili da ripetere per ciascun gruppo
- **:workspace** o **:ws** (*opzionale*): il workspace di destinazione. Se omesso, utilizza il workspace successivo rispetto a quello attivo

```

1 unstack :key => :ser_no,
2         :keep => [:area, :sesso, :eta, :peso],
3         :group => :bimestre,
4         :vars => [:v1, :v4, :v6, :v7],
5         :ws => 2

```

Per assicurarsi di generare le variabili per tutti i bimestri anche se non sono presenti nei dati:

```

1 :group => {:bimestre => 1..6}

```

Attenzione: Tutte le variabili utilizzate nel comando *aggregate* non devono contenere il punto nel nome.

16.6.1 Esempio stack e unstack

Un data frame contiene il test di nove varianti di prodotto. Ogni intervistato testa solo tre varianti. Le risposte sono memorizzate in tre set di variabili. Le variabili `v1_1`, `v1_2` e `v1_3` contengono l'identificativo della variante testata.

Con *stack* si genera un data frame che contiene un solo set di variabili, ma tre record per ogni intervistato. Tale struttura è adatta per realizzare tavole con in pannello i diversi prodotti.

In seguito, con *unstack* si genera un data frame che contiene un solo record per intervistato e nove set di variabili, per ciascuna delle varianti di prodotto.

```
1  stack :keep => [:ser_no, :sesso, :eta, :peso],
2      :vars => {
3          :variante => s(:v1_, 3),
4          :v2 => s(:v2_, 3),
5          :v3 => s(:v3_, 3),
6          :v4 => s(:v4_, 3),
7          s(:v5_, 9) => ss("v5_#2_#1", 9, 3)
8      },
9      :group => {:test => ["Prodotti testati", ["Test 1", "Test 2", "Test 3"]]},
10     :ws => 2
11
12  ws 2
13
14  panels.add :_t, :variante
15  cat :v2
16  cat :v3
17  cat :v4
18  md s(:v5_, 9)
19
20  unstack :key => :ser_no,
21         :keep => [:sesso, :eta, :peso],
22         :vars => [:v1, :v2, :v3, :v4, s(:v5_, 9)],
23         :group => {:variante => 1..9},
24         :ws => 3
```

Gestione variabili

Per le espressioni vedi anche *Espressioni*

17.1 rename

Rinomina le variabili.

```

1  rename old_name_list => new_name_list
2
3  rename :sesso => :sex
4  rename [:sesso, :d1, 'd5.1'] => [:sex, :d100, :d5]
5  rename s(:d8_,7) => s(:x8_,7)

```

17.2 label

Imposta l'etichetta della variabile.¹

```

1  label var => label, varlist => label
2
3  label :sex => "Sesso dell'intervistato"
4  label :sex => "SESSO", :area => "MACROAREA"
5  label [:mcr1, :mcr2, :mcr3] => "MACROAREA"

```

17.3 levels

Imposta le etichette dei livelli.¹

```

1  levels var => labels, ...
2  levels :sesso => ['M', 'F']
3
4  levels var => {level => label, level => label}, ...
5  levels :area => {3 => "CENTRO", 4 => "Sud e Isole"}
6
7  levels :sesso => ['M', 'F'], :area => {3 => "CENTRO", 4 => "Sud e Isole"}
8
9  levels varlist => ...
10 levels s(:r100,8) => ["Sì", "No"]

```

¹ Nelle etichette delle variabili di pannello è possibile utilizzare il carattere _ (parametro :hyphen_char) e | (parametro :break_char). Sono utili nei pannelli dell'output di testo: il primo inserisce il trattino e manda a capo l'etichetta, il secondo va a capo senza inserire il trattino. Negli altri formati vengono ignorati ed eliminati.

17.4 add_levels

Aggiunge nuovi livelli alle variabili in coda a quelli esistenti.

```
1 add_levels varlist => levels
2
3 add_levels :v1 => ["Non sa", "Non risponde"]
4 add_levels [:v10, :v20] => "Non sa"
5 add_levels s(:v3_, 10) => "Rosso"
```

17.5 copy_label

Etichetta una o più variabili con l'etichetta di un'altra variabile.

```
1 copy_label var => varlist
2
3 copy_label :v1 => :r1
4 copy_label :v1 => [:r11, :r12, :r13]
```

17.6 copy_levels

Etichetta i livelli di una o più variabili con le etichette dei livelli di un'altra variabile.

```
1 copy_levels var => varlist
2
3 copy_levels :v1 => :r1
4 copy_levels :v1 => [:r11, :r12, :r13]
```

17.7 lprefix

Aggiunge un prefisso alle etichette dei livelli della variabile.

```
1 lprefix varlist => prefix
2
3 lprefix :v123 => "Kg. "
4 lprefix [:v123.1, :v123.2] => "Kg. "
```

17.8 lsuffix

Aggiunge un suffisso alle etichette dei livelli della variabile.

```
1 lsuffix varlist => suffix
2
3 lsuffix :etaf => " anni"
4 lsuffix [:eta1f, :eta2f, :eta3f] => " anni"
```

17.9 level_to_label

Copia l'etichetta di livello come etichetta della variabile.

```

1 level_to_label varlist, level_id
2
3 level_to_label s(d7_,10), 1

```

17.10 label_to_level

Copia l'etichetta della variabile come etichetta di un livello.

```

1 label_to_level varlist, level_id
2
3 label_to_level s(d7_,10), 1

```

17.11 factor

Trasforma una variabile numerica in factor e imposta le etichette. La variabile deve contenere numeri interi.

```

1 factor :etacon, :n7, :d1 # solo trasformazione in factor
2
3 factor :etacon => "Età puntuale", # con etichetta variabile
4 :sesso => ["SESSO", {1 => "M", 2 => "F"}], # con etichette variabile e livelli
5 :d1 => ["A","B","C","D"] # solo etichette livelli
6
7 factor :area => ["AREA GEOGRAFICA", ["NO", "NE", "C", "S"]]
8
9 factor s(:d7_,10) => ["Sì","No"] # più variabili con gli stessi livelli
10
11 factor :n1, :n2, :fill => true # controlla i valori esistenti nei dati e genera la sequenza di co
12
13 factor :d1 => ["X","Y","Z"]
14 # equivale a:
15 factor :d1 => {1=>"X", 2=>"Y", 3=>"Z"}

```

Avvertimento: E' necessario porre particolare attenzione ai casi in cui la variabile da trasformare non presenta una sequenza di valori continui che partono da 1.

```

1 Se la variabile v1 contiene i valori: 2, 4 e 5:
2
3 v1
4 # < N. % >
5 2 1 93 43.66
6 4 2 50 23.47
7 5 3 70 32.86
8 N. NA 213 100.00
9
10 factor :v1
11 # produce il warning:
12 + trovati 3 livelli per la variabile <v1>: 2, 4, 5 (ATTENZIONE: i codici dei livelli non corrispon
13 # e genera una variabile factor corretta, ma dove la sequenda dei livelli non corrisponde più ai
14
15 v1
16 # < N. % >
17 2 1 93 43.66
18 4 2 50 23.47
19 5 3 70 32.86
20 N. NA 213 100.00
21
22 factor :v1 => {4=>"D", 5=>"E"}
23 # produce l'errore:
24 - Errore nella sintassi del comando <factor>: [v1] i livelli (4,5) non corrispondono ai valori tr

```

23

24 `factor :v1 => ["B", "D", "E"]`25 `# produce l'errore:`26 `- Errore nella sintassi del comando <factor>: [v5] i livelli (1,2,3) non corrispondono ai valori f`27 `# equivale infatti a:`28 `factor :v1 => {1=>"B", 2=>"D", 3=>"E"}`

29

30 `factor :v1 => {2=>"B", 4=>"D", 5=>"E"}`31 `# produce il warning:`32 `+ trovati 3 livelli per la variabile <v1>: B, D, E (ATTENZIONE: i codici dei livelli non corrispon`33 `# e genera una variabile factor corretta, ma dove la sequenda dei livelli non corrisponde più ai`34 `v1`35 `# < N. % >`36 `B 1 93 43.66`37 `D 2 50 23.47`38 `E 3 70 32.86`39 `N. NA 213 100.00`

40

41 `factor :v1 => {1=>"A", 2=>"B", 3=>"C", 4=>"D", 5=>"E"}`42 `# genera una variabile factor completa anche dei livelli mancanti:`43 `v1`44 `# < N. % >`45 `A 1 0 0.00`46 `B 2 93 43.66`47 `C 3 0 0.00`48 `D 4 50 23.47`49 `E 5 70 32.86`50 `N. NA 213 100.00`

51

52 `factor :v1, :fill => true`53 `# genera una variabile factor completa anche dei livelli mancanti e con le etichette uguali ai va`54 `v1`55 `# < N. % >`56 `1 1 0 0.00`57 `2 2 93 43.66`58 `3 3 0 0.00`59 `4 4 50 23.47`60 `5 5 70 32.86`61 `N. NA 213 100.00`

17.12 numeric

Trasforma una o più variabili factor in variabili numeriche.

```
1 numeric varlist => new_varlist
2
3 numeric s(:d6_,10) => s(:x6_,10)
```

17.13 string

Trasforma una o più variabili factor in variabili stringa.

```
1 string varlist => new_varlist
2
3 string s(:d6_,10) => s(:txt6_,10)
```

17.14 create

Genera una nuova variabile senza dati. Utilizza la stessa sintassi di *factor*. Le variabili per cui non si specificano etichette, vengono definite come numeriche.

17.15 delete

Cancella le variabili elencate.

17.16 keep

Tiene solo le variabili elencate.

17.17 copy

Fa una copia di una variabile esistente.

```
1 copy oldvar => newvar
2 copy :d1 => :d1bis, :d2 => :d2bis
3 copy [:d1, :d2] => [:d1bis, :d2bis]
```

17.18 missing

Ricodifica a blank i codici indicati delle variabili.

```
1 missing var => value list, :drop => true
2 missing :d1 => 9
3 missing :d1 => [8, 9], :drop => true
4 missing 'd10.1' => [8,9], :d2 => 9
5 missing :d1 => "non so"
```

Se si indica più di un valore, i valori vanno messi tra parentesi quadre (Array). `:drop => true` elimina i livelli della variabile che hanno frequenza uguale a zero; in caso contrario il livello compare comunque con frequenza 0 nella tabella. Se si utilizza l'etichetta, questa deve essere esattamente uguale, compresi spazi e maiuscolo/minuscolo.

17.19 drop_levels

Elimina tutti i livelli della variabile che hanno frequenza uguale a zero.

```
1 drop_levels varlist
2 drop_levels :d1
3 drop_levels [:d1, :d2, :d3]
4 drop_levels s(:x10_, 8)
```

17.20 na_if

Imposta le variabili a blank se è vera la condizione. Se la condizione non è specificata, l'istruzione agisce su tutti i casi.

```
1 na_if varlist[, expression]
2
3 na_if :d1, :sesso => 1
4 na_if :d2
```

17.21 value

Assegna alle variabili i valori specificati. Accetta un solo valore per tutte le variabili o una lista di valori, uno per ciascuna variabile.

```
1 value varlist, valuelist
2
3 create :x1
4 value :x1, 1
5 value [:v1, :v2, :v3], 99
6 value [:v11, :v12, :v13], [6,7,8]
```

17.22 value_if

Imposta le variabili uguali al valore se è vera la condizione. Accetta un solo valore per tutte le variabili o una lista di valori, uno per ciascuna variabile. Se la condizione non è specificata, l'istruzione agisce su tutti i casi comportandosi come l'istruzione **value**.

```
1 value_if varlist, valuelist[, expression]
2
3 value_if :d5, 1, :d5 => :na
4 value_if s(:d7_10), 1, :d6 => 1
5 add_levels :d8 => "Non risponde"
6 value_if [:d8, :x8], [6,99] :d8 => :na
```

17.23 na_to_level

```
1 na_to_level "NON SO", :d5
```

17.24 recode

Ricodifica una variabile.

```
1 # con elenco di valori (pesi): ricodifica in una variabile numerica
2 recode var => [value list]
3 recode :d5 => [0, 10, 25, 50, 100]
4 recode :d5 => [NA, 10, 25, 50, 100]
5
6 # con stringa di ricodifica
7 recode var => expression, labels => []
8 recode :d5 => "1=NA;2:3=2;4:5=1", :labels => ["Sì", "No"]
9 recode :d5 => "NA=6", :labels => get_levels(:d5)+["Non si applica"] # vecchie etichette + nuova
10 recode :d5 => "3=NA", :labels => get_levels(:d5).ids(1,2,4,5) # elimino livello 3
11 recode :country => "c(15,44)=1;c(8,9,18,14,26,29,35)=2;c(24,40,42)=3;c(1,25,37,49)=4;else=NA"
12 recode s(:d2_,15) => 'NA=11', :labels => ['1','2','3','4','5','6','7','8','9','10','NON SO']
13 recode :x => "lo:1.10=1;1.11:1.20=2;1.21:hi=3", :labels => ["Basso", "Medio", "Alto"], :into =>
14
15 # con indicazione del numero complessivo di livelli
```

```

16 recode s(:d2_,15) => 'NA=11', :nlevels => 11
17
18 # ricodifica in una nuova variabile
19 recode :d5 => "1=NA;2:3=2;4:5=1", :labels => ["Sì", "No"], :into => :d5bis
20 recode :v100 => "1:3=1;else=0", :into => :v100_1 # else comprende anche le mancate risposte
21 recode :v100 => "1:3=1;NA=NA;else=0", :into => :v100_1 # non ricodifica le mancate risposte
22
23 # più variabili
24 recode [:d5, :d6] => "1=NA;2:3=2;4:5=1", :labels => ["Sì", "No"], :into => [:r5, :r6]
25
26 # con condizione: la variabile ottenuta è numerica
27 recode :n100 => "5:hi=5", :if => {:sesso => 1}
28 recode :n100 => "5:hi=5", :if => "#sesso == 1"
29 recode :n120 => "1=NA", :if => "etatum >= 18"

```

17.25 interaction

Combina i livelli di due o più fattori in un'unica variabile.

```

1 interaction newvar => [varlist], :sep => "string", :label => "label"
2
3 interaction :mcrsexeta => [:mcr, :sesso, :eta] # Il separatore di default per le etichette è ":
4 interaction :sexeta => [:sesso, :eta], :sep => " - ", :label => "CLASSI DI ETA' PER SESSO"

```

17.26 expand

Genera un set di variabili dummy da una variabile factor. Le variabili generate sono di tipo numerico, non factor. Le nuove variabili vengono etichettate con l'etichetta del livello corrispondente.

```

1 expand varname => new_var_root
2
3 expand :eta => :age_ # genera: age_1, age_2, age_3, ecc.

```

17.27 compress

Genera una variabile factor da più variabili. Il caso più comune è la ricostruzione di una variabile singola a partire da variabili dummy che in realtà non sono multiple.

Se non si specifica il parametro `:codes`, viene utilizzato il codice 1. E' possibile utilizzare anche un codice diverso per ciascuna variabile.

I valori vengono attribuiti seguendo l'ordine in cui sono state elencate le variabili. In caso di sovrapposizione, i nuovi valori sovrascrivono le precedenti assegnazioni.

Le etichette dei livelli delle nuove variabili, vengono prese dai livelli delle variabili originarie, nel caso di variabili *factor*, oppure dall'etichetta della variabile, nel caso di variabili numeriche.

```

1 compress varlist => [newvar, "label"], :codes => value|valuelist
2
3 compress s(:eta_,5) => [:eta, "ETA'"]
4 compress s(:eta_,5) => [:eta, "ETA'"], :codes => 1
5 compress s(:eta_,5) => [:eta, "ETA'"], :codes => [1,2,3,4,5]

```

17.28 mc_to_md

Trasforma un set di variabili *multicategory* in un set di variabili *multidicotomy*. Restituisce l'elenco delle variabili create.

```
1 mc_to_md varlist => new_vars_root
2
3 mc_to_md s(:d5_,10) => :x5_
4 mc_to_md s(:d5_,10) => 'x5.'
```

17.29 md_to_panel

Trasforma un set di variabili *multidicotomy* in nuove variabili adatte ad essere utilizzate nel pannello. Le variabili generate sono variabili *factor* con un unico livello etichettato con l'etichetta della variabile originaria. Se non viene specificato l'elenco delle variabili di destinazione, i nomi delle variabili vengono generate automaticamente. Se non viene specificato il codice delle variabili *multidicotomy*, viene utilizzato il valore 1. Restituisce l'elenco delle variabili create.

```
1 md_to_panel varlist, label, new_varlist, code
2
3 md_to_panel s(:v10_,1..3,8,9), "POSSESSO BENI", s(:beni_,5), 1
4 # poi, nel pannello
5 panels.add :_t, :sesso, s(:beni_,5) :area
6
7 # oppure direttamente nell'istruzione panels.add anche senza specificare i nomi
8 panels.add :_t, :sesso, md_to_panel(s(:v10_,1..3,8,9), "POSSESSO BENI"), :area
```

17.30 reorder

Riordina i livelli di una variabile *factor* riordinando anche le etichette.

- un vettore con l'elenco dei livelli ordina la variabile in modo arbitrario
- `:rev` invertire l'ordine dei livelli
- `:freq` ordina in base alle frequenze in senso crescente
- `:dfreq` ordinare in base alle frequenze in senso decrescente

Se è attiva una variabile di ponderazione vengono considerate le frequenze pesate.

```
1 # riordina i livelli della variabile v1
2 reorder :v1, 3, 2, 5, 4, 1
3
4 # inverte l'ordine dei livelli della variabile v1
5 reorder :v1, :rev
6
7 # riordina i livelli in base alle frequenze
8 reorder :v1, :freq
9
10 # riordina i livelli in base alle frequenze in senso decrescente
11 reorder :v1, :dfreq
```

17.31 switch

Scambia due livelli di una variabile *factor* scambiando anche le etichette.

```

1  # inverte il codice 70 della variabile v30 con il 75
2  switch :v30, 70, 75

```

17.32 move_before

Sposta dei livelli di una variabile factor prima di un determinato livello. Riordina anche le etichette.

```

1  # sposta il livello 1 prima del livello 6
2  move_before :v100, 1, 6
3
4  # sposta i livelli 4, 6 e 7 prima del livello 1
5  move_before :v100, [4,6,7], 1

```

17.33 move_after

Sposta dei livelli di una variabile factor dopo di un determinato livello. Riordina anche le etichette.

```

1  # sposta il livello 1 dopo il livello 6
2  move_after :v100, 1, 6
3
4  # sposta i livelli 5, 6 e 7 dopo il livello 1
5  move_after :v100, 5..7, 1

```

17.34 if_any

Crea una nuova variabile che è uguale a 1 se almeno una delle variabili contiene il valore indicato e 0 in caso contrario.

```

1  if_any variables_list, code, new_var, label
2  if_any [:d7_2, :d7_3], 1, :x7, "EMOZIONI"

```

17.35 if_all

Crea una nuova variabile che è uguale a 1 se tutte le variabili contengono il valore indicato e 0 in caso contrario.

```

1  if_all variables_list, code, new_var, label
2  if_all s(:d7_,5..8), 1, :x7, "Sia ... sia ..."

```

17.36 if_none

Crea una nuova variabile che è uguale a 1 se tutte le variabili non contengono il valore indicato e 0 in caso contrario.

```

1  if_none variables_list, code, new_var, label
2  if_none s(:d7_,10), 1, :x7

```

17.37 if_only

Crea una nuova variabile che è uguale a 1 se almeno una delle variabili del primo set contiene il valore indicato mentre nessuna delle variabili dell'altro set lo contengono. Il secondo set può anche contenere tutto l'insieme delle variabili, pTabs provvederà a sottrarre le variabili del primo set.

```
1 if_only variables_list_1, code, variables_list_2, new_var, label
2 if_only [:d7_3,:d7_5], 1, s(:d7_,1..10), :x7, "Solo 3 o 5"
```

17.38 split_vars

Crea nuove variabili uguali alle variabili di partenza, ma che contengono solo i gruppi di casi definiti da una data variabile. Restituisce un vettore con i nomi delle variabili create.

Con `:names` è possibile definire un modello per i nomi delle nuove variabili: `#` è il segnaposto per il livello della variabile che definisce i gruppi, `##` è il segnaposto per il nome della variabile di origine. Se `:names` non viene specificato, i nomi vengono generati automaticamente secondo il modello `NomeVariabileGruppi_#_##`.

```
1 split_vars variable_group => variables_list, :names => template
2
3 # crea delle copie delle variabili per ciascun livello di :sesso
4 split_vars :sesso => [:d1,:d2]+s(:d7_,10), :names => "##_sex#"
5
6 # assegna a varlist l'elenco delle variabili create
7 varlist = split_vars :sesso => [:d1,:d2]+s(:d7_,10)
```

17.39 derotate

Attenzione: Funzionalità in test. Verificare il risultato.

Scambia le risposte tra diversi set di variabili secondo la rotazione di appartenenza di ciascun caso.

Le rotazioni possono essere definite in due modi:

- da una variabile numerica o factor che indica l'appartenenza del record a una rotazione (cella). In tal caso è necessario specificare anche il parametro `:rot`.
- da una serie di variabili numeriche o factor che indicano le varie permutazioni. Ogni variabile identifica l'oggetto di ciascun set di domande somministrate. Il numero di queste variabili deve essere uguale al numero di set di variabili da derotare. Il parametro `:rot` non deve essere specificato (il dataset contiene le informazioni necessarie).

`:rot` specifica lo schema delle rotazioni. E' un vettore che, per ciascun livello/valore della variabile che identifica la rotazione, contiene la sequenza di rotazione.

Le variabili da derotare sono un vettore di vettori di nomi di variabili: [[:x1, :y1, :z1], [:x2, :y2, :z2], ...].

Nell'esempio, le ipotetiche domande *a* e *b* vengono ripetute per tre volte per testare tre diversi oggetti. Le domande vengono somministrate ai vari soggetti in quattro diverse sequenze. E' da notare che il numero di oggetti da testare e il numero di permutazioni non sono necessariamente uguali.

```
1 derotate variable_id_rotazione|variables_list_sequenza => variables_list, :rot => schema
2
3 # l'elenco delle variabili da derotare, memorizzato per comodità nella variabile varlist
4 varlist = [ [:a1]+s(:b1_,10),
5             [:a2]+s(:b2_,10),
6             [:a3]+s(:b3_,10) ]
```

```

5
6 # ...oppure, più sinteticamente:
7 varlist = 3.rep.map { |i| ["a#{i}"]+s("b#{i}_",10) }
8
9 # ...il vettore multidimensionale risultante:
10 [ ["a1", "b1_1", "b1_2", "b1_3", "b1_4", "b1_5", "b1_6", "b1_7", "b1_8", "b1_9", "b1_10"],
11   ["a2", "b2_1", "b2_2", "b2_3", "b2_4", "b2_5", "b2_6", "b2_7", "b2_8", "b2_9", "b2_10"],
12   ["a3", "b3_1", "b3_2", "b3_3", "b3_4", "b3_5", "b3_6", "b3_7", "b3_8", "b3_9", "b3_10"] ]

1 # Metodo 1: con id rotazione
2
3 # :rotazione è la variabile che identifica la rotazione/cella
4   rotazione
5     1     2
6     2     4
7     3     3
8     4     1
9     5     1
10    6     2
11    ...
12
13 # rot_schema contiene l'elenco delle permutazioni utilizzate
14 # (deve corrispondere ai livelli della variabile :rotazione)
15 rot_schema = [ [1,2,3],           # per :rotazione = 1
16               [2,3,1],           # per :rotazione = 2
17               [3,1,2],           # per :rotazione = 3
18               [1,3,2] ]          # per :rotazione = 4
19 derotate :rotaz => varlist, :rot => rot_schema

1 # Metodo 2: con variabili sequenza
2
3 # le variabili :ro1, :ro2 e :ro3 contengono la sequenza degli elementi (l'ordine dei test)
4 # della permutazione utilizzata
5   ro1 ro2 ro3
6     1  2  3  1
7     2  1  3  2
8     3  3  1  2
9     4  1  2  3
10    5  1  2  3
11    6  2  3  1
12    ...
13
14 derotate s(:ro,3) => varlist

```

17.40 fill

Genera dati casuali. E' utile per svolgere simulazioni e test di tavole. Per le variabili *factor* vengono generati i valori corrispondenti ai livelli della variabile. Per le variabili numeriche è necessario specificare il range di valori. Vengono generati solo valori interi.

Nel caso sia impostato *filter*, *fill* genera valori per i soli casi non filtrati.

Opzioni:

- *:range*: vettore con il valore minimo e il valore massimo. Solo per numeriche.
- *:cases*: *:na* (*default*) genera valori solo per i casi in cui la variabile è missing, *:valid* genera valori solo per i casi in cui la variabile ha un valore, *:all* genera valori per tutti i casi
- *:probs*: vettore con le probabilità di estrazione per ciascun valore da estrarre. Devono essere in numero uguale ai diversi valori da generare. Il parametro *:auto* calcola automaticamente le probabilità in base alla distribuzione dei casi validi della variabile (casi filtrati se è attivo un filtro).

- `:seed`: un numero intero che viene usato come *seed* per la generazione dei valori casuali

Quando viene fornito il parametro `:probs` e non il parametro `:seed`, viene calcolata iterativamente la soluzione migliore.

```
1 fill :dl
2 fill :sesso, :probs => [45, 55]
3 fill :area, :probs => [0.2, 0.2, 0.5, 0.1]
4 fill :v100, :range => [1,10], :cases => :all
5 fill :v4500, :cases => :na, :probs => :auto
6
7 filter :sesso => 1, :label => "Maschi"
8 fill :v221, :seed => 3
```

Produce nell'output una tabella con le distribuzioni di frequenze dei valori iniziali della variabile, dei valori generati e dei valori finali:

```
> fill :x1, :range => [1,2], :cases => :na
```

```
Fill x1
  Value   Pre  ->  Gen  ->  Post
1     1     NA    66    66
2     2     NA    80    80
3     9    154    NA   154
4 <NA>   146     0     0
```

17.41 seed

Imposta il *seed* per la generazione delle sequenze di numeri casuali. Permette la riproducibilità delle sequenze di numeri casuali.

```
1 seed 2
```

Note

Espressioni

Diverse funzioni in pTabs (*filter*, *select*, *rake*, *value_if*, *recode*, *ecc.*) richiedono o consentono l'utilizzo di espressioni sulle variabili del data frame. E' possibile specificare le espressioni in tre modi diversi:

- il semplice **nome** della variabile o nome preceduto da ! per indicare i casi validi o non validi sulla variabile
- un **hash** (*dizionario*) con le coppie variabile-valori
 - più coppie variabile-valori sono unite in *AND* nell'espressione
 - `:na` e `:valid` rappresentano, rispettivamente, i valori mancanti e i valori validi
 - in relazione al contesto, in alcuni casi le parentesi graffe `{ }` possono essere omesse
- una **stringa** con un'espressione R
 - operatori relazionali: `==`, `!=`, `<`, `>`, `<=`, `>=`
 - operatori logici: `&` (AND), `|` (OR), `!` (NOT)
 - appartenenza: `%in%`
 - il cancelletto (`#`) davanti alla variabile converte la variabile in numerica ¹

Esempi con sintassi alternative:

```

1  # valori validi
2  {:x1 => :valid}
3  :x1
4  "!is.na(x1)"
5
6  # valori mancanti
7  {:x1 => :na}
8  "!x1"
9  "is.na(x1)"
10
11 # uguale a un valore
12 {:sesso => 1}
13 "#sesso == 1"
14 "as.numeric(sesso) == 1"
15
16 # minore di un valore
17 {:etanum => '<25'}
18 "etanum < 25"
19
20 # due condizioni diverse in AND
21 {:sesso => 1, :etanum => '<25'}
22 "#sesso == 1 & etanum < 25"
23
24 # più valori di una stessa variabile (in OR)

```

¹ Quest'uso del carattere # non è in realtà parte del linguaggio R. pTabs lo sostituisce con la funzione `as.numeric()` traducendo così i livelli di una variabile factor in una sequenza di numeri interi che partono da 1. Per esempio `#x1` viene trasformato in `as.numeric(x1)`.

```

25  {:region => [1,3,7]}
26  "#region %in% c(1,3,7)"
27
28  # range
29  {:etanum => 18..24}
30  "etanum %in% 18:24"

```

Esempi di espressioni realizzabili solo con espressioni R:

```

1  # diverso
2  "#sesso != 1"
3
4  # NOT
5  "!etanum == 18"
6
7  # due condizioni in OR
8  "#campio == 1 | #area == 3"
9
10 # etichetta del livello
11 "sesso == 'Maschio'"
12
13 # parentesi
14 "(v1.1 == 1 | v1.2 == 1) & v1.3 == 0"
15 "! (v1.1 == 1 & v1.2 == 1)"
16
17 # funzioni
18 "apply(dfr[,c('d7_1', 'd7_2', 'd7_3')]==1, 1, all)"

```

18.1 parse, parse!, parse?

parse espande/converta in R l'espressione e la restituisce come stringa

parse! espande/converta in R l'espressione, la stampa e la restituisce come stringa

parse? espande/converta in R l'espressione, controlla che sia valida, la stampa e restituisce un vettore di tre elementi:

- il numero di casi
- l'espressione R
- l'eventuale messaggio di errore

```

1  x = parse(:region => 3..7, :sesso => 2)
2  debug x
3  # => <String> "(as.numeric(region) %in% 3:7) & (as.numeric(sesso)==2)"
4
5  parse! :region => 3..7, :sesso => 2
6  # => <String> "(as.numeric(region) %in% 3:7) & (as.numeric(sesso)==2)"
7
8  x = parse?(:region => 3..7, :sesso => 2)
9  debug x
10 # => <Array> [102, "(as.numeric(region) %in% 3:7) & (as.numeric(sesso)==2)", ""]
11 ncasi, exp, err = x
12 debug ncasi, exp, err
13 # => <Fixnum> 102
14 # => <String> "(as.numeric(region) %in% 3:7) & (as.numeric(sesso)==2)"
15 # => <String> ""
16
17 parse?(:region => 3..7, :sesso => 2)
18 # => <Array> [0.0, "(as.numeric(xregion) %in% 3:7) & (as.numeric(sesso)==2)", "(checkExpression)
19 ...
20 ...

```

- 21 - R ha restituito 1 errore/i:
- 22 - (checkExpression) oggetto "xregion" non trovato

Vedi anche *get_selection* e *get_selection_u*

Note

Gruppi di variabili

L'oggetto **sets** permette la definizione e la gestione di gruppi di variabili multiple e le batterie di variabili singole.

I **set** trovati possono essere utilizzati con gli *hole count* o con le *tabelle* e vengono automaticamente utilizzati dalla procedura di *top-line*.

Il parametro di configurazione `sets_display` permette di mostrare nell'output i dettagli dei gruppi trovati.

19.1 Metodi dell'oggetto sets

19.1.1 sets.find

Cerca di ricostruire i gruppi di variabili multiple e, eventualmente, le batterie di variabili singole riconoscendo la radice comune nel nome delle variabili. Eventuali set di variabili esistenti vengono cancellati.

Parametri:

- **:start**: caratteri di inizio (espressione regolare)
- **:sep**: caratteri separatori (espressione regolare)

Le impostazioni di default e altri parametri sono definiti nei *Parametri di configurazione*.

I set generati sono oggetti Ruby con nome uguale al prefisso impostato dal parametro di configurazione `sets_pref` più il nome della radice.

19.1.2 sets.find_add

Cerca e aggiunge i set di variabili trovati a quelli esistenti.

19.1.3 sets.all

Cerca tutti i gruppi di variabili non solo i gruppi di variabili multiple.

19.1.4 sets.all_add

Cerca tutti i gruppi di variabili (non solo i gruppi di variabili multiple) e li aggiunge a quelli esistenti.

19.1.5 sets.show

Stampa nell'output i dettagli dei set definiti.

19.1.6 sets.list

Stampa nell'output una lista dei nomi dei set definiti.

19.1.7 sets.get(nome) o sets[nome]

Restituisce un *set* dal nome.

19.1.8 sets.rename

Rinomina un set.

19.1.9 sets.delete

Elimina un set.

19.2 Metodi degli oggetti set

19.2.1 nomeset.title

Restituisce / assegna il titolo di un set.

19.2.2 nomeset.vars

Restituisce / accede all'elenco dei nomi delle variabili del set.

Anche il solo nome del set può essere usato nella maggior parte dei contesti per restituire l'elenco delle variabili

19.2.3 nomeset.multi!

Forza il tipo di set come multiplo.

19.2.4 nomeset.single!

Forza il tipo di set come singolo.

19.3 Esempi

```
1 sets_display!  
2  
3 sets.find_add  
4 # --- defining set: m_d7 [multi] (n.12): d7_1,d7_2,d7_3,d7_4,d7_5,d7_6,d7_7,d7_8,d7_9,d7_10,d7_99  
5 # --- defining set: m_q50_1 [multi] (n.4): q50_1_1,q50_1_2,q50_1_3,q50_1_4  
6 # --- defining set: m_q50_2 [multi] (n.4): q50_2_1,q50_2_2,q50_2_3,q50_2_4  
7 # --- defining set: m_q50_3 [multi] (n.4): q50_3_1,q50_3_2,q50_3_3,q50_3_4  
8 # --- defining set: m_q50_4 [multi] (n.4): q50_4_1,q50_4_2,q50_4_3,q50_4_4  
9  
10 sets.rename "m_d7" => "m7"  
11 # --- defining set: m7 [multi] (n.12): d7_1,d7_2,d7_3,d7_4,d7_5,d7_6,d7_7,d7_8,d7_9,d7_10,d7_99,d  
12 m7.pop # esclude l'ultima variabile (d7_99_na_)  
13 m7.title "Domanda Q7"
```

```
14
15 sets.show
16 # --- Sets:
17 # --- m7 [multi] (n.11): d7_1,d7_2,d7_3,d7_4,d7_5,d7_6,d7_7,d7_8,d7_9,d7_10,d7_99
18 # --- m_q50_1 [multi] (n.4): q50_1_1,q50_1_2,q50_1_3,q50_1_4
19 # --- m_q50_2 [multi] (n.4): q50_2_1,q50_2_2,q50_2_3,q50_2_4
20 # --- m_q50_3 [multi] (n.4): q50_3_1,q50_3_2,q50_3_3,q50_3_4
21 # --- m_q50_4 [multi] (n.4): q50_4_1,q50_4_2,q50_4_3,q50_4_4
22
23 md m7, m7.title
24 # --- compute table # 1 - Domanda Q7

1 sets.find
2 topline
```

Date e tempo

20.1 chron

L'oggetto **chron** presenta vari metodi che permettono di convertire stringhe in data e ora e di estrarre informazioni dalle variabili data/ora.

20.1.1 Funzioni di conversione

I metodi **time** e **date** generano una variabile data/tempo a partire da una costante stringa o da una variabile contenente una stringa. Il parametro `:format` permette di specificare il formato di lettura se diverso da quello di default: `%Y-%m-%d %H:%M:%S` per *time* e `%Y-%m-%d` per *date*.

I metodi **ymd** e **ymdhms** convertono direttamente i formati `%Y%m%d` e `%Y%m%d%H%M%S`.

```

1 chron.time :oras => :o1, :format => "%d/%m/%Y %H.%M.%S"
2 chron.time "2013/03/15 14.13.01" => :t1, :format => "%Y/%m/%d %H.%M.%S"
3 chron.date "15.03.2013" => :g1, :format => "%d.%m.%Y"
4
5 chron.ymd "20130228" => :sd1
6 chron.ymdhms "20130130123001" => :sd2

```

20.1.2 Funzioni di estrazione

I seguenti metodi estraggono informazioni da una variabile in formato data/tempo:

- **year**: anno
- **month**: mese (numero)
- **day**: giorno del mese
- **yday**: giorno dell'anno
- **wday**: giorno della settimana (numero da 0 a 6 a partire da domenica)
- **hour**: ora
- **min**: minuti
- **sec**: secondi

```

1 # t1 => 15/02/2013 14:35:01
2 chron.year :t1 => :yyyy # 2013
3 chron.month :t1 => :mm # 2
4 chron.day :t1 => :dd # 5
5 chron.yday :t1 => :yd # 36
6 chron.wday :t1 => :wd # 2
7 chron.hour :t1 => :hh # 14

```

```
8  chron.min :t1 => :mi      # 35
9  chron.sec :t1 => :se      # 1
```

20.2 spss.time e spss.date

Convertono le variabili data/tempo di SPSS in variabili data/tempo di R.

spss.date esclude le informazioni relative all'ora.

Nota: SPSS memorizza le date come numero di minuti a partire dal 14 ottobre 1582.

```
1  spss.time :x1 => :z1, :x2 => z2
2  spss.date :giorno => :day
```

Funzioni di utilità

Le seguenti funzioni di utilità sono disponibili all'interno del blocco di istruzioni del report.

21.1 s

`s` o `seq` genera una lista di nomi di variabili. Il carattere # ¹, se presente, viene sostituito dal numero della sequenza. E' possibile utilizzare interi (10), range (2..7), vettori (1, 3, 5, 7, 9). Un numero negativo esclude l'elemento dalla sequenza. Eventuali elementi non numerici ('net.2') vengono inseriti come sono. Specificando solo la radice del nome, vengono restituiti i nomi di variabili presenti nel file di dati che iniziano con quei caratteri. Un numero negativo esclude l'elemento dalla sequenza. `true` come secondo argomento, esegue il padding con lo zero.

```

1  s(string, sequences/ranges/values/names)
2  s(string, padding, sequences/ranges/values/names)
3  s([pre,post], padding, sequences/ranges/values/names)
4
5  s(:x_, 5) # => x_1 x_2 x_3 x_4 x_5
6  s('x.', 99,98,7..10,1..3) # => x.99 x.98 x.7 x.8 x.9 x.10 x.1 x.2 x.3
7  s(:x_, true, 7..10) # => x.07 x.08 x.09 x.10
8  s(:x1_, 1..7,-4,-6) # => x1_1 x1_2 x1_3 x1_5 x1_7
9  s(:q, 1..7,-6..-4) # => q1 q2 q3 q7
10 s(:x, 1..3, :net1,4..7,'net.2',99) # => x1 x2 x3 net1 x4 x5 x6 x7 net.2 x99
11 s([:v,'.1'], 101..104,199) # => v101.1 v102.1 v103.1 v104.1 v199.1
12 s('v#.1', 101..104,199) # => v101.1 v102.1 v103.1 v104.1 v199.1
13
14 s(:v10_) # => ricerca automatica delle variabili nel file di dati
15 s(:v10_, :a) # => ricerca automatica con ordinamento alfabetico

```

21.2 ss

`ss` o `sseq` genera una lista di liste (matrice) di nomi di variabili. Specificare una stringa contenente i nome della variabile con i segnaposti #1 e #2 che verranno sotituiti dalle due sequenze numeriche. E' possibile utilizzare interi (10), range (2..7), vettori ([1, 3, 5, 7, 9]). Un numero negativo esclude l'elemento dalla sequenza. Se si utilizza un elemento composito, passare tutto l'argomento come vettore: [true,2..7,-5,-6]. `true` come secondo argomento, esegue il padding con lo zero. Se lo zero deve essere aggiunto a una sola sequenza, specificare `true` come primo elemento dell'argomento.

La funzione è utile per generare le liste di variabili nelle tabelle `grid`.

¹ I caratteri # e . possono essere utilizzati solo nelle stringhe, non nei simboli. Il carattere # non è un carattere valido nei nomi di variabile. Se presente nel file originale, viene convertito in . o _ a secondo delle impostazioni di `dot_to_underscore` e `underscore_to_dot`.

21.4 rev

`rev` è un metodo applicabile a *Array* e *Range* che inverte l'ordine degli elementi.

```
1 (1..7).rev          # => [7, 6, 5, 4, 3, 2, 1]
2 s(:d1_,5).rev      # => ['d1_5', 'd1_4', 'd1_3', 'd1_2', 'd1_1']
```

21.5 ids

`ids(indici)` è un metodo applicabile a *Array*. Restituisce un vettore con solo gli elementi con gli indici indicati.

```
1 ('A'..'Z').to_a.ids(2,17)  # => ['C', 'R']
2 get_levels(:d5).ids(1,2,4,5) # => etichette dei livelli 1, 2, 4 e 5 della variabile d5
```

21.6 in_groups_of

`in_groups_of(n)` è un metodo applicabile a *Array*. Raggruppa gli elementi in gruppi restituendo un vettore di vettori.

```
1 varlist = ["q5_1_1", "q5_1_2", "q5_1_3", "q5_1_4", "q5_2_1", "q5_2_2", "q5_2_3", "q5_2_4"]
2 varlist.in_groups_of(4) # => [["q5_1_1", "q5_1_2", "q5_1_3", "q5_1_4"], ["q5_2_1", "q5_2_2", "q5_2_3", "q5_2_4"]]
```

21.7 pr

`pr` stampa nel log il valore dell'oggetto e restituisce il valore.

```
1 md s(:v1_,3).pr          # => [:v1_1, :v1_2, :v1_3]
```

21.8 common

`common(string)` è un metodo applicabile a *String* che confronta due stringhe e restituisce la sottostringa comune.

```
1 label :v1_2 => get_label(:v1).common(get_label(:v2))
2
3 get_label(:v1)      # => "Gradimento prodotto XXX"
4 get_label(:v2)      # => "Gradimento prodotto YYYYY"
5 get_label(:v1_2)    # => "Gradimento prodotto"
```

21.9 log_width

Imposta il numero di colonne dell'output di *R*.

```
1 log_width 200
```

21.10 next

`next`, all'interno del blocco di un report, esclude l'esecuzione delle istruzioni seguenti. Le istruzioni sospese devono comunque essere sintatticamente corrette.

21.11 stop! / start!

Sospende/riavvia l'esecuzione delle istruzioni dello script all'interno del blocco di un report.

`stop!` sospende l'esecuzione da quel punto dello script in avanti. Le istruzioni sospese devono comunque essere sintatticamente corrette.

`start!` riavvia l'esecuzione.

Si possono inserire più istruzioni `stop!` e `start!` nello script. Le istruzioni che vengono 'saltate' devono comunque essere sintatticamente corrette.

21.12 r.exec o r <<

E' possibile eseguire direttamente istruzioni **R**. A questo proposito è da notare che il data frame utilizzato da *pTabs* è chiamato `dfr`.

```
1 r.exec "istruzione R"
2 r << "istruzione R"
3
4 r << <<-END
5   istruzione R
6   istruzione R
7   ...
8 END
```

21.13 r.source

Carica ed esegue un file di istruzioni **R**.

```
1 r.source "r_file1.R"
2 r.source "prog/r_file2.R"
3 r.source 'c:\rscripts\r_file3.R'
```

21.14 \$p

Contiene un riferimento all'oggetto report corrente. E' utile per eseguire metodi del report all'interno di un altro contesto, per esempio in un'unione di tabelle.

```
1 htjoin "Tabella di confronto di tre ponderazioni", "PONDERAZIONI" do
2   $p.weight.by :ipf1
3   md s(:d7_,1..10), "Peso 1"
4   $p.weight.by :ipf2
5   md s(:d7_,1..10), "Peso 2"
6   $p.weight.by :ipf3
7   md s(:d7_,1..10), "Peso 3"
8 end
```

In questo caso può essere direttamente utilizzata l'istruzione `weight!`.

21.15 css

L'output *HTML* e *PDF* sono controllati da **fogli stile CSS**. L'utente può ridefinire i propri fogli stile, ma per operazioni più veloci e/o temporanee di modifica degli stili è possibile inserire delle **istruzioni css** direttamente nello script tramite l'istruzione `css`.

```
1 # disattiva gli stili dei NET
2 css << "table td.net { font-weight: normal; }"
3 css << "table td.lev1, table td.lev2, table td.lev3 { font-style: normal; }"
4 css << "table td.label.lev1, table td.label.lev2, table td.label.lev3 { padding-left: 6px; }"
```

21.15.1 css.table, css.chart, css.map, css.doc

Specifica delle **istruzioni css** solo per l'ultima tavola/grafico/mappa/testo prodotta. Le istruzioni vanno poste dopo l'istruzione che produce la tabella/grafico/mappa/testo. L'istruzione `css` viene preceduto dal selettore di tabella/grafico/mappa/testo.

```
1 css.table << " td.net { font-weight: normal; color: blue }"
2 # div#ptabs > div.table table#t1 td.net { font-weight: normal; color: blue }
3
4 css.chart << "table.chart.gauge svg > g > circle:nth-child(1) { fill: #4682B4; stroke: #616161; }"
5 # div#ptabs > div.chart table#t3 table.chart.gauge svg > g > circle:nth-child(1)
```

21.15.2 css.load

pTabs carica automaticamente i fogli stili con i nomi definiti nei *parametri di configurazione*. I file vengono cercati nella cartella di installazione di pTabs, nella *cartella dell'utente* e nella cartella di lavoro.

Con l'istruzione `css.load(filename)` è inoltre possibile caricare degli ulteriori **fogli stile css** nello script.

```
1 css.load 'my_syle.css'
```

21.16 debug

Stampa nel log il contenuto di una variabile, di un oggetto o il valore di ritorno di un metodo.

```
1 x = 'info'.yload
2 debug x
```

21.17 Scrittura e lettura di informazioni

Gli *Hash* (`{:key => value}`) dispongono di alcuni metodi che consentono la scrittura e la lettura del loro contenuto in formato *YAML* e *JSON*.

Tale funzione è utile, per esempio, per configurare uno script tramite un file esterno o per passare dati tra diversi script.

ydump e **jdump** scrivono l'Hash rispettivamente su file *YAML* e *JSON*. Se il file esiste viene sovrascritto.

yload e **jload** caricano l'Hash rispettivamente da file *YAML* e *JSON*. Se il file non esiste compare un messaggio di warning, ma l'esecuzione prosegue senza problemi.

```
1 # Scrittura file YAML
2
3 rake_result = rake ...
4
```

```
5 { :eff => rake_result.efficiency,
6   :min => rake_result.min,
7   :max => rake_result.max }.ydump 'ponderazione'
8
9 # genera il file: ponderazione.ydump
10 ----
11 :eff: 88.2
12 :min: 0.3576
13 :max: 2.0513
14
15 # Scrittura file JSON
16
17 {:d5 => get_counts(:d5)}.jdump 'd5'
18
19 # genera il file: d5.json
20 {"d5": [[1, "Non risponde", 20.0], [2, "Per niente d'accordo", 16.0], [3, "Poco d'accordo", 83.0],
21 [4, "Abbastanza d'accordo", 95.0], [5, "Molto d'accordo", 56.0]]}
22
23 # Lettura file YAML
24
25 # info.yaml
26 ----
27 :data_file: dataFileName
28 :titolo: Titolo report
29 :filtro:
30   :sesso: 1
31
32 # nello script:
33 x = 'info'.yload
34 # accedo agli elementi dell'Hash x con l'operatore []
35 title x[:titolo]
36 spss.open x[:data_file]
37 filter x[:filtro]
38
39 # oppure, utilizzando la funzione defvar, le variabili saranno direttamente disponibili nello sc
40 defvar 'info'.yload
41 title titolo
42 spss.open data_file
43 filter filtro
```

Note

Funzioni informative

Le funzioni seguenti restituiscono informazioni relative alle variabili e ai dati presenti nel dataset. Sono utili per titoli e note o per condizionare il comportamento e rendere più flessibile lo script di pTabs.

22.1 Variabili

22.1.1 exist?

Controlla se le variabili esistono nel dataframe corrente e restituisce `true` se tutte le variabili esistono e `false` se almeno una variabile non esiste.

```
1 cat :v10045 if exist?(:v10045)
2 ord [:eta, etaq] if exist?(:eta, etaq)
3 md s(:v100,10) if exist?(s(:v100,10))
```

22.1.2 existing

Controlla se le variabili esistono nel dataframe corrente e restituisce l'elenco (Array) delle solo variabili esistenti.

```
1 cat existing(:v1, :v2, :v3)
2 md existing(s(:d7_, 99))
```

22.1.3 varnames

`varnames(varlist)` restituisce un vettore di nomi di variabili partendo dall'elenco di variabili dato ed espandendo gli eventuali net.

22.1.4 size

`size(varlist)` restituisce il numero di elementi partendo dall'elenco di variabili dato ed espandendo gli eventuali net.

22.1.5 usedvars

`usedvars` restituisce un vettore con i nomi di tutte le variabili utilizzate nelle tavole (pannelli e fiancate).

```
1 # esporta un file contenente solo le variabili utilizzate nella tabulazione
2 excelx.save "variabili_usate", :vars => usedvars, :as_num => false
```

22.1.6 allvars

`allvars` restituisce un vettore con i nomi di tutte le variabili presenti nel file di dati.

Il parametro `:sets => true` restituisce i *set* di variabili, dove presenti, al posto delle singole variabili.

```
1 # esporta un file contenente solo le variabili NON utilizzate nella tabulazione
2 excelx.save "variabili_non_usate", :vars => allvars-usedvars, :as_num => false
```

22.2 Etichette e livelli

22.2.1 factor?

`factor?(varname)` restituisce `true` se la variabile è una variabile `factor`, altrimenti restituisce `false`.

22.2.2 get_label

`get_label(varname)` restituisce la stringa con l'etichetta della variabile.

22.2.3 get_labels

`get_labels(varlist)` restituisce un vettore con le etichette delle variabili.

```
1 get_labels(s(:d7,10)).rep do |label|
2   ...
3 end
```

22.2.4 get_levels

`get_levels(varname)` restituisce un vettore con le etichette dei livelli della variabile.

```
1 get_levels(:d1).rep do |label, index|
2   filter :d1 => index, :label => label
3   ord :d5
4 end
```

22.2.5 get_codes_and_levels

`get_codes_and_levels(varname)` restituisce un hash con i codici e le etichette dei livelli della variabile.

```
1 get_codes_and_levels(:area).each do |code, lab|
2   filter :area => code, :label => "Area: "+lab
3   cat :d1
4   ord :d5
5 end
```

22.2.6 get_values

`get_values(varname)` restituisce un array con i valori presenti nella variabile. I valori sono restituiti come valori decimali.

```
1 p get_values(:eta)
2 [1.0, 2.0, 3.0, 4.0]
```

22.2.7 get_max_level

`get_max_level(varname)` restituisce il numero totale dei livelli della variabile. Equivale a `get_max_level(varname).size`.

22.2.8 info

`info(varname)` restituisce un oggetto `PTables::VariableInfo` contenente varie informazioni relative alla variabile o al set di variabili.

Metodi dell'oggetto `VariableInfo`:

- **name**: restituisce il nome della variabile
- **set?**: restituisce `true` se la variabile è un set di variabili
- **type**: restituisce il tipo di variabile: `:numeric`, `:factor`, `:alpha`
- **numeric?**: restituisce `true` se la variabile è numerica
- **factor?**: restituisce `true` se la variabile è factor
- **alpha?**: restituisce `true` se la variabile è alfanumerica
- **multi?**: restituisce `true` se la variabile è multipla
- **label**: restituisce l'eticheta della variabile
- **levels**: se la variabile è una factor restituisce le etichette dei valori
- **values**: restituisce i valori della variabile
- **range**: se la variabile è numerica restituisce il range dei valori
- **print**: stampa le informazioni nell'output

```
1 i5 = info(:d5)
2 md i5 if i5.multi?
3
4 info(m7).print
```

22.2.9 df?

`df?` o `show_dataframe` stampa nell'output la struttura del data frame.

```
1 df?
2
3 'data.frame':      300 obs. of  14 variables:
4 $ ser_no      : int  1 2 3 4 5 6 7 8 9 10 ...
5 $ area       : int  1 1 1 1 1 1 1 1 1 1 ...
6 $ sesso     : int  1 2 1 2 2 2 2 2 2 1 ...
7 $ eta       : int  3 2 4 3 3 4 3 2 2 4 ...
8 $ titolo    : int  3 3 4 4 3 2 3 3 4 3 ...
9 $ prof      : int  3 1 1 4 2 4 1 2 1 1 ...
10 $ dc        : int  4 4 2 2 2 2 6 4 4 1 ...
11 $ d1       : int  3 4 4 2 2 4 4 4 3 4 ...
12 $ d2       : int  1 3 3 1 1 3 3 1 1 1 ...
13 $ d5       : int  NA NA NA NA NA NA NA NA NA ...
14 $ w1       : Factor w/ 4 levels " ,25","1,00",...: 3 4 4 2 2 4 4 4 3 4 ...
15 $ let      : Factor w/ 2 levels "A","B": 1 2 1 2 1 2 1 2 1 2 ...
16 $ ._tot    : Factor w/ 1 level " ": 1 1 1 1 1 1 1 1 1 1 ...
17 $ ._fake.weight: num  1 1 1 1 1 1 1 1 1 1 ...
18 - attr(*, "variable.labels")= Named chr  "ser_no" "area" "sesso" "eta" ...
19 ..- attr(*, "names")= chr  " " " " " " ...
20 - attr(*, "value.labels")= list()
```

22.3 Dati

22.3.1 list

`list` stampa nell'output i valori delle variabili dei record del data frame.

- **vars**: elenco variabili
- **rows**: numero di riga o intervallo di righe (ignorato se specificato `:if`)
- **if**: filtro per selezionare un sottoinsieme di casi (*Espressioni*)
- **numeric**: (`false|true`) converte i valori delle variabili in valori numerici (default `false`)

```
1 list :vars => [:ser_no, :sesso, :eta, :etaq, :area, s(:d7_, 4)], :rows => 1..10
```

	ser_no	sesso	eta	etaq	area	d7_1	d7_2	d7_3	d7_4
1	1	Maschio	25-30	anni	25 Nord Ovest	1	1	1	0
2	2	Femmina	18-24	anni	24 Nord Ovest	0	0	1	0
3	3	Maschio	31-34	anni	31 Nord Ovest	1	0	0	1
4	4	Femmina	25-30	anni	25 Nord Ovest	1	1	0	0
5	5	Femmina	25-30	anni	30 Nord Ovest	1	1	1	0
6	6	Femmina	31-34	anni	33 Nord Ovest	1	0	0	0
7	7	Femmina	25-30	anni	30 Nord Ovest	1	1	1	1
8	8	Femmina	18-24	anni	20 Nord Ovest	0	0	1	1
9	9	Femmina	18-24	anni	24 Nord Ovest	1	0	1	0
10	10	Maschio	31-34	anni	32 Nord Ovest	1	1	0	0

```
1 list :vars => [:ser_no, :sesso, :etaq, :d1], :if => "#sesso==1 & etaq>=33", :numeric => true
```

	ser_no	sesso	etaq	d1
15	5	1	34	1
37	37	1	34	3
114	117	1	34	4
122	125	1	33	4
138	141	1	34	4
192	195	1	33	3
226	229	1	34	4
240	243	1	33	4
257	260	1	33	4
279	282	1	34	3
281	284	1	33	4
296	299	1	33	3
299	302	1	34	4
300	303	1	33	3

22.3.2 get_base

`get_base(varname, value)` restituisce il numero di casi pesati di una variabile o di un suo livello (se specificato `value`).

Se non si specifica `varname` verrà restituito il numero totale di casi pesati del dataframe.

```
1 base_d5 = get_base(:d5)
2 if base_d5 > 100
3   ord :d5
4 end
5
6 ord :d1 if get_base(:d1, 2) > 30
```

22.3.3 get_base_u

`get_base_u(varname, value)` restituisce il numero di casi non pesati di una variabile o di un suo livello (se specificato `value`).

Se non si specifica `varname` verrà restituito il numero totale di casi non pesati del dataframe.

22.3.4 get_counts

`get_counts(varname [, :drop])` restituisce i livelli di una variabile con le distribuzioni di frequenza ponderate (se è attiva una variabile di ponderazione). Il parametro `:drop` elimina i livelli con frequenze pari a zero. `get_counts` restituisce un vettore con un elemento per ciascun livello della variabile costituito a sua volta da un vettore di tre elementi: codice livello, etichetta livello, conteggio frequenze:

```
p get_counts(:sesso)
[[1, "Maschio", 154.0], [2, "Femmina", 146.0]]
```

```
1 get_counts(:d1, :drop).each do |code, lab, count|
2   filter :d1 => code, :label => "D1: #{lab} (n.#{count.round})"
3   ord :d5
4 end
```

22.3.5 get_counts_u

`get_counts(varname [, :drop])` funziona in modo analogo a `get_counts`, ma restituisce sempre le distribuzioni di frequenza non ponderate.

22.3.6 xtab

`xtab(varname[, by: varname])` calcola una tabella con le distribuzioni di frequenza di una variabile (*variabile di riga*). Specificando un'ulteriore variabile con il parametro `by` (*variabile di colonna*) calcola una tabella di contingenza.

Importante: Se è attiva una variabile di ponderazione, `xtab` calcola i dati **ponderati**. Per produrre dati *non ponderati* usare `xtab_u`.

Se è attivo un **filtro**, `xtab` restituisce i dati filtrati, cioè solo per i casi attivi.

Opzioni:

- **by:** specifica un nome di variabile da incrociare
- **drop:** elimina i livelli/valori con frequenze pari a zero (default: `false`)
- **decimal:** numero di decimali delle percentuali (default: valore impostato nel parametro di configurazione `:frm_pct`)
- **count_dec:** numero di decimali dei valori assoluti (default: valore impostato nel parametro di configurazione `:frm_count`)
- **if:** filtro per selezionare un sottoinsieme di casi (*Espressioni*)

`xtab` restituisce un oggetto che dispone di vari metodi per estrarre i dati.

Metodo	Tipo dato restituito	Descrizione
nrows	Integer	Numero righe tabella
ncols	Integer	Numero colonne tabella
size	Array	Dimensioni tabella (righe, colonne)
codes	Array	Valori/codici variabile
label	String	Etichetta variabile riga
labels	Array	Etichette delle modalità della variabile di riga
clabel	String	Etichetta variabile colonna
clabels	Array	Etichetta delle modalità della variabile di colonna
data (col=nil, simplify=true)	Array	Array con le serie di dati della tabella (una serie per il totale e una per ciascuna modalità della variabile di colonna)
counts (col=nil)	Array	Valori assoluti
pcts (col=nil)	Array	Percentuali di colonna
tcunts (simplify=true)	Array	Totali colonne
tpcts (simplify=true)	Array	Percentuale dei totali delle colonne
print	Object	Stampa nell'output una versione testuale della tabella (restituisce lo stesso oggetto)

I metodi `data`, `counts` e `pcts` accettano come argomento il numero di colonna dei dati da estrarre (0 è la colonna del totale) o l'etichetta della colonna. I vettori di dati restituiti potranno quindi essere costituiti da una o più serie di dati.

```

1 x = xtab(:d1, by: :sesso).print
2 p x.counts
3 # [[27, 43, 56, 170, 4], [12, 25, 32, 84, 0], [15, 17, 23, 81, 4]]
4 p x.counts(1)
5 # [12, 25, 32, 84, 0]
6 p x.pcts
7 # [[9.0, 14.3, 18.7, 56.7, 1.3], [7.8, 16.3, 20.9, 54.9, 0], [10.7, 12.1, 16.4, 57.9, 2.9]]
8 p x.pcts('maschio')
9 # [7.8, 16.3, 20.9, 54.9, 0]

```

I dati sono *semplificati* in modo da restituire il tipo più semplice di dato. Utilizzare `false` per non semplificare i dati.

```

1 p xtab(:d1, by: :sesso).tcunts
2 # <Array> [300, 153, 140]
3
4 p xtab(:d1).tcunts
5 # <Fixnum> 300
6
7 p xtab(:d1).tcunts(false)
8 # <Array> [300]

```

Il metodo `data` restituisce un array per ciascuna serie di dati contenente un array costituito da una struttura con quattro campi per ciascuna riga: `code`, `label`, `count`, `pct`:

```

1 x = xtab(:sesso)
2 p x.data
3 # [#<struct PTables::Counts::Row code=1, label="Maschio", count=153, pct=52.2>,
4 # #<struct PTables::Counts::Row code=2, label="Femmina", count=140, pct=47.8>]

```

Esempi utilizzo:

- ripetizione sui valori di una variabile con controllo della numerosità e indicazione della numerosità nell'etichetta del filtro:

```

1 xtab(:d1).data.each do |x|
2   next if x.count < 50

```

```

3   filter :d1 => x.code, :label => "D1: #{x.label} (n.#{x.count})"
4   ord :d5, "D5.#{x.code}"
5   end

```

print:

```
1 xtab(:d1, by: :sesso).print
```

Con quale frequenza frequenza di solito vai...?

by Sesso

Value	Label	TOTALE	%	Maschio	%	Femmina	%
1	una volta alla settimana o più spesso	27	9.0	12	7.8	15	10.0
2	una volta ogni due settimane	43	14.3	25	16.3	17	12.0
3	una volta ogni tre settimane	56	18.7	32	20.9	23	16.0
4	una volta al mese	170	56.7	84	54.9	81	57.0
5	meno spesso	4	1.3	0	0	4	2.0
	Totale	300	100.0	153	51.0	140	46.0

22.3.7 xtab_u

`xtab_u(varname[, by: varname])` funziona in modo analogo a `xtab`, ma usa dati **non ponderati**.

22.3.8 get_data

`get_data(varlist, options={})` restituisce dei vettori con i dati delle variabili indicate.

Opzioni:

- **as_num**: converte i dati in numerici (default `false`)
- **jitter**: aggiunge “rumore” ai dati numerici (`true` oppure un valore, default `false`)
- **if**: filtro per selezionare un sottoinsieme di casi (*Espressioni*)

22.3.9 get_last_id

`get_last_id` restituisce l’id dell’ultima tabella generata.

22.3.10 valid?

`valid?(varname)` restituisce `true` se la variabile contiene casi validi, altrimenti restituisce `false`.

22.3.11 missing?

`missing?(varname)` restituisce `true` se la variabile contiene solo casi mancanti, altrimenti restituisce `false`.

22.3.12 get_selection

`get_selection(expression)` restituisce il numero di casi pesati selezionati dall’espressione.

Vedi anche *Espressioni e parse*

22.3.13 get_selection_u

get_selection_u(expression) restituisce il numero di casi non pesati selezionati dall'espressione.

Vedi anche *Espressioni e parse*

22.4 Data e ora

L'oggetto **time** contiene la data e l'ora corrente. E' possibile estrarre alcuni dati applicando vari metodi:

- *year*: anno
- *month*: numero del mese
- *day*: numero del giorno del mese
- *wday*: numero del giorno della settimana
- *yday*: numero del giorno dell'anno
- *today(format)*: stringa con la data corrente; formato di default: %Y%m%d
- *now(format)*: stringa con la data e l'ora corrente; formato di default: %Y%m%d%H%M%S
- *days*: vettore con il nome dei giorni della settimana in inglese
- *giorni*: vettore con il nome dei giorni della settimana in italiano
- *months*: vettore con il nome dei mesi in inglese
- *mesi*: vettore con il nome dei mesi in italiano
- *freeze!*: blocca la data e l'ora anche per le successive richieste all'oggetto time

```
1 # se è il 31/12/2012 23:59:59
2 spss.open "data_{time.year}" # => data_2012
3 spss.open "data_{time.today}" # => data_20121231
4 spss.open "data_{time.now}" # => data_20121231235959
5 spss.open "data_{time.year}_" + time.mesi[time.wday][0..2] # => data_2012_dic
6 spss.open "data" + time.year.to_s + time.mesi[time.wday][0..2].upcase # => data2012DIC
```

22.5 Funzioni per i titoli delle tabelle

22.5.1 \$vl()

\$vl(varname)¹ inserisce nel titolo della tabella l'etichetta della variabile. L'argomento *varname* è facoltativo, se omesso viene considerata la prima variabile della tabella. Altrimenti si può utilizzare qualsiasi nome di variabile esistente nel file di dati, anche se non fa parte della tabella. Il nome deve essere inserito senza virgolette. Le parentesi vanno comunque inserite.

Le varianti \$VL(), \$Vl() mettono l'etichetta rispettivamente in maiuscolo e con la prima lettera maiuscola.

```
1 cat :d12_1, "Esprima un giudizio relativamente ai seguenti personaggi;\n$vl()"
2
3 ord [:aggr12_1, :d12_1], "Esprima un giudizio relativamente ai seguenti personaggi;\n$VL(d12_1)"
```

¹ \$vl() sostituisce \$var_label() che viene comunque ancora accettato.

22.5.2 \$vn()

`$vn(start)` inserisce nel titolo della tabella il nome della variabile. Le varianti `$VN()`, `$Vn()` mettono il nome rispettivamente in maiuscolo e con la prima lettera maiuscola. Se sono presenti più variabili nella tabella, cercherà di inserire la radice comune. `start` è facoltativo e indica il numero di carattere del nome da cui partire. Il comando è utile per i parametri `:title_template` e `:title_prefix`.

```
| title_template "Tab.$tn() - $VN() - $v1()"
```

22.5.3 \$tn()

`$tn(offset)` inserisce nel titolo della tabella il numero progressivo della tavola. `offset` è facoltativo e deve essere un numero positivo o negativo che verrà sommato al numero della tavola. Il comando è anche utile per i parametri `:title_template` e `:title_prefix`.

```
| title_prefix "Tabella $tn(-1) : "
```

Note

Ponderazione

23.1 weight

Pondera i dati in base a una variabile di pesi. ¹

```
1 # pondera i dati sulla variabile ipf
2 weight! :ipf
3
4 # disattiva la ponderazione dei dati
5 weight_
```

E' possibile indicare una variabile di ponderazione differente per ciascuna variabile del pannello. Eventuali variabili di pannello duplicate devono essere considerate una sola volta.

```
1 panels.add :_t, :sesso, :eta
2 panels.add :_t, :area, :titolo
3
4 # solo 5 variabili di ponderazione perché :_t è indicata due volte
5 weight! :w1, :w2, :w3, :w4, :w5
```

23.1.1 Pannello di confronto tra ponderazioni

E' possibile realizzare una tabella con un pannello che mette a confronto ponderazioni differenti.

- Metodo 1 con *htjoin*. Utile quando si deve produrre solo una tavola.

```
1 # :ipf1, :ipf2 e :ipf3 sono tre diverse variabili di ponderazione
2
3 # quattro tabelle con ponderazione differente unite in un'unica tabella tramite htjoin
4 weight_
5 htjoin "Tabella di confronto tra ponderazioni", "PONDERAZIONI" do
6   cat :voto, "Non ponderato"
7   weight! :ipf1
8   cat :voto, "Peso 1"
9   weight! :ipf2
10  cat :voto, "Peso 2"
11  weight! :ipf3
12  cat :voto, "Peso 3"
13 end
14
15 # equivalente dell'esempio precedente, ma realizzato con un ciclo di ripetizione
16 weight_
17 htjoin "Tabella di confronto tra ponderazioni", "PONDERAZIONI" do
18   4.times do |i|
```

¹ I nuovi metodi sono da preferire rispetto a `weight.by` e `weight.off`, in quanto possono essere utilizzati anche all'interno delle istruzioni `vjoin`, `hjoin` e `htjoin`.

```

19   cat :voto, i == 0 ? "Non ponderato" : "Peso #{i}"
20   weight! "ipf#{i+1}"
21   end
22 end

```

- Metodo 2 con pannello esplicito e corrispondenti variabili di ponderazione. Utile quando questo pannello va utilizzato per tante tavole.

```

1  # il peso per la colonna "Non ponderato" è una variabile inizializzata a 1: :ipf0
2  # il pannello è costituito da variabili inizializzate a 1: :t0, :t1, :t2, :t3
3  create :ipf0, :t0, :t1, :t2, :t3
4  value [:ipf0, :t0, :t1, :t2, :t3], 1
5  factor :t0 => ["PONDERAZIONI", ["Non ponderato"]], # oppure se si vuole una colonna separata d
6         :t1 => ["PONDERAZIONI", ["Peso 1"]],
7         :t2 => ["PONDERAZIONI", ["Peso 2"]],
8         :t3 => ["PONDERAZIONI", ["Peso 3"]]
9
10 panels.add :t0, :t1, :t2, :t3
11 weight! :ipf0, :ipf1, :ipf2, :ipf3
12
13 # in questo modo è molto semplice generare tutte le tabelle seguenti con questo pannello
14 cat :voto
15 ord :d5
16 md s(:d7_, 1..10)

```

23.2 wby

wby è un sottocomando delle tabelle che permette di specificare la/le variabile/i di ponderazione per una specifica tabella.

```

1  cat :d1 do
2    wby :peso1
3  end

```

23.3 weight.action

weight.action permette di indicare quale azione intraprendere nel caso di mancata corrispondenza tra il numero di variabili di pannello e il numero di variabili di ponderazione (nel caso siano più di una).

- **:error**: genera un errore in caso di mancata corrispondenza e arresta il programma
- **:message**: stampa un messaggio di avvertimento nel log e scarta eventuali variabili in eccesso e riutilizza l'ultima variabile di ponderazione per le variabili di pannello aggiuntive
- **:none**: come *message* ma non stampa nessun avvertimento

Il comportamento predefinito è indicato nel parametro di configurazione :weights_vars_action.

```

1  weight.action :message
2  panels.add :_t, :sesso, :eta
3  weight.by :w1, :w2, :w3, :w4
4  cat :d1

```

- Il numero di variabili di ponderazione è superiore al numero di variabili di pannello,
- le variabili in più verranno escluse:
- `._tot => w1; sesso => w2; eta => w3`

```

1  weight.action :message
2  panels.add :_t, :sesso, :eta

```

```
3 weight.by :w1, :w2
4 cat :d1
```

- Il numero di variabili di ponderazione è inferiore al numero di variabili di pannello,
- l'ultima variabile di ponderazione verrà utilizzata anche per le restanti variabili di pannello
- `._tot => w1; sesso => w2; eta => w2`

```
1 weight.action :error
2 panels.add :_t, :sesso, :eta
3 panels.add :_t, :area
4 weight.by :w1, :w2
5 cat :d1
```

- Errore nella sintassi del comando `<weight!>`: Il numero di variabili di ponderazione non corrisponde
- (2) `w1, w2`
- (4) `._tot, sesso, eta, area.`

23.4 rake

Genera delle variabili di ponderazione dei dati in base alle distribuzioni marginali.

Package R: `Survey`

Eventuali variabili già esistenti verranno sovrascritte.

Se l'istruzione **filter** è attiva, *rake* opererà solo sul sottoinsieme di casi.

23.4.1 Parametri

- **names:** la lista dei nomi delle variabili di ponderazione che devono essere create: la variabile di ponderazione con media 1, più eventuali altre variabili di espansione. (default: `ipf`)
- **counts:** la lista delle numerosità totali dei casi delle variabili di espansione.
- **population:** un hash con i marginali delle variabili di ponderazione, nel formato: `nome_var => [marginali]`. I valori vengono interpretati come proporzioni. Le variabili non devono avere casi mancanti. Il numero di marginali di ciascuna variabile deve corrispondere al numero di livelli della variabile.
- **pw:** il nome della variabile peso di partenza.
- **if_na:** comportamento in caso di valori mancanti nelle variabili di ponderazione:
 - **:error:** (il default) interrompe l'esecuzione
 - **:na:** assegna peso uguale a NA (*valore mancante*)
 - **valore:** assegna peso uguale a *valore*
- **if:** filtro per selezionare un sottoinsieme di casi da ponderare². Ha il sopravvento rispetto a un eventuale filtro impostato con l'istruzione **filter**.
- **if_not:** valore del peso da assegnare ai casi esclusi dal filtro (altrimenti viene lasciato il valore preesistente o NA se la variabile di ponderazione non esiste già)
- **xls:** esporta le variabili di ponderazione create in un file Excel 97-2003
- **xlsx:** esporta le variabili di ponderazione create in un file Excel 2007
- **sav:** esporta le variabili di ponderazione create in un file SPSS
- **xpt:** esporta le variabili di ponderazione create in un file XPORT per SAS

² Per le espressioni vedi anche *Espressioni*

- **fwf**: esporta le variabili di ponderazione create in un file di testo in formato fisso (*Fixed Width Format*). Oltre ai dati viene salvato anche un file con il tracciato record (.info).
- **id**: specifica l'identificativo dei record da utilizzare nel file di esportazione. I dati saranno ordinati sull'id.
- **print**: :all|:brief|:none controlla la stampa nel log dell'output. (default: :all)
- **maxit**: controlla il numero di iterazioni. (default: 10)
- **epsilon**: convergence is declared if the maximum change in a table entry is less than epsilon. If $\epsilon < 1$ it is taken to be a fraction of the total sampling weight. (default: 0.001)

```

1 # genera solo il peso per la ponderazione
2 rake :names => :peso,
3     :population => { :sesso => [150, 150], :eta => [25, 75, 150, 50] },
4     :xls => "ipf", :id => :ser_no
5
6 # genera 3 variabili di ponderazione
7 rake :names => [:ipf, :espcam, :espuni],
8     :counts => [320, 45000],
9     :population => { :sesso => [150, 150],
10                    :eta => [25, 75, 150, 50] ,
11                    :area => [75, 75, 75, 75] },
12     :print => :all,
13     :id => :ser_no,
14     :fwf => "ipf", :sav => "ipf"
15
16 # pondera in due step i dati
17 rake :names => :ipf, ... :if => {:cambio => 1}
18 rake :names => :ipf, ... :if => {:cambio => 2}

```

23.4.2 Output

*** Raking ***

	#	n.	-ideale-	%	n.	-sample-	%	(s.-i.)	n.	-pond.-	%	(p.-i.)
Maschio	1	150	50.00	154	51.33	1.33333	150	50.04	0.04219			
Femmina	2	150	50.00	146	48.67	-1.33333	150	49.96	-0.04219			
=== totale sesso	0	300	100.00	300	100.00	0.00000	300	100.00	0.00000			
14-17 anni	1	25	8.33	46	15.33	7.00000	25	8.34	0.01011			
18-24 anni	2	75	25.00	86	28.67	3.66667	75	24.96	-0.03623			
25-30 anni	3	150	50.00	116	38.67	-11.33333	150	50.01	0.01179			
31-34 anni	4	50	16.67	52	17.33	0.66667	50	16.68	0.01433			
=== totale eta	0	300	100.00	300	100.00	0.00000	300	100.00	0.00000			
Nord Ovest	1	75	25.00	93	31.00	6.00000	75	25.00	0.00000			
Nord Est	2	75	25.00	50	16.67	-8.33333	75	25.00	0.00000			
Centro	3	75	25.00	70	23.33	-1.66667	75	25.00	0.00000			
Sud	4	75	25.00	87	29.00	4.00000	75	25.00	0.00000			
=== totale area	0	300	100.00	300	100.00	0.00000	300	100.00	0.00000			

	#	n	sum	min	max	range	median	mean	sd	se	esp
ipf	1	300	300	0.4031	2.0312	1.6281	0.9916	1.0000	0.3725	0.0215	1.0000
espcam	2	300	320	0.4300	2.1666	1.7367	1.0577	1.0667	0.3973	0.0229	1.0667
espuni	3	300	4500	6.0464	30.4682	24.4218	14.8737	15.0000	5.5869	0.3226	15.0000

Efficienza: 87.9 %

23.4.3 Valori di ritorno

rake restituisce un oggetto contenente le statistiche della ponderazione, che, se assegnato a una variabile, può essere utilizzato nello script per condizionarne l'esecuzione. L'oggetto dispone dei metodi :efficiency, :n,

:sum, :min, :max, :range , :median, :mean, :sd, :se, :esp che restituiscono le statistiche relative alla prima variabile di ponderazione. E' inoltre possibile accedere alle statistiche delle altre variabili di ponderazione con la sintassi `variabile["nome_peso"]`: per es. `result["ipf2"].esp`

```
1 # prima ponderazione
2 result = rake :names ...
3 if result.min < 0.3 || result.max > 3
4   # nuova ponderazione, con meno variabili marginali o con classi più aggregate
5   result = rake :names ...
6 end
7 # se i valori non sono ancora soddisfacenti
8 raise "Attenzione: valori dei pesi troppo alti: #{result.min} <> #{result.max}" if result.min < 0.3 || result.max > 3
9
10 # oppure considerando l'efficienza
11 if result.ency < 60
12   ...
```

Note

Test statistici e accentuazioni

24.1 Chi-quadrato

`chisq_test!` attiva il test del **chi-quadrato** per le tabelle di tipo *cat*, *ord*, *md* e *grid*.

	TOTALE	Area geografica				Sesso		Sesso: Eta'							
		Nord Ovest	Nord Est	Centro	Sud	Maschio	Femmina	Maschio: 14-17 anni	Maschio: 18-24 anni	Maschio: 25-30 anni	Maschio: 31-34 anni	Femmina: 14-17 anni	Femmina: 18-24 anni	Femmina: 25-30 anni	Femmina: 31-34 anni
Totale	300	75	75	75	75	150	150	10	39	75	26	15	36	76	24
% row	100,0%	25,0%	25,0%	25,0%	25,0%	50,1%	49,9%	3,4%	13,0%	24,8%	8,8%	4,9%	11,9%	25,2%	7,9%
una volta alla settimana o più spesso	32	5	10	9	9	13	18	0	3	7	2	1	3	15	-
% col	10,7%	6,1%	13,5%	11,6%	11,4%	9,0%	12,3%	4,7%	8,6%	9,6%	9,4%	3,5%	9,6%	19,3%	-
una volta ogni due settimane	45	18	3	17	7	25	20	0	5	15	5	3	3	13	1
% col	15,0%	24,4%	4,4%	22,1%	9,0%	16,7%	13,3%	3,9%	13,1%	19,5%	19,3%	17,5%	7,1%	17,6%	5,9%
una volta ogni tre settimane	56	16	23	6	11	33	23	2	9	15	7	3	6	9	5
% col	18,7%	21,8%	30,8%	8,0%	14,2%	21,9%	15,5%	17,8%	22,1%	20,2%	28,3%	23,1%	16,5%	12,4%	19,0%
una volta al mese	167	36	38	44	49	79	88	8	22	38	11	8	24	38	18
% col	55,7%	47,7%	51,3%	58,2%	65,4%	52,4%	58,9%	73,6%	56,2%	50,8%	43,0%	55,9%	66,9%	50,7%	75,1%
meno spesso	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
% col	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
χ^2	-	0,000 *				0,000 *		0,000 *							
df	-	12				4		28							
Sig.	-	0,000				0,000		0,000							

* sono presenti valori attesi ≤ 1 , il valore di χ^2 potrebbe non essere corretto

E' possibile impostare i seguenti parametri:

- **chisq_labels**: etichette: chi-quadrato, gradi di liberta', significativita'.
- **chisq_formats**: formati: chi-quadrato, gradi di liberta', significativita'.
- **chisq_warning_cel_freq**: avvertimento celle con numerosita' bassa.

```
1 chisq_test!
2 cat :d1
3 ord :d5
4 chisq_test_
```

Funzione R: `chisq.test {stats}`

24.2 Pairwise Comparisons: medie

`pcm_test!` attiva il **t-test** delle medie per le tabelle di tipo *scale* e *ord*.

	TOTALE	Area geografica				Sesso		Sesso: Eta'								
		Nord Ovest	Nord Est	Centro	Sud	Maschio	Femmina	Maschio: 14-17 anni	Maschio: 18-24 anni	Maschio: 25-30 anni	Maschio: 31-34 anni	Femmina: 14-17 anni	Femmina: 18-24 anni	Femmina: 25-30 anni	Femmina: 31-34 anni	
		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	(m)	(n)	(o)
Totale	274	49	75	75	75	138	136	10	35	70	23	14	31	68	23	
	% row	18,0%	27,3%	27,3%	27,3%	50,4%	49,6%	3,7%	12,7%	25,4%	8,5%	5,1%	11,4%	24,7%	8,4%	
Non risponde		20	3	5	4	8	6	14	1	1	4	-	1	2	10	2
	% col	7,4%	6,6%	6,5%	5,6%	10,8%	4,6%	10,4%	8,4%	3,6%	6,0%	-	7,1%	5,5%	14,4%	7,3%
Per niente d'accordo		19	8	9	3	-	9	10	1	1	5	1	-	1	7	2
	% col	7,0%	15,9%	11,8%	3,4%	-	6,3%	7,7%	8,8%	3,3%	7,6%	6,0%	-	4,1%	10,2%	10,0%
Poco d'accordo		90	19	30	24	16	52	38	1	13	29	8	3	12	14	9
	% col	32,8%	39,2%	40,1%	32,1%	21,9%	37,5%	28,0%	14,1%	38,5%	41,1%	35,2%	20,0%	38,8%	20,5%	40,0%
Abbastanza d'accordo		96	15	30	32	18	45	51	6	9	19	10	8	10	27	5
	% col	34,9%	30,4%	40,1%	43,2%	24,2%	32,3%	37,4%	55,2%	26,4%	27,8%	44,7%	58,2%	33,1%	40,3%	22,4%
Molto d'accordo		49	4	1	12	32	27	22	1	10	12	3	2	6	10	5
	% col	17,9%	7,8%	1,5%	15,8%	43,1%	19,4%	16,5%	13,5%	28,2%	17,5%	14,1%	14,7%	18,5%	14,7%	20,2%
	base	274	49	75	75	75	138	136	10	35	70	23	14	31	68	23
	media	3,49	3,17	3,18	3,60	3,89	3,56	3,42	3,57	3,72	3,43	3,67	3,73	3,55	3,31	3,38
	pairwise comparisons			bde	be	b				hl	hilmo	hl		hl	hilmo	hl
	dev. standard	1,10	1,02	0,90	0,98	1,28	1,02	1,17	1,15	1,04	1,06	0,81	0,99	1,03	1,26	1,16

pairwise comparisons: test a una coda; livello di significativita' 0,5; metodo di correzione Bonferroni

E' possibile impostare i seguenti parametri:

- **label_pcm_test**: etichetta Pairwise Comparisons
- **pcm_test_sig**: array con i livello di significativita' t-test (max 9 livelli)
- **pcm_test_tails**: 2=due code; 1=una coda
- **pcm_test_minbase**: numero minimo di casi per eseguire il test
- **pcm_test_note**: attiva/disattiva le note
- **pcm_test_adj**: metodo di correzione p-value:
 - :holm - Holm
 - :hoch - Hochberg
 - :hommel - Hommel
 - :bonf - Bonferroni
 - :bh o fdr - Benjamini & Hochberg
 - :by - Benjamini & Yekutieli
 - :none - *nessuno*

L'istruzione `pcm_test([sig [,adj [,tail]]])` attiva il t-test per le medie e imposta il livello di significatività, il metodo di correzione e il tipo di coda.

```
1 pcm_test 0.5, :bonf
2 scale :espo2b
3 ord :d5
4 pcm_test_
```

Funzione R: `t.test {stats}`

24.3 Pairwise Comparisons: proporzioni

`pcp_test!` attiva il **test sulle proporzioni**.¹

E' possibile impostare i seguenti parametri:

- **label_pcp_test**: etichetta Pairwise Comparisons
- **pcp_test_sig**: array con i livelli di significativita' t-test (max 9 livelli)
- **pcp_test_tails**: 2=due code; 1=una coda

¹ Metodo non parametrico basato sul test del chi-quadro.

- **pcp_test_minbase**: numero minimo di casi per eseguire il test
- **pcp_test_note**: attiva/disattiva le note
- **pcp_test_adj**: metodo di correzione p-value:
 - :holm - Holm
 - :hoch - Hochberg
 - :homm - Hommel
 - :bonf - Bonferroni
 - :bh o fdr - Benjamini & Hochberg
 - :by - Benjamini & Yekutieli
 - :none - *nessuno*
- **pcp_yates_corr** attiva/disattiva la correzione di Yates (correzione per la continuità)

L'istruzione `pcp_test([sig [,adj [,tail [,yates]]])` attiva il t-test per le percentuali e imposta il livello di significatività, il metodo di correzione, il tipo di coda e la correzione di Yates.

```
1 pcp_test [0.10,0.05,0.01]
2 cat :d1
3 ord :d5
4 pcp_test_
```

Funzione R: `prop.test {stats}`

24.4 Accentuazioni

`phl!` e `mhl!` attivano rispettivamente le accentuazioni sulle distribuzioni e sulle medie.

	TOTALE	Area geografica				Sesso		Sesso: Eta'							
		Nord Ovest	Nord Est	Centro	Sud	Maschio	Femmina	Maschio: 14-17 anni	Maschio: 18-24 anni	Maschio: 25-30 anni	Maschio: 31-34 anni	Femmina: 14-17 anni	Femmina: 18-24 anni	Femmina: 25-30 anni	Femmina: 31-34 anni
Totale	270	63	50	70	87	140	130	19	42	55	24	25	33	49	23
NON RISPONDE	% row 20	4	3	4	9	7	13	2	2	3	-	1	2	8	2
	% col 7,4%	6,3%	6,0%	5,7%	10,3%	5,0%	10,0%	10,5%	4,8%	5,5%	-	4,0%	6,1%	16,3%	8,7%
NEGATIVO	% col 99	32	24	23	20	56	43	5	16	24	11	6	12	14	11
(0) Per niente d'accordo	% col 16	9	5	2	-	9	7	2	1	4	2	-	1	4	2
(25) Poco d'accordo	% col 83	23	19	21	20	47	36	3	15	20	9	6	11	10	9
	% col 30,7%	36,5%	38,0%	30,0%	23,0%	33,6%	27,7%	15,8%	35,7%	36,4%	37,5%	24,0%	33,3%	20,4%	39,1%
POSITIVO	% col 151	27	23	43	58	77	74	12	24	28	13	18	19	27	10
(50) Abbastanza d'accordo	% col 55,9%	42,9%	46,0%	61,4%	66,7%	55,0%	56,9%	63,2%	57,1%	50,9%	54,2%	72,0%	57,6%	55,1%	43,5%
(100) Molto d'accordo	% col 95	22	22	31	20	46	49	9	11	17	9	14	11	19	5
	% col 35,2%	34,9%	44,0%	44,3%	23,0%	32,9%	37,7%	47,4%	26,2%	30,9%	37,5%	56,0%	33,3%	38,8%	21,7%
	% col 56	5	1	12	36	31	25	3	13	11	4	4	8	8	5
	% col 20,7%	7,9%	2,0%	17,1%	43,7%	22,1%	19,2%	15,8%	31,0%	20,0%	16,7%	16,0%	24,2%	16,3%	21,7%
base	250	59	47	66	78	133	117	17	40	52	24	24	31	41	21
media	49,70	36,86	35,64	49,62	67,95	49,44	50,00	48,53	55,62	47,12	44,79	52,08	52,42	48,78	46,43
dev. standard	30,51	26,00	19,33	27,21	32,71	31,30	29,73	29,94	33,25	31,18	29,47	24,36	31,19	30,08	33,81

Pesi: non risponde = n.a., per niente d'accordo = 0, poco d'accordo = 25, abbastanza d'accordo = 50, molto d'accordo = 100

Le accentuazioni sono controllate dai seguenti parametri di configurazione:

- **phl_ic**: calcola accentuazioni in base all'indice di concentrazione e non alle percentuali di colonna.
- **phl_lev**: vettore con i limiti degli intervalli delle accentuazioni sulle distribuzioni. Si può definire un massimo di 3 valori.
- **phl_min**: numero minimo di casi nella cella.
- **mhl_lev**: vettore con i limiti degli intervalli delle accentuazioni delle medie. Si può definire un massimo di 3 valori. Impostando `:auto` utilizza i valori: 1/3 deviazione standard, 1/2 deviazione standard, 1 deviazione standard. Per impostare valori differenti in termini di deviazione standard oppure uno solo di questi valori, utilizzare la sintassi `:sd1_5`, `:sd0_3`, `:sd2`, rispettivamente 1 sdev e 1/2, 1/3 sdev, 2 sdev (l'underscore rappresenta il punto decimale)

- **mhl_min**: numero minimo di casi nella cella.

In *phl_lev* e *mhl_lev* è possibile utilizzare l'argomento `:pos` per indicare che devono essere mostrate solo le accentuazioni positive.

Le istruzioni **phl** e **mhl** permettono di attivare le accentuazioni e impostare direttamente gli intervalli.

```
1  phl 5, 10, 15
2  mhl 0.2, 2, 3
3  cat :d1
4  ord :d5
5  md s(:d7_, 10)
6
7  mhl :auto
8  scale :espo2b
9
10 mhl :sd0_3, :pos
11 scale s(:d6_, 10)
12
13 i_conc!
14 phl 5, 10, 15, :ic, :pos
15 cat :d10
16
17 phl_
18 mhl_
```

Le statistiche su cui vengono calcolate le accentuazioni (percentuali di colonna/indice di concentrazione/medie) devono essere attivate prima dei comandi *phl* e *mhl*.

24.4.1 Colori

I colori possono essere personalizzati modificando i file dei *temi di colore*. Per il formato XLS vedere vedere la tabella disponibili.

Note

Analisi delle corrispondenze

L'istruzione **ca** esegue l'analisi delle corrispondenze semplici sui risultati di una tabella.

E' necessario aver installato la libreria R **ca**.

Per eseguire l'analisi è necessario assegnare la tabella a una variabile e poi richiamarla nell'istruzione **ca**.

Opzioni disponibili nell'istruzione **ac**:

- **title**: titolo per l'analisi, altrimenti utilizza il titolo della tavola
- **stat**: statistica su cui eseguire l'analisi. Default: `:count`
- **rows**: elenco numeri righe incluse nell'analisi. Se non specificato vengono utilizzati tutti i livelli della variabile di riga.
- **cols**: elenco numeri colonne incluse nell'analisi. Se non specificato vengono utilizzate tutte le colonne dei pannelli tranne il totale
- **rsup**: elenco numeri righe supplementari
- **csup**: elenco numeri colonne supplementari
- **plot_frm**: formato file grafici: `:pdf`, `:png`, `:wmf`
- **file**: nome (senza estensione) del file dei grafici, altrimenti viene generato automaticamente (utilizzarlo specie se si produce più analisi sulla stessa tabella)
- **arrows**: punti (`false`) o frecce (`true`). Un valore per X e uno per Y

Inoltre è possibile modificare i seguenti parametri di configurazione:

Pa- ra- me- tro	Ti- po	Default	Descrizione
ca_dims	Fi- x- num	2	Numero dimensioni grafici
ca_plots	Boo- lean	true	Genera i grafici
ca_plot_frm	Sim- bol	:pdf	Formato file grafici: :pdf, :png, :wmf
ca_out	Boo- lean	true	Genera l'output testuale
ca_arrows	Ar- ray	[false, false]	Punti (false) o frecce (true). Un valore per X e uno per Y
ca_contr	Ar- ray	[:relative, :relative]	Indicazione dei contributi tramite l'intensita' dei colori: :none, :absolute, :relative. Un valore per X e uno per Y
ca_mass	Ar- ray	[true, true]	Indicazione delle masse come area dei punti. Comunque disabilitato se presenti punti supplementari. Un valore per X e uno per Y
ca_db	Boo- lean	true	Esporta coordinate nel db

```

1  d7 = md s(:d7_,10), "Domanda 7" do
2    id :d7
3    none "NON INDICA"
4  end
5  ca d7, :rsup => 11, :csup => [2,3]
6
7  d50 = grid "Domanda 50" do
8    vars ss("q50_#1_#2", 4, 4)
9    label "PRODOTTI"
10   labels "A", "B", "C", "D"
11 end
12 ca d50, title: "Immagine dei prodotti", rows: 1..8, file: "ac_d50_1"
13 ca d50, title: "Immagine dei prodotti", rows: 1..6, file: "ac_d50_2"
14
15 ca_plot_frm :wmf
16 ca_dims 3
17 d6 = htjoin "Domanda 6", "AFFERMAZIONI" do
18   s(:d6_,5).rep { |v| ord v }
19 end
20 ca d6

```

ac produce un output testuale nel file di log e i grafici con le mappe.

Output:

```

Simple correspondence analysis
=====

```

```

Domanda 7
ca_d7

```

Table data:

	Maschio	Femmina	Nord	Ovest	Nord Est	Centro	Sud
Vivere intense emozioni	117	103	70	37	56	57	
Qualcosa che non si puo' rimandare	65	64	30	17	36	46	
Occasione per conoscere altre realta'	89	89	50	31	45	52	
Momento piacevole con la famiglia	68	59	32	26	32	37	
Puro divertimento, svago	115	107	72	35	59	56	
Entrare in una dimensione di finzione e irrealta'	112	95	68	29	47	63	

Stare in compagnia degli amici	117	109	74	34	55	63
Argomento di cui parlare con gli altri	113	109	69	34	56	63
Modo per rilassarmi	118	112	74	37	56	63
Dimenticarmi del resto del mondo	93	87	56	28	42	54
NON INDICA	12	11	4	9	3	7

Indices of supplementary rows: 11
 Indices of supplementary columns: 12

Principal inertias (eigenvalues):

dim	value	%	cum%	scree plot
1	0.003263	43.3	43.3	*****
2	0.002636	35.0	78.4	*****
3	0.000873	11.6	90.0	*****
4	0.000599	8.0	97.9	*****
5	0.000100	1.3	99.3	*
6	0.000056	0.7	100.0	
7	00000000	0.0	100.0	

Total: 0.007527 100.0

Rows:

	name	mass	qlt	inr	k=1	cor	ctr	k=2	cor	ctr
1	Vvrn	113	797	36	-35	508	43	26	289	30
2	Qlcs	66	984	307	125	450	319	-136	534	469
3	Occs	92	216	71	-15	39	6	-32	177	36
4	Mmnt	65	965	276	-148	687	437	-94	278	219
5	Prdv	114	493	75	-2	1	0	49	493	106
6	Entr	107	248	70	29	164	27	20	84	17
7	Strn	116	874	80	56	600	111	38	274	63
8	Argm	114	771	15	22	491	17	16	281	12
9	Mdpr	118	753	32	-29	408	30	26	345	31
10	Dmnt	93	280	37	-19	118	10	-22	162	17
11	(*)NONI	<NA>	154	<NA>	-82	25	<NA>	-187	129	<NA>

Columns:

	name	mass	qlt	inr	k=1	cor	ctr	k=2	cor	ctr
1	(*)Msch	<NA>	171	<NA>	-8	126	<NA>	5	45	<NA>
2	(*)Fmmn	<NA>	171	<NA>	9	126	<NA>	-5	45	<NA>
3	NrdO	153	929	174	3	1	0	89	928	462
4	NrdE	79	856	113	-94	830	216	-17	26	8
5	Cntr	125	158	48	10	34	4	-19	124	17
6	Sud	143	849	146	41	214	72	-70	635	265
7	1417	80	801	97	62	428	95	-58	373	103
8	1824	142	914	132	62	552	168	50	363	137
9	2530	187	97	66	-13	62	9	-10	35	7
10	3134	91	841	225	-125	838	436	-8	3	2

Principal inertias (eigenvalues):

	1	2	3	4	5	6	7
Value	0.003263	0.002636	0.000873	0.000599	0.0001	0.000056	0
Percentage	43.35%	35.02%	11.6%	7.96%	1.33%	0.74%	0%

Rows:

	Vivere intense emozioni	Qualcosa che non si puo' rimandare	Occasione per conoscere altre
Mass	0.113344		0.066461
ChiDist	0.049128		0.186597
Inertia	0.000274		0.002314
Dim. 1	-0.612818		2.191866

Dim. 2	0.514730	-2.656738	-0
	Puro divertimento, svago	Entrare in una dimensione di finzione e irrealtà	Stare in compagnia
Mass	0.114374	0.106646	
ChiDist	0.070329	0.070442	
Inertia	0.000566	0.000529	
Dim. 1	-0.032336	0.499124	
Dim. 2	0.961578	0.397583	
	Argomento di cui parlare con gli altri	Modo per rilassarmi	Dimenticarmi del resto del mondo
Mass	0.114374	0.118496	0.0927
ChiDist	0.031138	0.044816	0.0548
Inertia	0.000111	0.000238	0.0002
Dim. 1	0.381874	-0.501158	-0.3296
Dim. 2	0.321236	0.512766	-0.4307

Columns:

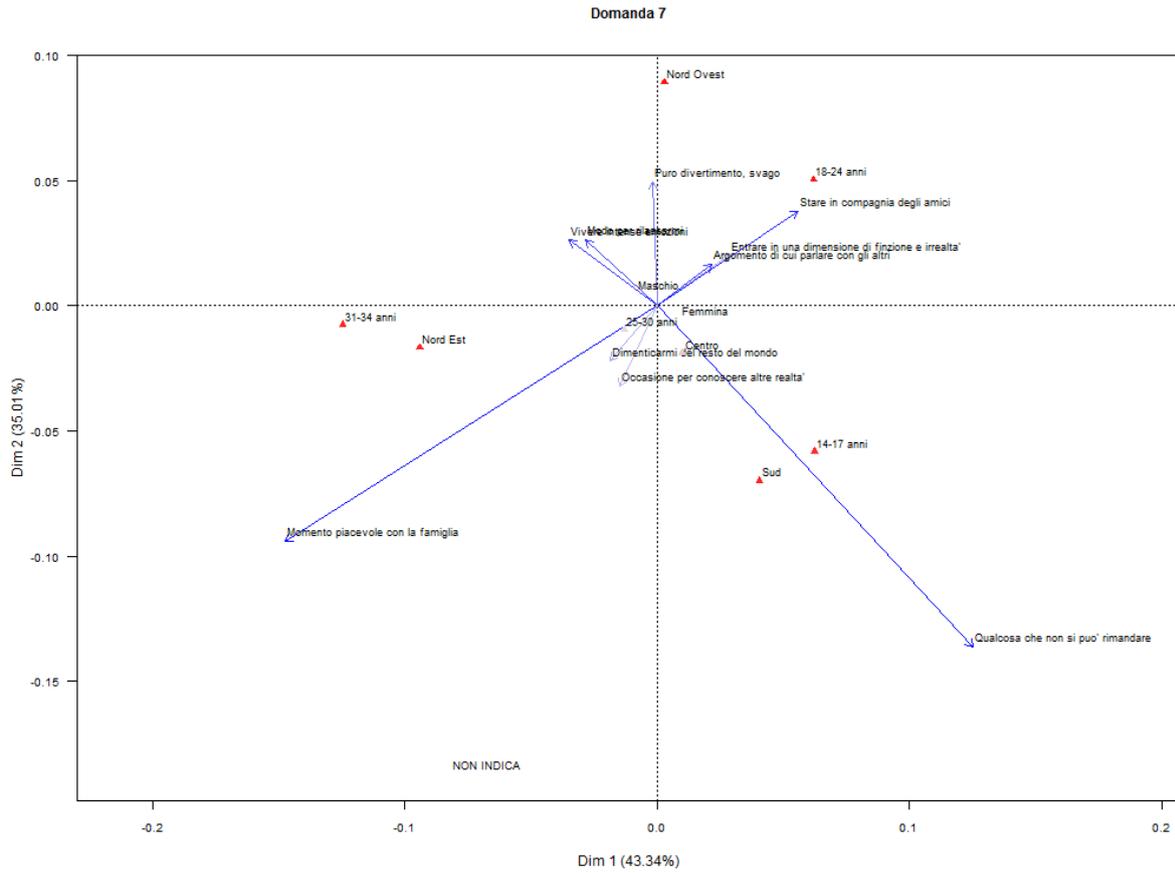
	Maschio (*)	Femmina (*)	Nord Ovest	Nord Est	Centro	Sud	14-17 anni	18-24 anni	25-34 anni
Mass	NA	NA	0.153272	0.079341	0.124678	0.142710	0.079856	0.141937	0.141937
ChiDist	0.023835	0.025698	0.092505	0.103324	0.053828	0.087697	0.095425	0.083617	0.083617
Inertia	NA	NA	0.001312	0.000847	0.000361	0.001098	0.000727	0.000992	0.000992
Dim. 1	-0.148414	0.160014	0.050718	-1.648219	0.174031	0.709825	1.092996	1.087315	1.087315
Dim. 2	0.098356	-0.106043	1.736215	-0.325483	-0.368617	-1.361711	-1.134654	0.981014	0.981014

Row standard coordinates:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	-0.61281808	0.5147298	0.52731706	0.4465344	-1.1085008	0.27668759	1.3719266
[2,]	2.19186596	-2.6567380	-0.19640487	0.8694899	-0.5020040	0.73123109	1.2101852
[3,]	-0.26226927	-0.6255764	2.21414439	-0.4888146	1.1625417	0.12763090	0.6800267
[4,]	-2.58569664	-1.8313531	-0.87740800	0.6957201	0.6136822	-1.43791161	0.9260354
[5,]	-0.03233586	0.9615784	-0.59158376	1.9083807	-0.3424321	0.34509207	0.6569642
[6,]	0.49912374	0.3975825	-1.76830829	-1.1906784	1.2309778	-0.04370269	1.3492569
[7,]	0.97712465	0.7342014	0.64553242	-0.1981052	-0.3282066	-2.14409703	1.0221186
[8,]	0.38187387	0.3212362	-0.08812026	0.3944653	1.0829180	0.26769884	-0.4606688
[9,]	-0.50115793	0.5127662	0.39374112	-0.6258675	0.1672557	1.48247609	1.2994152
[10,]	-0.32968460	-0.4307615	-0.51598381	-1.5988294	-1.9985320	0.11833165	-0.1188676
[11,]	-1.43320726	-3.6347782	7.15484741	-0.5587685	6.9764141	-56.40828072	-32.0103928

Column standard coordinates:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	-0.14841388	0.09835575	-0.5279940	-0.1003608	-0.07888730	-0.84846145	2.133019e+14
[2,]	0.16001368	-0.10604309	0.5692612	0.1082048	0.08505301	0.91477589	-2.299732e+14
[3,]	0.05071838	1.73621460	-0.4127548	-0.8026215	-0.82935253	-0.07044754	1.318517e+00
[4,]	-1.64821926	-0.32548332	0.9980073	0.6969722	0.66374613	-2.42053345	1.318517e+00
[5,]	0.17403132	-0.36861726	0.4860567	1.8741264	-0.52021917	1.35472024	1.318517e+00
[6,]	0.70982480	-1.36171132	-0.5361887	-1.1627885	0.97620402	0.23782671	1.318517e+00
[7,]	1.09299590	-1.13465434	-1.1346811	0.4725188	-2.07432898	-1.49227486	5.113843e-01
[8,]	1.08731458	0.98101389	-0.1538066	0.7663244	1.48136187	-0.30884995	5.113843e-01
[9,]	-0.22485791	-0.18772926	1.4923227	-0.8335205	-0.38095813	0.44918475	5.113843e-01
[10,]	-2.18286501	-0.14842845	-1.8181277	0.1002163	0.29015841	0.86513403	5.113843e-01



Analisi dei valori mancanti

26.1 show_missing

Produce nel log un report dei valori mancanti su tutto il data frame o sulle variabili indicate. I valori validi sono indicati con 1 e i valori mancanti con 0. Le righe rappresentano le diverse combinazioni di valori mancanti, il numero di variabili con valori mancanti e il numero di osservazioni. L'ultima riga conteggia il numero di valori mancanti per variabile.

```
1 show_missing :area, :sesso, :etacon, :titolo, :d1, :d5
```

Output:

```
"Missing values:"
  area sesso titolo etacon d1 d5
231   1     1     1     1   1  1  0
  13   1     1     1     0   1  1  1
  26   1     1     1     1   0  1  1
  28   1     1     1     1   1  0  1
   1   1     1     1     0   1  0  2
   1   1     1     1     1   0  0  2
      0     0     0    14  27 30 71
```

Se viene specificato il parametro `:out` con il nome di un file, produce in aggiunta una serie di grafici. Il formato di file predefinito è **png**. E' possibile salvare i file in formato **pdf** o **wmf** inserendo, oltre al nome, l'estensione. Indicando invece `:screen`, i grafici vengono mostrati a video. E' possibile indicare le variabili che devono essere evidenziate con il parametro `:highlight` o `:hl`.

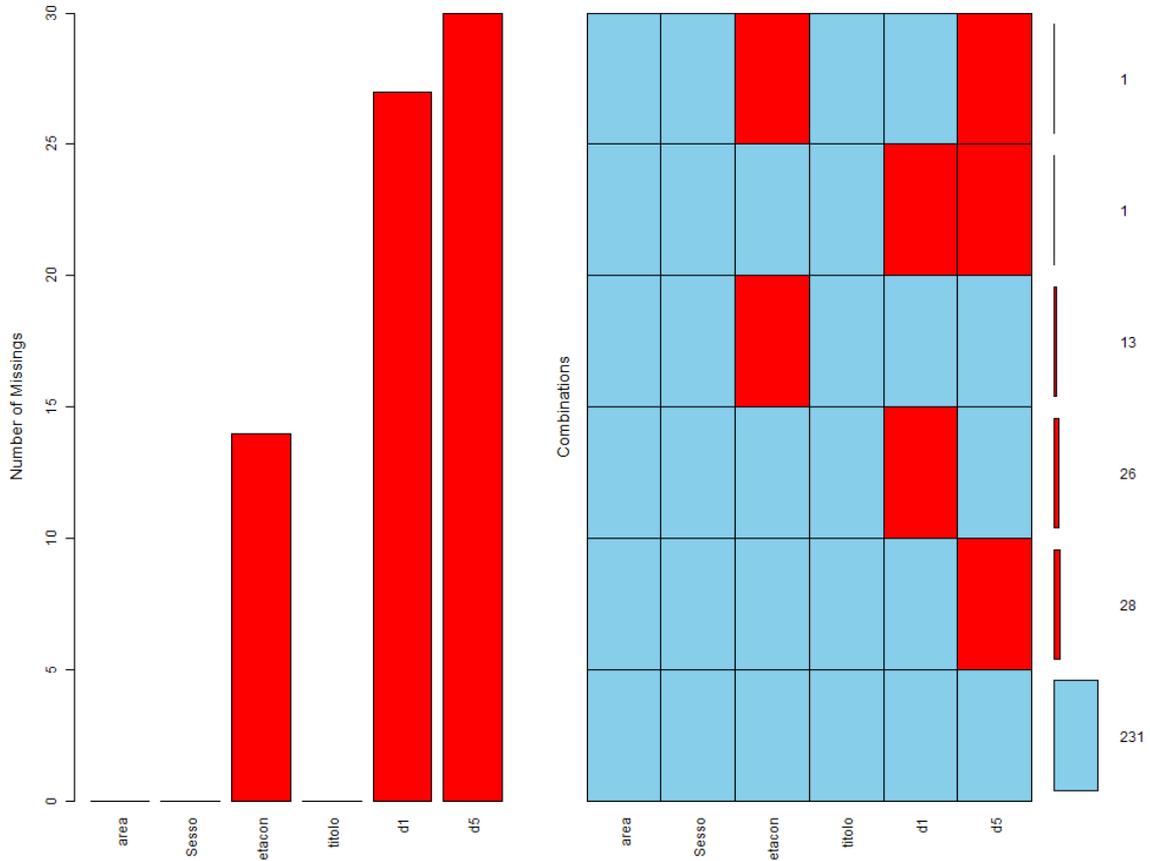
- Numero di missing per variabile e numero di missing per ogni combinazione di variabili.
- Matrice di grafici scatterplot di tutte le coppie di variabili con a margine i valori mancanti. Prodotto solo se le variabili sono al massimo 10.
- Parallel coordinate plot con evidenziati i valori mancanti delle variabili indicate.¹
- Matrice dei dati con i valori validi rappresentati in toni di grigio (dal valore più piccolo al più grande) e i valori mancanti in rosso. I dati sono ordinati sui valori della prima variabile indicata. Prodotto solo se le variabili sono al massimo 50.¹
- Matrice di grafici scatterplot di tutte le coppie di variabili con evidenziati i valori mancanti delle variabili indicate. Prodotto solo se le variabili sono al massimo 10.¹

```
1 show_missing var_list, :file => "filename"
2
3 show_missing
4 show_missing :out => "report_na"
5 show_missing :highlight => [:d1, :d5], :out => "report_na.pdf"
```

¹ Grafico interattivo.

```
6 show_missing :d23_1, :d24_1, s(:d10_,1..5)
7 show_missing :d23_1, :d24_1, s(:d10_,1..5), :hl => :d23_1, :out => :screen
```

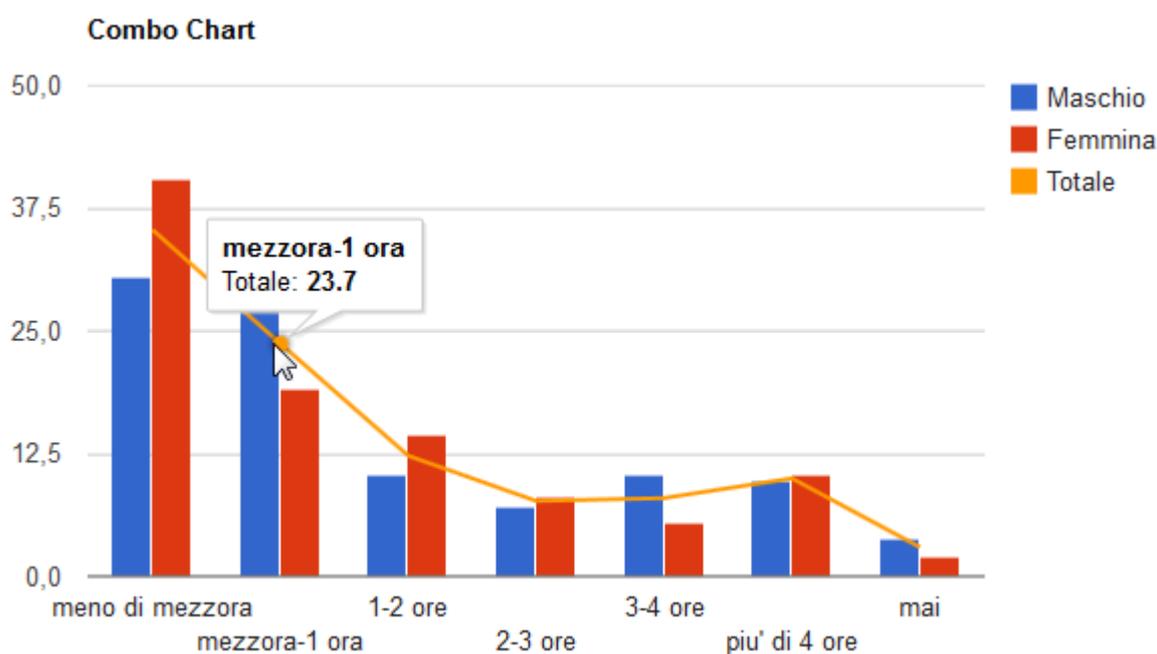
Suggerimento: Utilizzando il parametro `:out => :screen` i grafici sono visualizzati in modalità interattiva. In questa modalità è possibile cambiare l'evidenziazione delle variabili o l'ordinamento dei dati.



Note

Grafici

I **grafici** sono disponibili nell'output **HTML**, **PDF** e **pTabsWeb**.



I grafici utilizzano Google Chart Tools.

Attenzione: Per produrre i grafici in formato **PDF** è necessaria una *connessione internet attiva*. Per visualizzare i grafici nel report **HTML** è necessaria una *connessione internet attiva* e un browser aggiornato. Internet Explorer è supportato dalla *versione 9*.

Sono disponibili due comandi per produrre i grafici: **chart** e **chr**. Il primo calcola i valori dal file di dati, mentre per il secondo l'utente deve fornire i valori da rappresentare.

27.1 Tipi di grafici

Simbo- lo	Grafico	variabili accettate (<i>chart</i>)	Tipo di dati parametro <i>:series (chr)</i>
gauge	termometro	Una + eventuale livello	Un numero (intero o decimale)
pie	grafico a torta	Una	Una serie numerica (un array)
bar	grafico a barre	Una + una di incrocio	Una o più serie numeriche (un array di array)
col	grafico a colonne	Una + una di incrocio	Una o più serie numeriche (un array di array)
line	grafico a linee	Una + una di incrocio	Una o più serie numeriche (un array di array)
curve	grafico a curve	Una + una di incrocio	Una o più serie numeriche (un array di array)
area	grafico a area	Una + una di incrocio	Una o più serie numeriche (un array di array)
step	grafico a gradini	Una + una di incrocio	Una o più serie numeriche (un array di array)
scatter	tabella	Una + una per la seconda serie	Due serie numeriche (un array con due array)
table	tabella	Una + una di incrocio	Una o più serie numeriche (un array di array)
txt	paragrafo di testo	Nessuna: passare <code>nil</code> al posto della variabile	Nessuno

27.2 Opzioni

Parametro	chart	chr	Descrizione
by	x		il nome o simbolo della seconda variabile (la variabile da <i>incrociare</i>)
level	x		il codice della variabile per i grafici di tipo <i>gauge</i> , se non specificato genera un termometro per ciascun codice
total	x		specifica se includere o meno la serie di dati del totale (default = <i>false</i>)
line	x	x	specifica il numero di serie o un array di numeri di serie che devono essere mostrati come linee (grafici <i>combo</i>)
curve	x	x	specifica il numero di serie o un array di numeri di serie che devono essere mostrati come curve (grafici <i>combo</i>)
stat	x	x	la statistica da utilizzare: <i>pct</i> <i>count</i> (default = <i>pct</i> per <i>gauge</i> , <i>count</i> per gli altri)
stacked	x	x	serie impilate: <i>true</i> <i>false</i>
percent	x	x	serie impilate: <i>true</i> <i>false</i>
donut	x	x	grafico a torta con un buco nel centro. Indicare un valore tra 0 e 1 (consigliato un valore tra 0.4 e 0.6). Solo per grafici : <i>pie</i> .
w	x	x	l'ampiezza del grafico
h	x	x	l'altezza del grafico
title	x	x	il titolo del grafico (default = etichetta della variabile)
labels	x	x	il vettore con le etichette dei dati (default = etichette dei livelli della variabile). In alternativa, accetta una funzione <i>lambda</i> che verrà applicata alle etichette estratte dai dati
header	x	x	il vettore con le intestazioni delle serie di dati (default = etichette dei livelli della variabile di incrocio)
animation	x	x	il grafico deve essere animato (<i>true</i>) oppure no (<i>false</i>). Solo per grafici : <i>gauge</i> .
jitter	x	x	aggiunge "rumore" alle serie di dati (<i>true</i> oppure un valore, default <i>false</i>). Solo per grafici "scatter"
note	x	x	la nota sotto il grafico
options	x	x	ulteriori opzioni per la personalizzazione del grafico: Customizing Charts
body	x	x	il corpo del testo. Solo per grafici "txt"
if	x	x	un filtro per selezionare un sottoinsieme di casi (<i>Espressioni</i>)
series		x	il vettore con la serie dei dati

27.2.1 chart

```
chart( :type, :varname, options={} )
```

chart calcola i dati a partire dalle variabili del file di dati e genera il grafico. Ogni istruzione **chart** produce un solo grafico. Più grafici possono essere raggruppati insieme con l'istruzione **charts**.

Il primo argomento del metodo è un simbolo con il tipo di grafico da produrre mentre il secondo argomento è il nome della variabile; un'ulteriore variabile *di incrocio* deve essere specificata con il parametro **:by**.

```
1 chart :gauge, :d5, level: 4, title: "Gauge"
2
3 (2..5).each do |code|
4   chart :gauge, :d5, level: code, labels: lambda{|x| x.upcase}
5 end
6
7 chart :col, :d5, by: :eta
8
```

```
9 chart :col, :d5, by: :eta, total: true, stat: :count, curve: 1
10
11 chart :scatter, :x1, by: :y1, jitter: true
```

Importante: Se è attivo un peso, i dati del grafico saranno **pesati**. Se è attivo un filtro, il grafico verrà prodotto sui dati **filtrati**.

27.2.2 chr

```
chr( :type, options={} )
```

A differenza di *chart*, **chr** produce un grafico partendo da dati imputati direttamente nell'istruzione e non da variabili presenti nel file di dati. I dati vengono passati con il parametro `:series`. Ogni istruzione `chr` produce un solo grafico. Più grafici possono essere raggruppati insieme con l'istruzione **charts**.

Il primo argomento del metodo è un simbolo con il tipo di grafico da produrre.

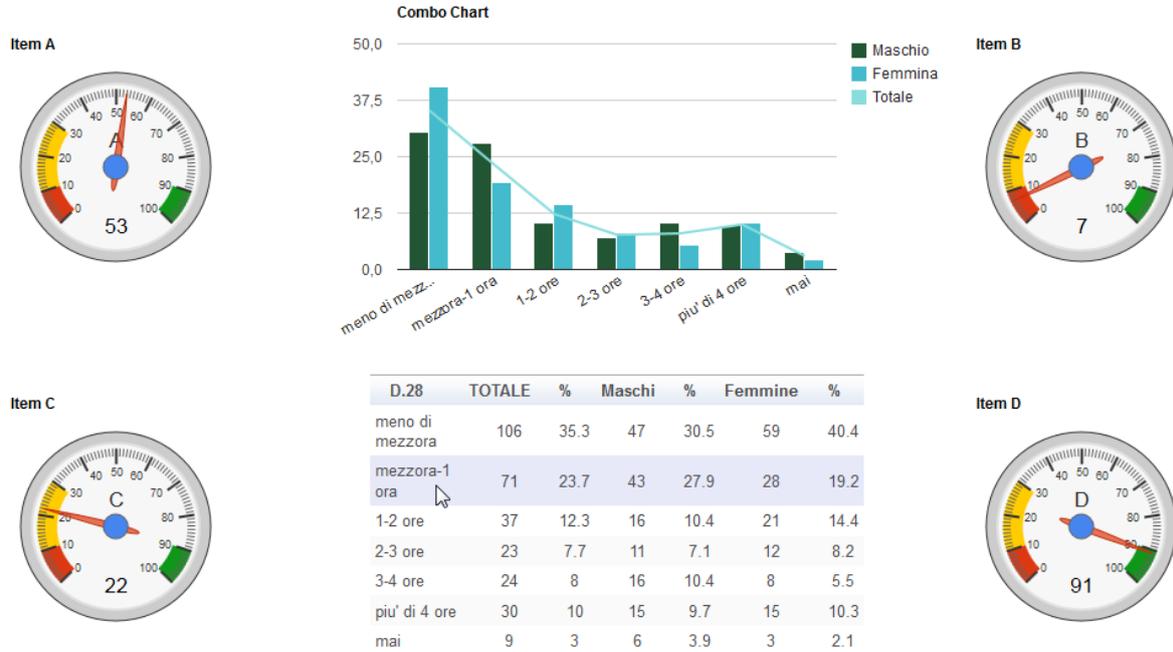
L'istruzione **chr** non effettua calcoli, i dati e le etichette utili per definire il grafico possono essere estratte dalle tabelle prodotte attraverso i metodi indicati nella sezione *Valori restituiti dalle tabelle* oppure tramite le funzioni indicate nella sezione *Funzioni informative*. Assegnare la tabella o il valore restituito da una funzione a una variabile e poi utilizzare i metodi di estrazione sulla variabile.

```
1 # --- interrogando la tabella prodotta
2
3 dom = ord :d28 do
4   w :legend, 15, 45, 90, 150, 210, 300, 0
5   end
6
7 chr :pie,
8   :title => "Pie Chart",
9   :labels => dom.labels,
10  :series => dom.column(:sesso, 1, :count)
11
12 chr :col,
13   :title => "Combo Chart",
14   :labels => dom.labels,
15   :header => dom.headers(:eta)+["Totale"],
16   :series => dom.columns(:eta, :colpct)+dom.columns(:_t, :colpct),
17   :curve => 5
18
19 chr :table,
20   :title => "Table Chart",
21   :labels => dom.labels,
22   :header => ["D.28", "TOTALE", "%", "Maschi", "%", "Femmine", "%"],
23   :series => dom.columns(:_t, [:count, :colpct])+dom.columns(:sesso, [:count, :colpct])
24
25 # --- con xtab
26
27 x = xtab(:area)
28 chr :gauge,
29   :title => x.label,
30   :labels => x.labels.map(&:upcase),
31   :series => x.pcts,
32   :w => 700, :h => "100%"
33
34 x = xtab(:d28 by: :eta)
35 chr :bar,
36   :title => x.label,
37   :labels => x.labels,
38   :header => x.clabels,
39   :series => x.pcts
```

Attenzione: Se è attivo un filtro, l'etichetta del filtro (base) verrà indicata nel grafico, ma è responsabilità dell'utente passare all'istruzione **chr** i dati corretti e coerenti con il filtro.

27.2.3 charts

charts raggruppa insieme diversi grafici prodotti con le istruzioni **chart** o **chr**.



Argomenti:

- il *titolo* del gruppo di tabelle
- il numero di *colonne* su cui disporre i grafici. Per indicare righe con numeri di colonne differenti, specificare un vettore con la sequenza di numeri di colonna per ciascuna riga

Il sottocomando `note` specifica una nota generale per il gruppo di tabelle.

```

1 charts "Titolo grafici", [1,2] do
2   chart ...
3   chart ...
4   chr ...
5   note "Nota generale grafici"
6 end

```

27.2.4 Personalizzazione

Alcuni *parametri di configurazione* permettono di controllare l'aspetto dei grafici:

Parametro	Tipo	Descrizione
chr_width	Fix-num	Larghezza grafico in pixel
chr_height	Fix-num	Altezza grafico in pixel
chr_legend_pos	Symbol	Posizione legenda: :top :bottom :right :none
chr_caption	Boolean	Presenza del titolo all'interno di gruppo di grafici
chr_gauge_options	hash	Opzioni grafico gauge
chr_gauge_width	Fix-num	Larghezza grafico gauge in pixel
chr_gauge_height	Fix-num	Altezza grafico gauge in pixel
chr_gauge_caption	Symbol	Presenza e posizione titolo grafico gauge all'interno di gruppo di grafici: :top :bottom :none
chr_gauge_label	Symbol	Presenza e posizione label grafico gauge: :inner :title :none
palette	Array	Una palette di colori per i grafici

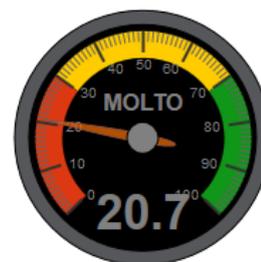
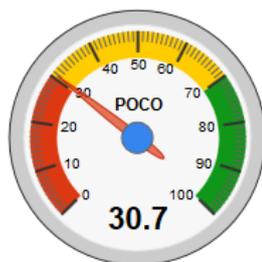
La larghezza e l'altezza possono essere specificati come numeri o come stringhe. I numeri vengono interpretati come pixel, mentre se si specifica una stringa occorre indicare l'unità di misura: "90%".

Inoltre nel foglio stile `ptabs.css`, con il selettore `div#ptabs > div.chart table.chart`, sono definiti diversi stili riferiti ai grafici e in particolar modo al grafico `gauge`:

```

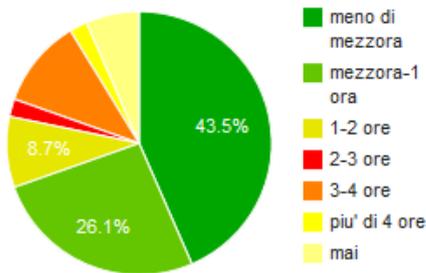
/* --- gauge --- */
/* title */      div#ptabs > div.chart table.chart.gauge td.title { font-size: 12px; text-align: center; }
/* outer-ring */ div#ptabs > div.chart table.chart.gauge svg > g > circle:nth-child(1) { fill: #ccc; }
/* inner-ring */ div#ptabs > div.chart table.chart.gauge svg > g > circle:nth-child(2) { fill: #ccc; }
/* circle pointer */ div#ptabs > div.chart table.chart.gauge svg > g > g > circle { fill: #3684ee; }
/* pointer */   div#ptabs > div.chart table.chart.gauge svg > g > g > path { fill: #dc3912; }
/* testo */     div#ptabs > div.chart table.chart.gauge svg text { fill: #000000; font-size: 12px; }
/* value */    div#ptabs > div.chart table.chart.gauge svg > g > g > text { fill: #000000; font-size: 24px; }
/* label */    div#ptabs > div.chart table.chart.gauge svg > g > text:nth-child(6) { fill: #000000; font-size: 24px; }

```

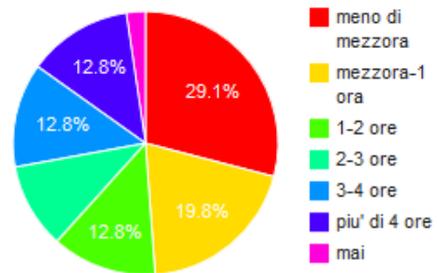


27.3 palette

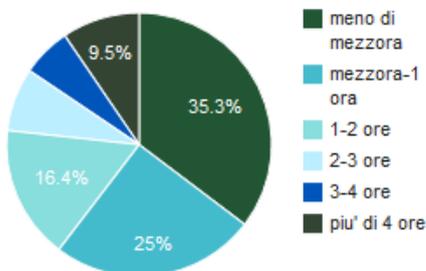
14-17 anni



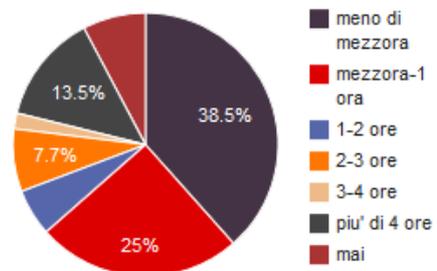
18-24 anni



25-30 anni



31-34 anni



Imposta la palette di colori utilizzata nei grafici qualora non vengano specificati esplicitamente nel parametro options.

```

1 palette ["#225533", "#44bbcc", "#88ddd", "#bbeeff", "#0055bb", "#334433", "#6699aa", "#88aaa",
2 palette %w(#443344 #dd0000 #5566aa #ff7700 #eebb88 #444444 #aa3333 #777799 #dd8833 #ccbb99)
3 palette %w(aqua black blue fuchsia gray green lime maroon navy olive orange purple red silver teal)
4 palette colors(:Greys)

```

27.4 colors

Restituisce i codici dei colori di alcune palette predefinite.

```

1 # restituisce l'intera palette
2 palette colors(:BrBG)
3
4 # restituisce solo 5 colori
5 palette colors(:Dark2, 5)

```



27.5 styles

Sono disponibili due stili per i grafici:

- `:frame`: bordo attorno al gruppo di grafici
- `:box`: bordo attorno a ogni grafico

```
1 styles << :frame
2
3 # elimina stili
4 styles.clear
```

27.6 Opzioni grafico gauge

Esempio di configurazione per il grafico `gauge`:

```
1 :options => {
2   redColor: '#DC3912',
3   yellowColor: '#FFCC00',
4   greenColor: '#109618',
5   redFrom: 0, redTo: 30,
6   yellowFrom: 30, yellowTo: 70,
7   greenFrom: 70, greenTo: 100,
8   minorTicks: 10,
9   majorTicks: ["0", "10", "20", "30", "40", "50", "60", "70", "80", "90", "100"],
10  fontSize: 8,
11 }
```

Sono disponibili quattro configurazioni predefinite per i grafici `gauge`:

- `default_gauge`: impostazioni base; il default se non specificato diversamente

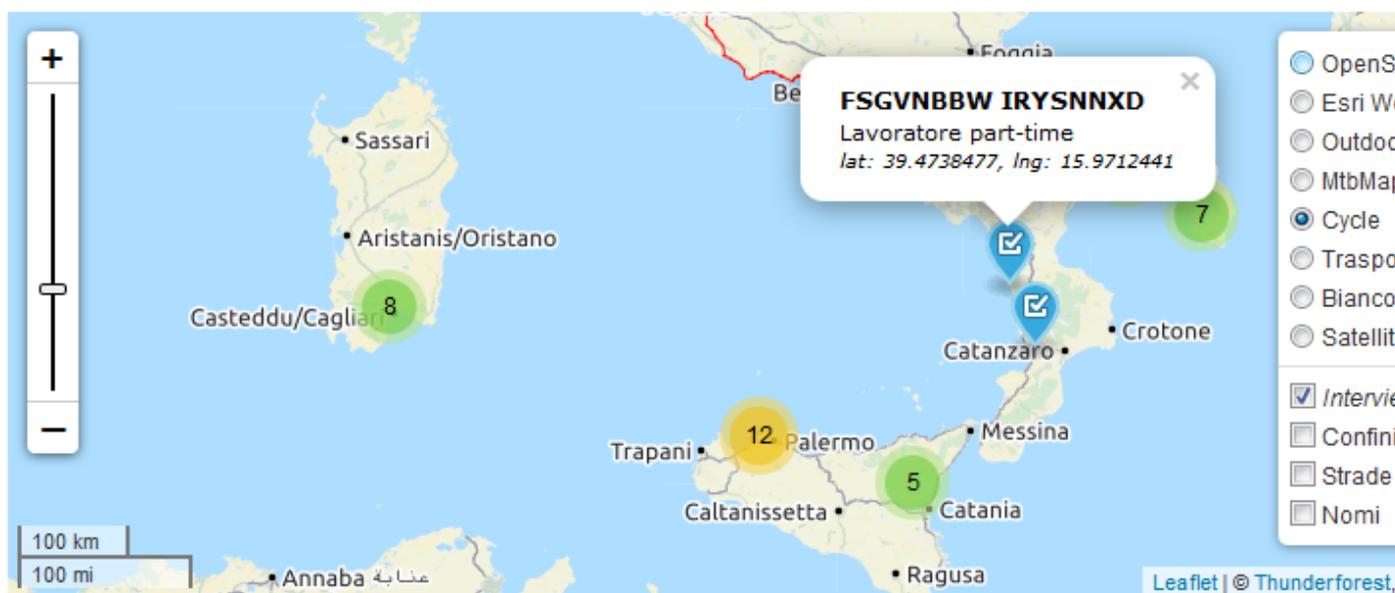
- **traffic_light**: tipo *semaforo* (possibile specificare dei limiti dei 3 colori)
- **used_resource**: warning se valori troppo alti (possibile specificare dei limiti dei 2 colori)
- **free_resource**: warning se valori troppo bassi (possibile specificare dei limiti dei 2 colori)

E' possibile passare le configurazioni predefinite direttamente nel parametro `:options` del grafico, oppure configurare globalmente le impostazioni con `chr_gauge_options`:

```
1  # configurazione generale
2  chr_gauge_options free_resource
3
4  # all'interno del grafico
5  chart, :gauge, ...
6    :options => traffic_light
7
8  # tre intervalli separati
9  chart, :gauge, ...
10   :options => traffic_light(0, 20, 40, 60, 80, 100)
11
12 # fine rosso/inizio giallo e fine giallo/inizio verde
13 :options => traffic_light(25, 65)
```


Mappe

L'istruzione **map** permette di inserire mappe geografiche nell'output **HTML**, **PDF** e **pTabsWeb**.



Le mappe possono essere prodotte a partire da coordinate presenti nel file di dati (per esempio le coordinate che identificano il luogo dell'intervista) e/o da altre coordinate presenti in un file esterno o passate direttamente dallo script (per esempio le coordinate che identificano i punti di campionamento).

I grafici utilizzano [OpenStreetMap](#).

Attenzione: Per produrre le mappe in formato **PDF** è necessaria una *connessione internet attiva*. Per visualizzare le mappe nel report **HTML** è necessaria una *connessione internet attiva*.

Opzioni per mappare i dati del dataframe:

Parametro	Tipo	Descrizione
lat	Varname	il nome della variabile numerica che contiene la latitudine in formato decimale
lng	Varname	il nome della variabile numerica che contiene la longitudine in formato decimale
caption	Varname	il nome della variabile stringa che contiene il testo che deve comparire come titolo nel fumetto
info	Varname	il nome della variabile stringa che contiene il testo che deve comparire nel fumetto
name	String	il nome della serie di dati
marker	Hash	il nome e/o il colore delle icone che rappresentano i punti

Tutti i precedenti parametri possono essere passati come *Array* per rappresentare più serie di dati sulla mappa. `:caption`, `:info` e `:name` se hanno lunghezza 1 vengono riciclati per tutte le serie.

Opzioni per mappare dei punti esterni:

Parametro	Tipo	Descrizione
points	Hash	le serie di punti da rappresentare sul grafico
filenames	String	il nome del file contenente un <i>hash</i> con le serie di punti da rappresentare sul grafico

Opzioni generali:

Parametro	Tipo	Descrizione
w	Fixnum	la larghezza in punti della mappa
h	Fixnum	l'altezza in punti della mappa
note	String	il testo della nota della mappa
js	String	una stringa o un Array di stringhe con istruzioni javascript

```

1 map "Posizione interviste", :lat => :latitudine, :lng => :longitudine, :caption => :name, :info => :info
2
3 map "Mappa con due serie dai dati", :lat => [:lat1, :lat2], :lng => [:long1, :long2],
4     :caption => [:cognome, :cognome], :name => ['Residenza', 'Lavoro']
5
6 map "Mappa con punti aggiuntivi",
7     :lat => :latitudine, :lng => :longitudine, :caption => :name,
8     :marker => {name: 'telephone', color: 'cadetblue'},
9     :points => points
10
11 map "Mappa con solo punti da file esterno", :filenames => "points.rb", :w => 500, :note => "Nota"

```

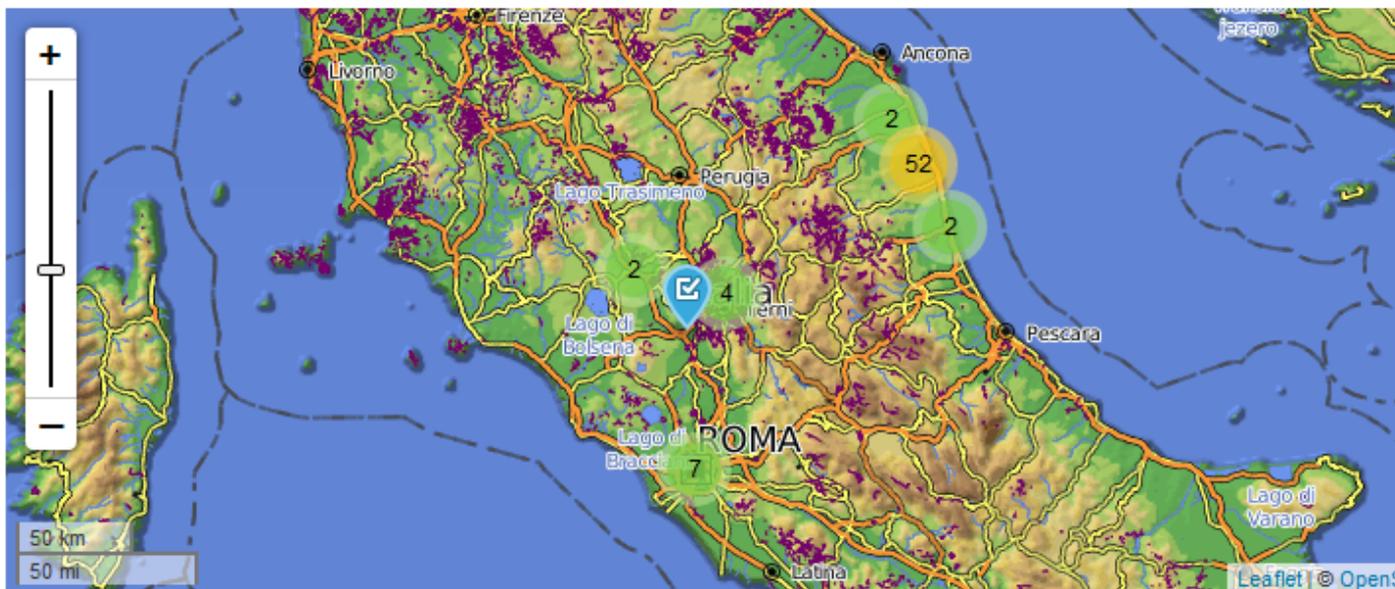
Alcuni *parametri di configurazione generale* impostano varie caratteristiche delle mappe:

Parametro	Tipo	Descrizione
map_width	Fixnum	Larghezza mappa in pixel
map_height	Fixnum	Altezza mappa in pixel
map_search	Boolean	Mostra il riquadro di ricerca
map_search_text	String	Testo riquadro di ricerca
map_zoom_control	Boolean	Mostra i pulsanti di controllo dello zoom
map_zoom_slider	Boolean	Mostra lo slider di controllo dello zoom
map_center	Array	Centro geografico iniziale
map_scale	Boolean	Mostra scala
map_layers_control	Boolean	Mostra controllo selezione layers
map_marker_type	Symbol	mk (Maki), :fa (Font Awesome)
map_marker_size	Symbol	:s :m :l (solo per Maki)
map_marker_icons	Array	Nomi icone per le serie di punti (Maki)
map_marker_icons_fa	Array	Nomi icone per le serie di punti (Font Awesome)
map_layers	Hash	Layers (map tiles principali)
map_overlays	Hash	Overlays (map tiles livelli sovrapponibili)
map_active_layer	Fixnum	Numero layer principale preselezionato
map_data_name	String	Nome serie dati principale
map_boundserie	Fixnum	Serie su cui centrare la mappa
map_clustered	Symbol/Fixnum	Mostra i punti della serie aggregati: # (una specifica serie); :all (tutte le serie); :none (nessuna serie)
map_show_position	Boolean	Stampa le coordinate geografiche nel fumetto del punto
map_show_series	Boolean	Stampa informazioni sulle serie di punti
map_point_label	Array	Etichetta 'punti'

28.1 Layers e overlays

map_layers e map_overlays permettono di definire le *map tiles* (*mappe di tasselli*) utilizzate nella mappa. map_layers definisce le map tiles disponibili come livello principale, mentre map_overlays definisce le map tiles utilizzabili come livelli sovrapposti a quello principale.

Vedere [Leaflet-providers preview](#) per le mappe disponibili.





28.2 Markers

map può utilizzare per le icone dei punti [Maki](#) oppure [Font Awesome](#).

Il parametro di configurazione `map_marker_type` definisce il provider utilizzato di default. Ciascuna serie di dati può comunque specificare un diverso provider attraverso il parametro `:type`.

Nell'istruzione *map*, il parametro `:marker` permette di specificare per ciascuna serie di punti il nome dell'icona, il colore e il provider. Se non specificato, *pTabs* utilizza i valori predefiniti di icone specificate nei parametri `:map_marker_icons_mk` e `:map_marker_icons_fa`.

```
1 :marker => {name: 'place-of-worship'}
2
3 :marker => {name: 'dot-circle-o', color: 'darkblue', type: :fa}
```

Il nome dell'icona deve corrispondere a un nome esistente nella serie di icone del provider scelto.

Utilizzando *Maki* è possibile modificare la dimensione del marker con il parametro `:size: :s, :m o :l`.

```
1 :marker => {name: 'fuel', color: 'green', type: :mk, size: 'm'}
```

I colori disponibili sono:

blue, red, purple, green, orange, darkblue, darkred, darkpurple, darkgreen, cadetblue, lightblue, lightgreen, beige, lightgray, gray, black, white

28.3 Coordinate esterne

In alternativa o in aggiunta alla mappatura di coordinate presente nel file di dati è possibile mappare altre serie di punti.

I dati delle serie possono essere passati al comando *map* con il parametro `:points` oppure memorizzati in un file di dati e il nome del file indicato nel parametro `:filename`.

In entrambi i casi i dati devono essere nel seguente formato:

```

1 {
2   chiave_serie: {
3     name: 'Nome serie',
4     marker: {name: 'nome-icona', color: 'nome-colore'},
5     points: [
6       { lat: 45.471, lng: 9.223, title: "Nome punto", info: "Info punto"},
7       ...
8     ]
9   },
10  altra_serie: { ... },
11  ...
12 }

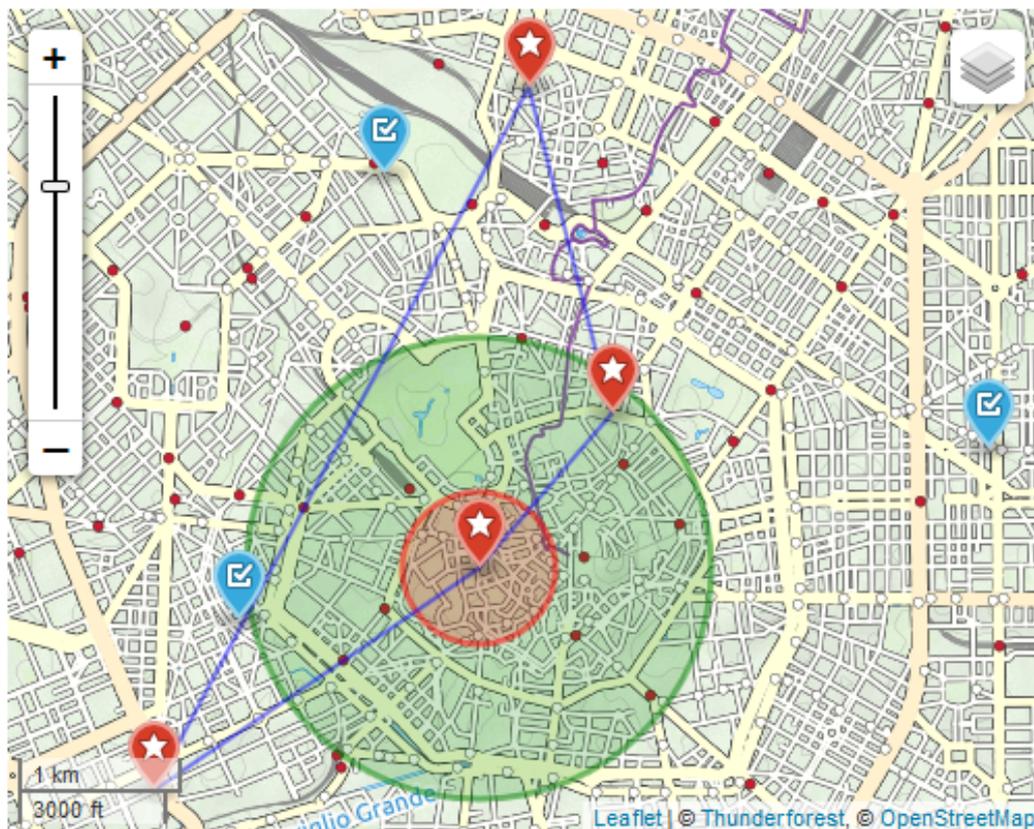
```

Le chiavi `:name`, `:marker` e `:info` sono facoltative.

Suggerimento: Per verificare la correttezza sintattica del file di coordinate, si consiglia di nominare il file con estensione `.rb` e di eseguirlo in *SciTE* con il pulsante *F5*.

28.4 Linee e forme

Attraverso il parametro `:js` è possibile inserire codice *javascript* per aggiungere cerchi, poligoni e altre forme alla mappa.



Nel parametro `js` è possibile utilizzare i seguenti simboli:

- `$divid`: l'ID del DIV HTML che contiene la mappa
- `$mapid`: il nome della variabile javascript che contiene la mappa

```
1 shapes = [  
2   "L.circle([45.464, 9.181], 1500, {color: 'green', weight: 3,}).addTo($mapid);",  
3   "L.circle([45.464, 9.181], 500, {color: 'red', weight: 3,}).addTo($mapid);",  
4   "L.polygon([[45.451, 9.154], [45.464, 9.181], [45.473, 9.192], [45.492, 9.185]], {color: 'blue'  
5 ]  
6 map "Forme", :filenames => "points.rb", :points => points, :js => shapes, :w => 600, :h => 400
```

Per i dettagli della sintassi consultare l'help di Leaflet.

Immagini

29.1 img

img permette di inserire un'immagine nella prima pagina, tra il titolo e l'indice del report.

E' possibile specificare un file locale oppure un indirizzo web (http) di un'immagine. All'immagine è possibile associare un link web che si attiva cliccando sull'immagine.

```
1 img "immagine" [, "link"]
2
3 img "logo.png"
4 img "logo.png", "www.miosito.it"
5 img "http://www.google.it/images/logo.png", "www.google.it"
```

Suggerimento: E' possibile impostare l'immagine della prima pagina anche nel file di configurazione `ptabs.yml` tramite i parametri `img` e `img_link`.

29.2 img_foot

img_foot permette di inserire un'immagine nel piè di pagina dei report PDF.

E' possibile specificare un file locale oppure un indirizzo web (http) di un'immagine. All'immagine è possibile associare un link web che si attiva cliccando sull'immagine.

```
1 img_foot "immagine" [, "link"]
2
3 img_foot "logo_small.png"
4 img_foot "logo_small.png", "www.miosito.it"
5 img_foot "http://www.google.it/images/logo_small.png", "www.google.it"
```

Suggerimento: E' possibile impostare l'immagine del piè di pagina anche nel file di configurazione `ptabs.yml` tramite i parametri `img_foot` e `img_foot_link`.

29.3 image

image permette di inserire un'immagine in qualsiasi punto del report.

E' possibile specificare un file locale oppure un indirizzo web (http) di un'immagine. Alle immagini è possibile associare un link web che si attiva cliccando sull'immagine.

Opzioni disponibili:

- **:link**: link web che si attiva cliccando sulle immagini
- **:title**: titolo dell'immagine
- **:notitle**: non stampa il titolo sopra l'immagine

Formati *HTML* e *PDF*

- **:note**: nota sotto l'immagine
- **:w**: ampiezza dell'immagine in pixel
- **:h**: altezza dell'immagine in pixel
- **:top**: margine superiore dell'immagine in pixel
- **:bottom**: margine inferiore dell'immagine in pixel

Formato *XLSX*

- **:col**: colonna in cui collocare l'immagine: default 1

```
1 image "immagine" [, options]
2
3 image "http://www.google.it/images/logo.png",
4   :link => "http://www.ptabs.eu/",
5   :title => "Titolo immagine",
6   :note => "Nota immagine",
7   :w => 900, :top => 100, :bottom => 100
```

Importante: Sono supportati i formati gif, jpeg, png, jpg. Le immagini gif non sono supportate nel formato PDF.

Importante: Le immagini sono disponibili per i report in formato HTML, PDF e XLSX. Non sono disponibili per il formato XLS.

Importante: Nei report in formato PDF i link non sono attivi.

Testo HTML

30.1 doc

doc permette di inserire una pagina contenente del testo in formato HTML.

L'output di *doc* è disponibile nei formati **HTML**, **PDF** e **pTabsWeb**.

Parametri:

- Il titolo
- `:css`: il codice *CSS* specifico della tavola
- `:markdown`: il markup *Markdown*
- `:haml`: il markup *Haml*
- `:html`: il markup *HTML*

Il titolo compare solo nell'indice non nella pagina.

Il parametro `css` è comunque facoltativo in quanto è già presente una formattazione di base.

Se si vuole inserire degli stili generici per tutte le pagine di testo, utilizzare l'istruzione `css` (*Funzioni di utilità*). In tal caso far precedere i selettori dal selettore `div.doc-wrapper`.

`:markdown`, `:haml` e `:html` sono alternativi.

```
1 doc "Titolo", :html => "<h1>Un titolo</h1><p>Un paragrafo</p><p>Un altro paragrafo</p>"
```

Data la lunghezza del codice da inserire è consigliabile utilizzare delle variabili di testo:

```
1 css_styles = <<-END
2   div#table-doc-wrapper h1 {
3     color: #B22222;
4   }
5   div#table-doc-wrapper ol li {
6     line-height: 32px;
7   }
8 END
9
10 html_text = <<-END
11   <h1>
12     Titolo della pagina
13   </h1>
14   <p>
15     Un paragrafo di testo:<br /><br />
16   </p>
17   <ol>
18     <li>il <b>primo</b> item</li>
19     <li>il <b>secondo</b> item</li>
```

```
20     <li>il <b>terzo</b> item</li>
21   </ol>
22 END
23
24 doc "Titolo", :css => css_styles :html => html_text
```

oppure dei file esterni:

```
1 css = File.read("filename.css")
2 markdown = File.read("filename.markdown")
3 doc "Titolo", :css => css, :markdown => markdown
```

Finestre di dialogo

Le **finestre di dialogo** permettono di mostrare avvisi all'utente e di richiedere l'inserimento di parametri utili per l'esecuzione dello script. Sono disponibili in tutto lo script, sia all'interno del blocco del report, sia all'esterno.

Le *finestre di dialogo* vengono mandate in esecuzione nel punto dello script in cui sono state dichiarate. L'esecuzione dello script viene sospesa per poi riprendere alla chiusura della finestra.

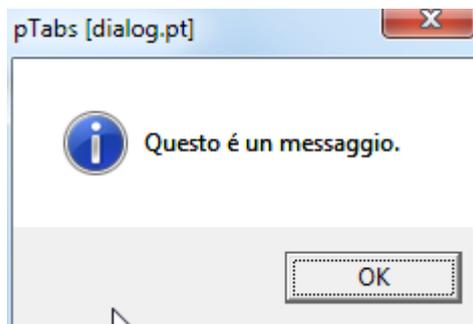
31.1 Messaggi

message, **warning** e **error** generano una finestra con un messaggio. Si differenziano unicamente per l'icona mostrata.

Il parametro `:title` permette di aggiungere un titolo alla finestra.

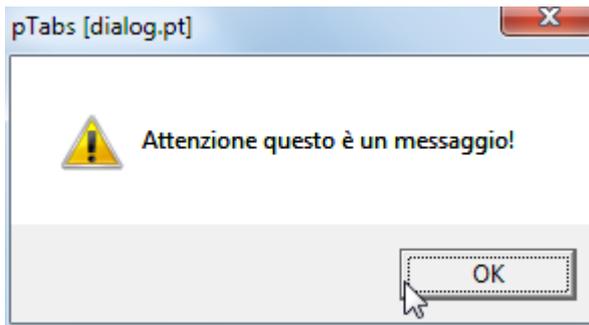
31.1.1 message

```
message("Questo é un messaggio.")
```



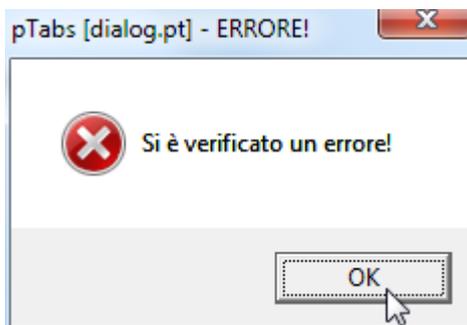
31.1.2 warning

```
warning("Attenzione questo è un messaggio!")
```



31.1.3 error

```
error("Si è verificato un errore!", :title => "ERRORE!")
```



31.2 Input utente

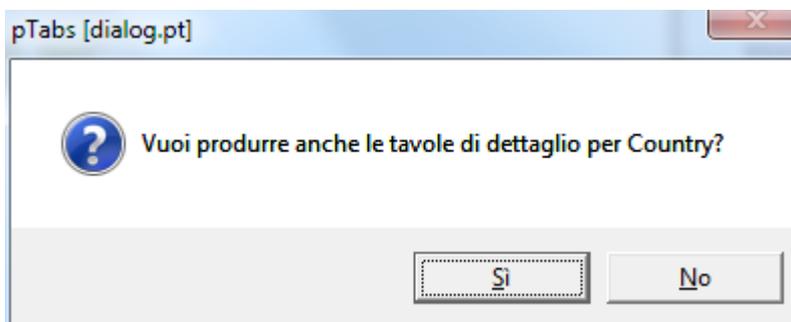
`yesno` e `ask` restituiscono dei valori corrispondenti alle scelte e ai dati inseriti dall'utente.

31.2.1 yesno

`yesno` genera una finestra di dialogo con i pulsanti Sì e No. Restituisce `true` se viene premuto il pulsante **Sì** e `false` nel caso sia stato premuto **No**.

Il parametro `:title` permette di aggiungere un titolo alla finestra.

```
if yesno("Vuoi produrre anche le tavole di dettaglio per Country?")  
  panels.add :_t, :country  
end
```



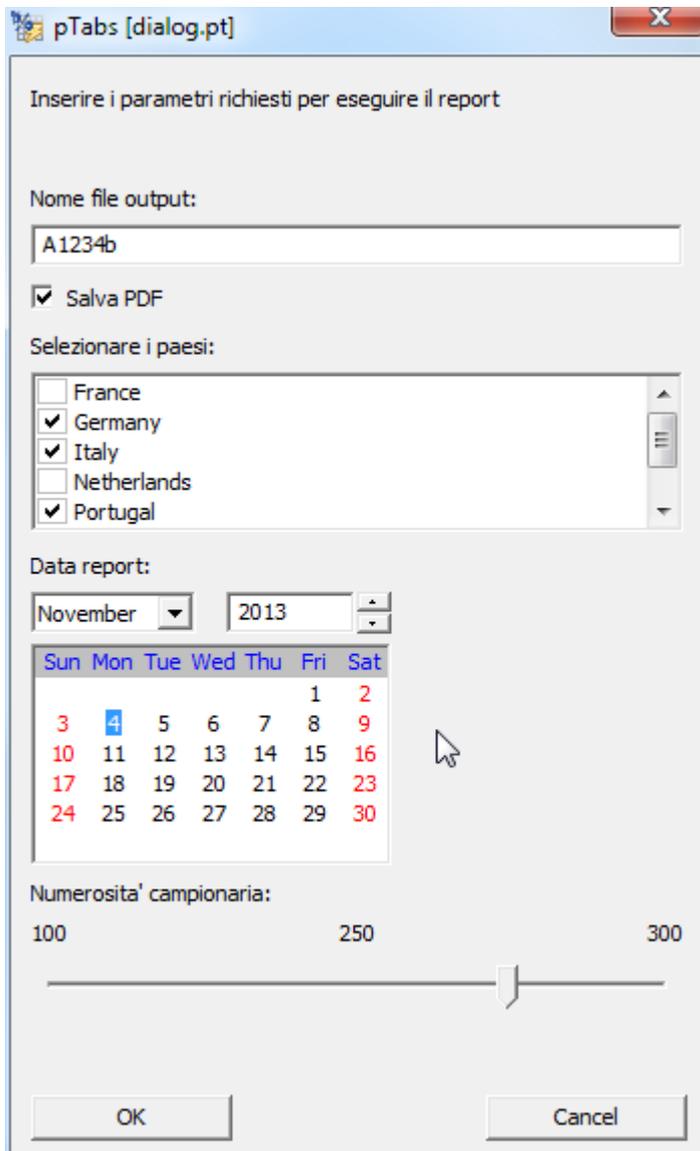
31.2.2 ask

ask genera una finestra di dialogo contenente un numero arbitrario di controlli. Restituisce un vettore contenente i dati inseriti dall'utente. Premendo il pulsante `Cancel` si arresta l'esecuzione dello script.

```

1  n = get_base
2
3  reportname, pdf, countries, day, nn = ask(
4    [
5      ["Nome file output", "A1234"],
6      ["Salva PDF", false],
7      ["Selezionare i paesi", get_codes_and_levels(:country), :type => :multi],
8      ["Data report", :type => :calendar],
9      ["Numerosità campionaria", 100..n, :type => :slider, :initial => (n/2)],
10   ],
11   :message => "Inserire i parametri richiesti per eseguire il report"
12 )
13 debug reportname, pdf, countries, day, nn
14 pt.defvar :pdf => pdf
15 name reportname

```



```

+ [0] Nome file output => A1234b
+ [1] Salva PDF => true

```

```
+ [2] Selezionare i paesi => [2, 3, 5]
+ [3] Data report => 2013-11-04
+ [4] Numerosità campionaria => 250

<String> "A1234b"
<TrueClass> true
<Array> [2, 3, 5]
<Date> #<Date: 2013-11-04 (4913201/2,0,2299161)>
<Fixnum> 250
```

Il primo argomento di *ask* è un vettore contenete i controlli che si desidera inserire nel form.

Il parametro `:title` permette di aggiungere un titolo alla finestra.

Il parametro `:message` permette di inserire un messaggio all'interno della finestra.

```
ask(controls=[], options={})
```

Ciascun controllo è costituito da un vettore contenete un messaggio di prompt, il valore con cui inizializzare il controllo (opzionale), un hash di parametri (opzionale). Il tipo di dato del valore di inizializzazione viene utilizzato per decidere quale controllo generare. Il parametro `:type` permette di modificare il tipo di controllo generato per default. Il parametro `:initial` permette di specificare il valore di default.

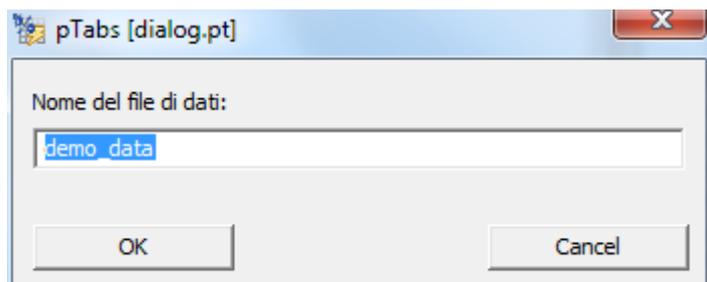
Sono disponibili dieci tipi di controllo:

- **casella di testo:** `[prompt, String]`: un box in cui inserire il testo
- **checkbox:** `[prompt, true/false]`: un checkbox
- **spin:** `[prompt, Range]`: un controllo con due pulsanti per incrementare/decrementare un valore numerico
- **slider:** `[prompt, Range, :type => :slider]`: un cursore per impostare un valore numerico
- **lista singola:** `[prompt, Array]`: una lista combinata per selezionare una voce da una lista
- **lista multipla:** `[prompt, Array, :type => :multi]`: una lista di checkbox per selezionare uno o più voci da una lista
- **lista singola con codici:** `[prompt, Hash]`: una lista combinata per selezionare una voce da una lista; restituisce il codice corrispondente
- **lista multipla con codici:** `[prompt, Hash, :type => :multi]`: una lista di checkbox per selezionare uno o più voci da una lista; restituisce i codici corrispondenti
- **calendario:** `[prompt, Date, :type => :calendar]`: un calendario per selezionare una data
- **data:** `[prompt, Date, :type => :date]`: una casella con una data e i pulsanti per incrementarla/decrementarla

Per generare una finestra di dialogo con semplicemente una casella di testo è possibile utilizzare la sintassi:

```
ask(prompt, default)

spss.open ask("Nome del file di dati", "demo_data")
```



Funzioni definite dall'utente

In pTabs è possibile definire delle proprie funzioni ¹ che si comportano in modo analogo ai comandi predefiniti. Una nuova funzione può essere utilizzata per eseguire una serie di istruzioni di pTabs o per memorizzare valori e dati da utilizzare nello script.

Una funzione corrisponde a quello che in altri software viene definito come *macro*.

Una funzione può ricevere dei parametri che possono condizionare la sua esecuzione o possono essere elaborati dalla funzione.

Una funzione che restituisce uno o più valori può essere utilizzata come argomento di un'altra istruzione pTabs.

Avvertimento: Attenzione a non utilizzare come nomi delle proprie funzioni e variabili nomi di istruzioni di pTabs. In tal caso la nuova funzione o variabile sovrascriverà quella di pTabs, che non sarà più visibile.

32.1 Funzioni e variabili interne al report

Per rendere più fluida la scrittura di uno script ed evitare ripetizioni, può essere utile definire delle variabili o delle proprie funzioni all'interno di un report.

Le **funzioni** vengono definite con l'istruzione `def`.

Una **variabile** viene creata semplicemente assegnando un valore a un identificatore.

```

1  report = ptabs.new do
2    ...
3
4    pesi = [0.25, 0.37, 0.75, 1.5, 2.5, 3, 3.5, 4, 5]
5
6    def my_tab(n, root)
7      ord "#{root}_#{n}"
8      w :legend, pesi
9    end
10   end
11
12   (1..10).rep {|i| my_tab i, :q45}
13   (1..15).rep {|i| my_tab i, :q47}
14   (1..18).rep {|i| my_tab i, :q52}
15
16   ...
17 end

```

La variabile `pesi` e la funzione `my_tab` saranno visibili solo all'interno del blocco `ptabs.new ... end`.

Nota: Questo tipo di sintassi è utilizzabile solo se la definizione e l'utilizzo della funzione o della variabile

¹ Tecnicamente in Ruby, come in genere nei linguaggi a oggetti, si tratta di metodi. Qui useremo il termine di funzioni in senso lato.

avviene all'interno dello stesso report.

32.2 Funzioni e variabili globali

Un modo più proficuo di scrivere le proprie funzioni, comporta la possibilità di riutilizzarle in differenti report dello stesso lavoro o tra diversi lavori.

Per definire tali tipi di funzioni e variabili è necessario utilizzare una sintassi specifica.²

32.2.1 Definizione

Le nuove funzioni possono essere definite con l'istruzione **defpfun** per le funzioni da usare al di fuori del report e **deffun** per le funzioni da usare all'interno del blocco del report (`ptabs.new ... end`). Il nome della funzione deve essere un simbolo.

```
1  defpfun :my_function_1 do |argomenti|
2    # corpo della funzione
3  end
4
5  deffun :my_function_2 do |argomenti|
6    # corpo della funzione
7  end
```

Le variabili possono essere definite con l'istruzione **defpvar** per le variabili da usare al di fuori del report e **defvar** per le funzioni da usare all'interno del blocco del report (`ptabs.new ... end`). Il nome della variabile deve essere un simbolo.

```
1  defpvar :my_var_1 => valore
2
3  defvar :my_var_2 => valore
4  defvar :my_var_3 => valore, :my_var_4 => valore, :my_var_5 => valore
5  defvar my_var_6: valore, my_var_7: valore
```

nome: valore è una variante sintattica equivalente a `:nome => valore`.

Un'unica istruzione *defpvar* o *defvar* permette di creare più variabili contemporaneamente.

32.2.2 Collocazione

Queste funzioni e variabili possono essere definite ovunque: nello stesso script o in file esterni.

I file esterni devono essere file *Ruby* con estensione `.rb` e possono essere caricati nello script con la sintassi:

```
1  require 'my_filename' # senza estensione
```

Se il file risiede nella stessa cartella dello script, verrà trovato senza la necessità di specificare il percorso.

user_functions.rb

La cartella `user`³ contiene lo stub file `user_function.rb` in cui è possibile aggiungere delle proprie funzioni di uso ricorrente. Questo file viene automaticamente caricato in ogni script di *pTabs*.

² Tecnicamente *defpfun*, *defpvar*, *deffun* e *defvar* generano dei nuovi metodi all'interno dei moduli `User::Ptabs` e `User::Report`. *defpvar* è un alias di *defpfun* e *defvar* è un alias di *deffun*

³ La cartella `user` si trova nel percorso di installazione di *pTabs*. Una nuova installazione non ne sovrascrive il contenuto. Nel caso sia stata definita la variabile d'ambiente `PTABS_USER` la cartella `user` non è più attiva e bisogna utilizzare al suo posto quella definita in `PTABS_USER`. In tal caso è necessario copiare il file `user_functions.rb`.

E' anche possibile inserire nella cartella `user` ulteriori file *Ruby*. Questi file dovranno essere caricati esplicitamente nello script con l'istruzione `require`, ma non è necessario specificarne il percorso.

Per fare in modo che vengano caricati automaticamente, inserire l'istruzione `require` direttamente nel file `user_functions.rb`.

Nel caso si voglia caricare un file presente in un percorso differente, magari di rete, aggiungere il percorso nell'istruzione `require`, ricordandosi di utilizzare il carattere `/` e non `\`:

```
1 require 'x:/my/path/my_filename'
```

32.2.3 defpfun e defpvar

defpfun e **defpvar** devono essere usati per generare funzioni e variabili che lavorano esternamente alla definizione del report e che hanno bisogno di far riferimento all'oggetto **ptabs**. Le nuove funzioni e variabili saranno disponibili con la sintassi: `ptabs.my_function_name`.

Esempio: esecuzione parametrizzata di un report

Un report viene definito e poi eseguito due volte con parametri differenti.

```
1 defpfun :myrep do |rep_name, caption, data_file|
2
3   report = ptabs.new(filename) do
4     title caption
5     spss.open data_file
6     ...
7   end
8   report.render :html
9
10 end
11
12 ptabs.myrep "rep001", "Report XXX", "dat001"
13 ptabs.myrep "rep002", "Report YYY", "dat002"
```

Esempio: generazione output

La funzione che esegue i metodi `render` desiderati del report, permettendo di scegliere se produrre tutti gli output (`:test`) o solo l'`html` (`:all`).

```
1 defpfun :print do |report, mode=:test|
2   report.render :html
3   if mode == :all
4     report.render :pdf
5     report.render :xls, :single_panel => true,
6       :sheets => { "VA %" => [:count, :rowpct, :colpct, :mresp, :base, :mean, :sd, :],
7         "%" => [ :rowpct, :colpct, :mresp, :base, :mean
8         "VA" => [:count, :mresp, :base, :mean
9     report.render :db
10  end
11 end
12
13 report = ptabs.new do
14   ...
15 end
16 ptabs.print report
```

32.2.4 deffun e defvar

deffun e **defvar** devono essere usati per generare funzioni e variabili che devono essere eseguite all'interno di un report (all'interno del blocco `ptabs.new ... end`) per esempio funzioni che definiscono tavole. Le nuove funzioni e variabili saranno direttamente disponibili all'interno di un report con la sintassi: `my_function_name`.

Esempio: tabella personalizzata

Si definisce una nuova istruzione che genera un tipo specifico di tavola da usarsi per variabili con scala a 10 punti più non risponde.

```
1  deffun :ord_net do |varname, caption|
2
3    ord varname, caption do
4      net "NEGATIVO" => 1..5, "POSITIVO" => 6..10, "NON RISPONDE" => 11
5      w :legend, "( # ) *", 1,2,3,4,5,6,7,8,9,10, :na
6      hide 14
7    end
8
9  end
10
11 report = ptabs.new do
12   ...
13   ord_net :v25, "Tabella D25 - ..."
14   ...
15 end
```

Esempio: pesi di uso ricorrente

Una variabile può servire semplicemente per memorizzare e poi restituire dei valori (dati o strutture di dati) da utilizzare nello script.

Nell'esempio, vengono definiti dei pesi da utilizzare per una specifica tipologia di domanda.

```
1  defvar :my_weights => [0.25, 0.37, 0.75, 1.5, 2.5, 3, 3.5, 4, 5]
2
3  # -- nel report
4  ord ... do
5    w :legend, my_weights
6  end
```

Il valore memorizzato nella variabile non è un semplice numero, ma è un vettore contenente più elementi. La variabile viene usata, nell'esempio, come argomento del sottocomando `w`.

Esempio: template del titolo della tabella

Si definisce un template personalizzato da usare nei titoli delle tabelle.

```
1  deffun :my_title(n, title, subtitle=nil)
2    "Dom.#{n} - #{title}" + (subtitle.empty? ? "" : "\n -- #{subtitle} --")
3  end
4
5  # -- nel report
6  cat :d1, my_title(1, "Titolo", "Sottotitolo")
```

Questa funzione non esegue nessuna istruzione pTabs, ma restituisce⁴ semplicemente una stringa di testo costruita a partire dai parametri di input.

⁴ Qualsiasi funzione in Ruby restituisce un valore: la variabile o il valore presente nell'ultima riga del corpo della funzione o comunque l'ultima espressione valutata.

La funzione viene usata, nell'esempio, come argomento dell'istruzione `cat` al posto del titolo.

Esempio: libreria di ponderazioni

L'esempio seguente illustra un semplice metodo per creare una libreria contenente i marginali e le numerosità degli universi da utilizzare nelle ponderazioni.

```

1  # -- file: users/universi.rb
2  # INDIVIDUI >= 18 ANNI
3  defvar :uni_18_n => 47_431_000,
4      :uni_18 => {
5      :sesso => [47.8, 52.2],
6      :ampc  => [18.6, 29.2, 17.4, 1.0, 23.8],
7      :istr  => [28.0, 36.3, 28.4, 7.3],
8      :area  => [10.7, 16.1, 11.8, 7.3, 13.4, 9.0, 18.9, 12.7],
9      }
10
11 # FAMIGLIE
12 defvar :uni_fam_n => 21_810_000,
13      :uni_fam => {
14      :area  => [11.8, 16.8, 11.8, 7.6, 13.0, 9.1, 17.5, 12.4],
15      :ncomp => [24.9, 27.1, 21.6, 18.9, 7.6],
16      :etacap => [11.9, 18.9, 18.5, 17.9, 32.8],
17      }
18
19 # -- nello stesso script o meglio in 'user_functions.rb'
20 require 'universi'
21
22 # -- nello script
23 rake :names => [:ipf, :esp],
24      :counts => [uni_18_n],
25      :population => uni_18

```

47_431_000 viene letto come un semplice numero intero. Nei numeri, il carattere sottolineato `_` può essere utilizzato come separatore delle migliaia per facilitare la lettura.

32.3 Callbacks

I *callbacks* sono funzioni che vengono richiamate durante la generazione dei report. Per comportamento predefinito non svolgono nessuna azione, ma l'utente può ridefinirle per inserire dei comportamenti specifici.

32.3.1 Funzioni dell'oggetto report

Queste funzioni vanno ridefinite con `deffun`.

render_title

`render_title` permette di modificare il titolo della tabella.

```

1  # Segnatura
2  render_title(title, index)
3
4  # Esempio di ridefinizione:
5  # - aggiunge "TAB.# - " e mette in maiuscolo il titolo della tavola
6  deffun :render_title do |title, index|
7      "TAB.#{index} - "+title.upcase
8  end
9

```

```
10 # Esempio di ridefinizione:
11 # - elimina "V###_#: Question ###_#: " dal titolo della tavola
12 deffun :render_title do |title, index|
13   title.gsub(/V[\d_]+:\sQuestion\s\d+:\s?/, '').strip
14 end
```

skip_variable

`skip_variable` permette di valutare se escludere o meno una variabile/set di variabili dal *topline* o dalla procedura *autotab*. Viene eseguito prima del calcolo delle statistiche.

`variable` è un oggetto `PTables::VariableInfo` (vedere *info*).

```
1 # Segnatura
2 skip_variable(variable)
3
4 # Esempio di ridefinizione:
5 # - la variabile compare nel topline solo se il metodo restituisce false
6 deffun :skip_variable do |variable|
7   return true if variable.alpha?
8   return true if %w(intnr inttime scrnt internr stime).include?(variable.name)
9   return true if variable.label =~ /^INTERVISTATORE:/i
10  return true if variable.levels.include?("CATI")
11  false
12 end
```

skip_summary

`skip_summary` permette di valutare se escludere o meno una variabile/set di variabili dall'output del topline. Viene eseguito dopo aver calcolato le statistiche.

`summary` è un oggetto `PTables::TopLine::Summary`.

Metodi dell'oggetto `Summary`:

- **varname**: restituisce il nome della variabile o la radice del nome del set di variabili
- **varnames**: restituisce un Array con i nomi delle variabili
- **caption**: restituisce il titolo
- **labels**: restituisce un Array con le etichette
- **id**: restituisce il progressivo della tabella
- **numeric?**: restituisce `true` se la variabile è numerica
- **single?**: restituisce `true` se la variabile è singola
- **multi?**: restituisce `true` se la variabile è multipla
- **base**: restituisce la base

```
1 # Segnatura
2 skip_summary(summary)
3
4 # Esempio di ridefinizione:
5 # - la variabile compare nel topline solo se il metodo restituisce false
6 deffun :skip_summary do |summary|
7   return true if summary.base < 10
8   false
9 end
```

auto_tab

`auto_tab` viene chiamato dalla procedura `autotab` per produrre automaticamente tutte le tabelle per un elenco di variabili o, se non specificato, per tutte le variabili del file di dati.

Di `auto_tab` viene già fornita un'implementazione completa, ma è possibile personalizzare la procedura ridefinendo la funzione.

```

1  # Segnatura
2  auto_tab(variables_list)
3
4  # Implementazione di default
5  deffun :auto_tab do |variables|
6    i = 0
7    variables.each do |var|
8      inf = info var
9      next if skip_variable(inf)
10     next if inf.alpha?
11     if inf.set?
12       if inf.multi?
13         md var.vars, "Tabella #{i+=1} - [$vn()] #{var.title}"
14       else
15         scale var.vars, "Tabella #{i+=1} - [$vn()] #{var.title}"
16         var.vars.each do |v|
17           ord v, "Tabella #{i+=1} - [$vn()] $v1()"
18         end
19       end
20     elsif inf.factor?
21       cat var, "Tabella #{i+=1} - [$vn()] $v1()"
22     elsif inf.numeric?
23       scale var, "Tabella #{i+=1} - [$vn()] $v1()"
24     end
25   end
26 end

```

Simboli per test significatività

Tre funzioni definiscono l'elenco di simboli utilizzati per identificare le colonne nei test di significatività.

Le funzioni devono restituire un *Hash* con le chiavi `:header` e `markers`:

- `:header`: simboli da utilizzare per le intestazioni delle colonne
- `:markers`: simboli da utilizzare nelle celle

markers_1_level

Simboli per pairwise comparisons con 1 solo livello di significatività.

```

1  # (implementazione di default)
2  deffun :markers_1_level do
3    markers = ('a'..'z').to_a + %w(* $ & @ # ^ = + - !) + ('A'..'Z').to_a
4    {
5      :header => markers,
6      :markers => markers
7    }
8  end

```

markers_2_level

Simboli per pairwise comparisons con 2 livelli di significatività. `markers` contiene un solo vettore di simboli. Per il secondo livello di significatività i simboli vengono convertiti in **maiuscolo**.

```
1 # (implementazione di default)
2 deffun :markers_2_level do
3   markers = (['']+(1..9).to_a).map{|i| ('a'..'z').to_a.map{|x| "#{x}#{i}"}}.flatten
4   {
5     :header => markers,
6     :markers => markers
7   }
8 end
```

markers_9_level

Simboli per pairwise comparisons fino a 9 livelli di significatività. `markers` è un vettore che contiene nove vettori di simboli, uno per ogni livello di significatività.

```
1 # (implementazione di default)
2 deffun :markers_9_level do
3   markers = ('a'..'z').to_a
4   {
5     :header => markers,
6     :markers => ((1..9).to_a).map{|i| markers.map{|x| "#{x}#{i}"}}
7   }
8 end
```

32.3.2 Funzioni dell'oggetto ptabs

Queste funzioni vanno ridefinite con `defpfun`.

before_report

`before_report` viene eseguito prima della definizione del report.

```
1 # Segnatura
2 before_report (ptabs)
3
4 # Esempio di ridefinizione:
5 # - scarica il file di dati da un sito ftp
6 defpfun :before_report do |ptabs|
7   require 'net/ftp'
8   ftp = Net::FTP.new 'ftp.name.com'
9   ftp.login "user", "password"
10  ftp.chdir "/path/ftp"
11  ftp.getbinaryfile "source.dat", "target.dat"
12  ftp.close
13 end
```

after_report

`after_report` viene eseguito al termine della definizione del report.

```
1 # Segnatura
2 after_report (report)
```

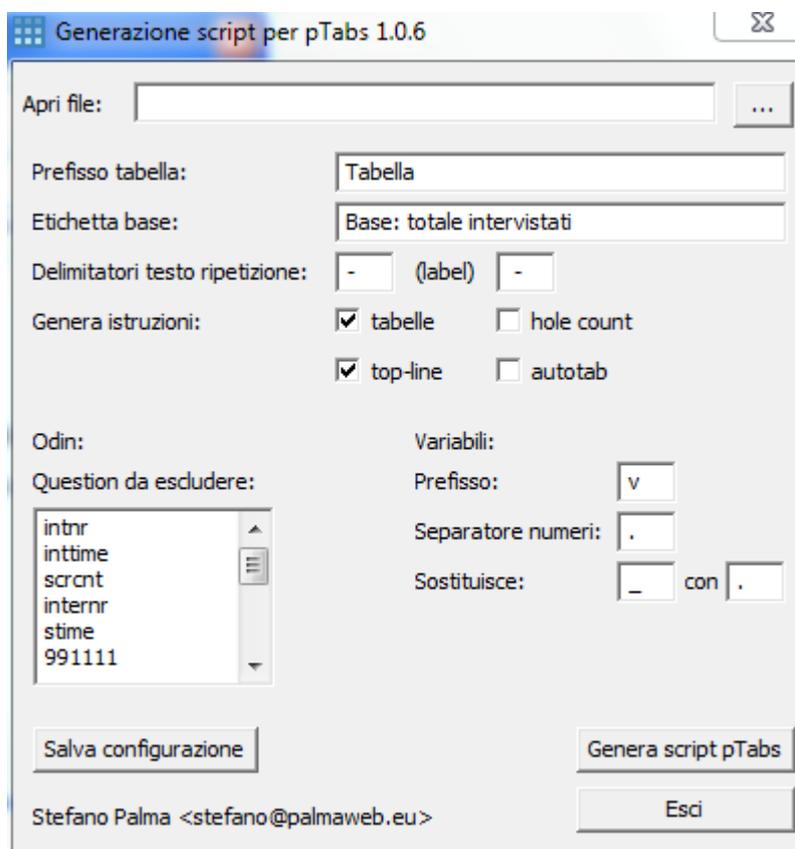
before_exit

`before_exit` viene eseguito prima del termine di pTabs.

```
1 # Segnatura
2 before_exit(ptabs)
3
4 # Esempio di ridefinizione:
5 # - copia il report prodotto
6 defpfun :before_exit do |ptabs|
7   system "copy report.pdf x:\\Public"
8 end
```

Note

ptGen.exe è un programma che permette di generare uno script per **pTabs** partendo da un questionario *Odin* o da un file di dati *SPSS*.



33.1 Questionario Odin

ptGen legge il questionario *Odin* e genera le istruzioni di tabelle per tutte le domande che contengono un testo. I nomi delle variabili vengono costruiti in base al valore del parametro `variable_prefix` più il numero della question. Nelle ripetizioni si aggiunge il progressivo della ripetizione preceduto dal carattere definito nel parametro `number_separator`. E' supportato qualsiasi livello di ripetizione.

33.2 File di dati SPSS

Partendo da un *system file SPSS*, **ptGen** genera le istruzioni di tabelle per tutte le variabili che iniziano con `variable_prefix`. Se in `variable_prefix` è specificata una stringa vuota, genera le tabelle per tutte le variabili. Gruppi di variabili multiple sono riconosciuti se la parte numerica progressiva è separata dal resto del nome della variabile da `number_separator`.

33.3 File Triple-S

ptGen genera le istruzioni di tabelle per tutte le domande definite in un file *xml Triple-S*.

33.4 Configurazione

Un file di configurazione `ptgen.yml`¹ contiene i parametri di configurazione del programma.

E' possibile utilizzare un proprio file `ptgen.yml` anche con parametri parziali. Più file di configurazione vengono letti in sequenza dalle seguenti posizioni:

- dal file eseguibile (non modificabile)
- dalla cartella che contiene l'eseguibile
- dalla cartella dell'utente²
- dalla cartella di lavoro

Cliccando sul pulsante "Salva configurazione" viene creato/modificato il file `ptgen.yml` nella cartella dell'utente.

Parametri del file `ptgen.yml`:

Parametro	Descrizione
<code>variable_prefix</code>	prefisso utilizzato per generare i nomi di variabile
<code>number_separator</code>	carattere utilizzato per searare i numeri nei nomi di variabile
<code>report_prefix</code>	prefisso per il titolo del report
<code>title_prefix</code>	prefisso per il titolo della tabella
<code>placeholder</code>	simbolo per sostituire * ?
<code>txt_rep_label_ini</code>	testo iniziale eichetta ripetizione
<code>txt_rep_label_end</code>	testo finale eichetta ripetizione
<code>output_format</code>	formato script generato
<code>exclude_questions</code>	elenco dei numeri di question odin da escludere
<code>exclude_font_codes</code>	vettore con le coppie di font di apertura/chiusura che delimitano il testo da eliminare
<code>weights</code>	nomi variabili pesi
<code>row_id</code>	nome variabile id di riga
<code>panels</code>	pannelli
<code>base</code>	etichetta totale campione
<code>xls_options</code>	opzioni per output Excel

Esempio di file `ptgen.yml`:

```
---
:xls_options:
  :single_panel: true
  :multiple_sheets: false
  :count: false
  :base: false
  :sd: false
```

¹ Il file è in formato *YAML*.

² La cartella definita dalla variabile d'ambiente `HOME` oppure dalle variabili `HOMEDRIVE` e `HOMEPATH`.

```

:se: false
:report_prefix: ""
:exclude_questions:
- 991111
- 991112
- 900210
- 900211
- 900600
- 900701
- 1000096
:exclude_font_codes:
- - 2
  - 1
- - 2
  - 0
:title_prefix: "Tabella "
:placeholder: <?>
:base: "Base: totale intervistati"
:row_id: :ser_no
:txt_rep_label_ini: "- "
:txt_rep_label_end: " -"
:weights:
- :ipf
- :espcam
- :espuni
:variable_prefix: v
:panels:
- - :_tot
  - :sesso
  - :etacl
  - :istr
- - :_tot
  - :mcr
  - :ampc
:output_format: :pt
:number_separator: _

```

33.5 Total Commander, Unreal Commander

Si può inserire un collegamento a *ptGen* nella barra degli strumenti di *Total Commander* o di *Unreal Commander*. Dopo aver inserito il pulsante è necessario eliminare qualsiasi percorso dal *percorso di avvio* del pulsante. In questo modo *ptGen* si avvierà nella cartella del pannello attivo.

Note

pTabsDB

pTabsDB (version 2)

Ricerca 0000 - Demo programma pTabs

Dati: demo Periodo: 1: Gennaio 2012

Basi: []

Tabella: 5 - Quanto è d'accordo con la frase ...?

Arrotonda valori:

Etichette colonne: []

Fiancata

#	variabile	cod.	net	etichetta
1	d5	0		Totale
2	d5	1	*	NON RISPONDE
3	d5	2		Non risponde
4	d5	3	*	NEGATIVO
5	d5	4		Per niente d'accordo
6	d5	5		Poco d'accordo
7	d5	6	*	POSITIVO
8	d5	7		Abbastanza d'accordo
9	d5	8		Molto d'accordo

Ordinamento fiancata

Nessuno

Inverte righe

Ordinamento ascendente (su I serie)

Ordinamento discendente (su I serie)

Inverte colonne pannello

Traspone righe e colonne

Pannello

#	variabile	cod.	etichetta
1	. tot	1	TOTALE
2	area	1	Nord Ovest
3	area	2	Nord Est
4	area	3	Centro
5	area	4	Sud
6	sezzo	1	Maschio
7	sezzo	2	Femmina
8	sexeta	1	Maschio: 14-17 anni
9	sexeta	2	Maschio: 18-24 anni
10	sexeta	3	Maschio: 25-30 anni
11	sexeta	4	Maschio: 31-34 anni
12	sexeta	5	Femmina: 14-17 anni
13	sexeta	6	Femmina: 18-24 anni
14	sexeta	7	Femmina: 25-30 anni
15	sexeta	8	Femmina: 31-34 anni
16	titolo	1	Nessun titolo/Elementare
17	titolo	2	Media inferiore
18	titolo	3	Media superiore
19	titolo	4	Laurea
20	prof	1	Lavoratore a tempo pieno
21	prof	2	Lavoratore part-time
22	prof	3	Studente

Statistica: % col

Grafico

Titolo

Etichette valori

Legenda

Modello: []

Tipo: colonne raggruppate

Tabella

Titolo

Intestazione statistica

Riga basi

Tipo oggetto: Foglio di lavoro Excel

Aggiornamento

Aggiorna etichette

Aggiunge serie

pTabsDB è un database Access dotato di un'interfaccia utente che permette di navigare i risultati della tabulazione e di produrre grafici e tabelle in *PowerPoint*. E' anche possibile aggiornare i grafici e le tabelle precedentemente create con dei nuovi dati.

Avvertimento: pTabsDB richiede Microsoft Access 2010

34.1 Selezione dei dati

Dati **Periodo**
Basi
Tabella

Il database può contenere più insiemi di dati selezionabili attraverso gli elenchi a discesa **Dati** e, se presente; **Periodo**. La casella **Basi** permette di selezionare le tabelle in base all'etichetta della base (corner della tabella). Infine in **Tabella** si trova l'elenco delle tabelle che soddisfano i precedenti criteri di selezione. Oltre al titolo della tabella, viene mostrato il suo identificativo e la tipologia di tabella.



elimina la selezione delle basi.



permette di visualizzare i dati della tabella selezionata.

Fiancata					Pannello			
#	variabile	cod.	net	etichetta	#	variabile	cod.	etichetta
1	d5	0		Totale	1	._tot	1	TOTALE
2	d5	1	*	NON RISPONDE	2	area	1	Nord Ovest
3	d5	3	*	NEGATIVO	3	area	2	Nord Est
4	d5	4		Per niente d'accodo	4	area	3	Centro
5	d5	5		Poco d'accordo	5	area	4	Sud
6	d5	6	*	POSITIVO	6	secco	1	Maschio

Statistica

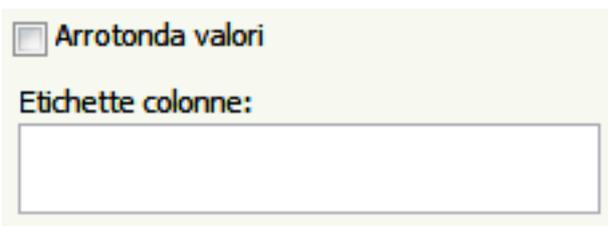
Selezionando una tabella, nella casella di riepilogo **Fiancata** vengono mostrate le righe della fiancata disponibile per quella tabella in relazione alla *Statistica* scelta, mentre in **Pannello** si trovano tutti breaks disponibili per quella tabella. Nell'elenco a discesa **Statistica** sono presenti le statistiche disponibili per quella tabella. Cambiando statistica selezionata viene riaggiornata la lista delle righe disponibili in *Fiancata*. E' possibile selezionare le righe di fiancata e le colonne che si desidera compaiano nel grafico o nella tabella che si vuole produrre.

Ordinamento fiancata

Nessuno
 Inverte righe
 Ordinamento ascendente (su I serie)
 Ordinamento discendente (su I serie)

Inverte colonne pannello
 Traspone righe e colonne

E' possibile ordinare diversamente gli item della fiancata o invertendone l'ordine, oppure ordinandoli in senso ascendente o discendente rispetto ai dati contenuti nella prima colonna selezionata del pannello. L'ordine delle colonne può essere invertito. Infine è possibile trasporre l'intera tabella invertendo le righe con le colonne e viceversa.



I valori decimali possono essere rappresentati come interi. E' possibile inserire delle nuove etichette per le colonne. Le etichette delle varie colonne devono essere separate con il carattere di punto e virgola (;). Quest'opzione è utile principalmente per indicare il periodo di riferimento quando si producono grafici o tabelle di serie storiche.

34.2 PowerPoint



apre una presentazione PowerPoint vuota.

Per aggiungere grafici e tabelle in una presentazione di propria scelta è sufficiente aprirla in PowerPoint. In caso contrario, ne verrà aperta una vuota alla creazione del primo grafico o tabella.



inserisce una nuova slide vuota in una presentazione.

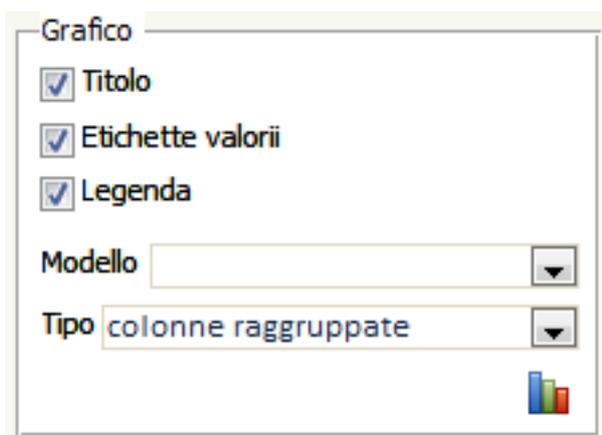


elimina la slide attiva.



cancella il contenuto della slide attiva.

34.3 Grafici



E' possibile scegliere il tipo di grafico da produrre selezionandone uno dall'elenco a discesa **Tipo**, oppure scegliere un template di grafico Excel da **Modello**. Saranno elencati tutti i modelli disponibili nel PC in uso.
1

Gli elementi *titolo*, *etichette dei valori* e *legenda* sono opzionali.

¹ In Windows 7 la cartella dei modelli di grafico è solitamente <cartella utente>\AppData\Roaming\Microsoft\Templates\Charts\.



genera il grafico.

34.4 Tabelle



E' possibile produrre due tipi di oggetti diversi scegliendoli dall'elenco a discesa **Tipo oggetto: Foglio di lavoro Excel** o **Tabella PowerPoint**.

E' possibile riportare nella slide anche il *titolo* della tabella, l'*intestazione della statistica* scelta e infine aggiungere una riga contenete la *base* delle diverse colonne.

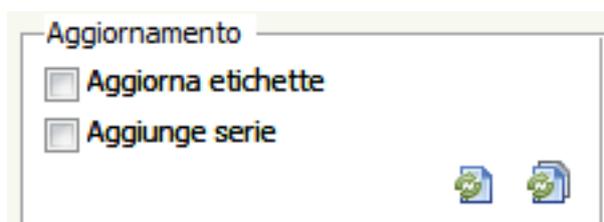


genera la tabella.



In alternativa alla produzione della tabella, questo pulsante permette di copiare la tabella come testo negli appunti. In questo modo è possibile incollare la tabella, per esempio, in un foglio Excel o in un documento Word.

34.5 Aggiornamento



E' possibile aggiornare i *grafici e le tabelle Excel* ² prodotte precedentemente da pTabsDB e salvati in una presentazione.

Nel foglio dati, oltre alla tabella vera e propria, vengono aggiunte in righe e colonne nascoste ulteriori informazioni che servono per ripetere l'interrogazione su un nuovo database e recuperare i dati aggiornati.

² Le tabelle in formato PowerPoint non possono essere aggiornate, in quanto non è possibile aggiungere ulteriori informazioni nascoste oltre a quelle visualizzate.

Attenzione: Per questo scopo è necessario che il nuovo database abbia la stessa struttura di quello precedente, ovvero le tabelle devono avere lo stesso identificativo, così come le variabili dell'indagine. Inoltre le righe della fiancata devono essere le stesse.^a

E' importante non corrompere la struttura dei grafici e delle tabelle, eliminando totalmente o parzialmente i metadati salvati.

Le tabelle in formato PowerPoint non possono essere aggiornate

^a Eventuali nuovi item aggiunti alla variabile e presenti nel nuovo database vengono, in realtà, ignorati, mentre eventuali item mancanti provocano un errore e il termine della procedura di aggiornamento.

L'aggiornamento utilizza le informazioni già salvate durante la generazione dei grafici e delle tabelle, pertanto eventuali selezioni di tabelle, item, statistiche, ordinamenti, ecc. nell'interfaccia sono irrilevanti.

E' possibile aggiornare anche le **etichette** oppure lasciarle invariate (utile specialmente quando sono state modificate successivamente la generazione del grafico/tabella).



aggiorna unicamente gli elementi presenti nella slide attiva.



aggiorna tutta la presentazione.

34.5.1 Serie storiche

Spuntando la casella **Aggiunge serie** è possibile, piuttosto che aggiornare i dati preesistenti, aggiungere una nuova serie (colonna) di dati.

Questa opzione avrà effetto solo su tabelle e grafici generati inizialmente su di una sola serie di dati. Non è inoltre possibile aggiungere una nuova serie su matrici trasposte.

La/le serie preesistente/i diventeranno "fisse", non più aggiornabili.

Quando si aggiunge una nuova serie di dati, può essere utile inserire un'etichetta per i nuovi dati nella casella **Etichette colonne**.

34.6 Altre funzioni



apre questa pagina.



apre una finestra di informazioni.



nasconde la finestra di Access se presente.



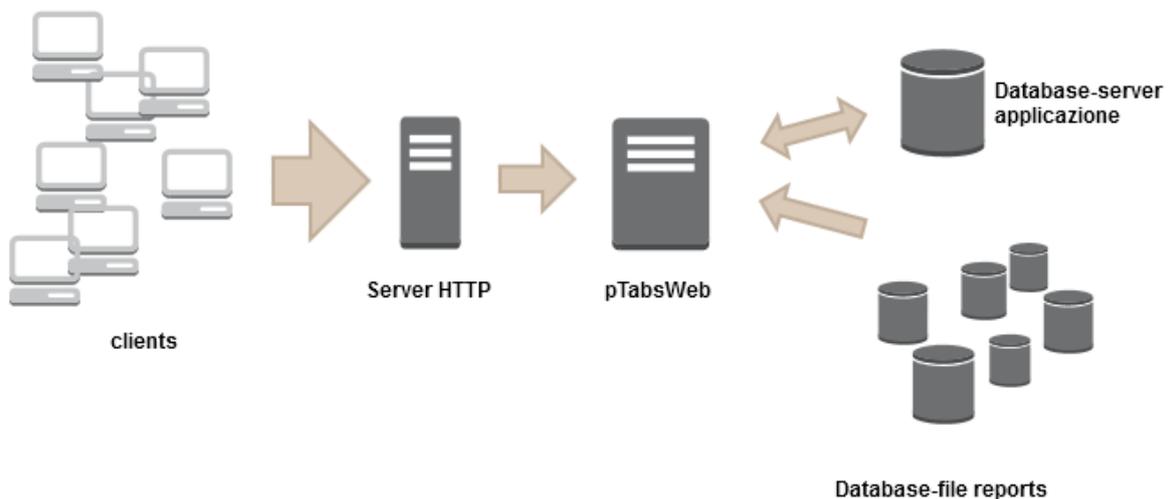
esce dal programma.

Note

pTabsWeb

IN SVILUPPO (Non ancora disponibile)

pTabsWeb è un'applicazione server che permette l'accesso via Web alle tabelle di un report prodotto con pTabs.¹



Prova una [demo](#) di pTabsWeb (Login: demo - Password: zd3ap8c9) o vai alla documentazione: [pTabsWeb Help](#).

Un nuovo formato di *output* di pTabs permette di creare i file necessari per alimentare l'applicazione. Non è necessario nessun passaggio di conversione o importazione dei dati, semplicemente si aggiunge una nuova istruzione di render del report nello script, come già viene fatto per i normali output in PDF, Excel, ecc. Il database generato contiene i dati relativi alle tabelle calcolate nel report e non i dati disaggregati originali. pTabsWeb non calcola nuove tavole, ma presenta le tavole definite nel report.²

```
1 # genera il file <nome_report.db>
2 report.render :ptweb, :file => "nome_report"
```

Un database centrale contiene i dati relativi al funzionamento dell'applicazione come credenziali di accesso, dati di sessione, informazioni relative ai progetti e ai report dei vari clienti.

¹ pTabsWeb è realizzato con il linguaggio di programmazione Ruby e il framework web Ruby on Rails.

² Le tavole di pTabsWeb sono simili ma non identiche a quelle degli altri output di pTabs. Alcuni elementi o tipologie di tavole al momento non sono supportate.

35.1 Screenshot

pTabsWeb Company Demo Utente Demo (DemoGroup) [logout](#)

Progetti Reports **Tavole** Utente

Test pTabsWeb 2 - Risultati febbraio 2015

Indagine Demo pTabsWeb

- 1 CAMPIONE
- 2 CAMPIONE 2
- 3 Tabella 1 - Di solito con quale frequenza va al cinema
Totale intervistati
- 4 **Tabella 2 - Rispetto a un anno fa diresti che è cambiato qualcosa nella frequenza con cui va al cinema?**
Totale intervistati
- 5 Tabella 5 - Quanto è d'accordo con la frase ...?
Totale intervistati
- 6 Tabella 6 - Quanto sono importanti per Lei questi aspetti?
Totale intervistati

Base:

Titolo:

il report contiene 30 tabelle

pTabsWeb Company Demo Utente Demo (DemoGroup) [logout](#)

Progetti Reports **Tavole** Utente

Test pTabsWeb 2 - Test

Tabella 5 - Quanto è d'accordo con la frase ...?

Totale intervistati	TOTALE	Area geografica				Sesso		
		Nord Ovest	Nord Est	Centro	Sud	Maschio	Femmina	
Totale	320	58	88	88	88	162	159	
NON RISPONDE	25	4	5	5	10	8	17	
NEGATIVO	124	31	45	30	18	69	55	
Per niente d'accordo	22	8	11	2	-	10	11	
Poco d'accordo	102	22	34	28	18	58	44	
POSITIVO	172	23	37	53	59	85	87	
Abbastanza d'accordo	114	19	35	39	21	54	59	
Molto d'accordo	58	5	2	13	39	31	27	
	base	296	54	82	82	77	154	142
	media	2,71	2,36	2,94	2,77	3,26	2,69	2,72
	dev. standard	0,87	0,86	0,73	0,75	0,82	0,87	0,86

Pesi: non risponde = n.a., per niente d'accordo = 1, poco d'accordo = 2, abbastanza d'accordo = 3, molto d'accordo = 4

Company Demo S.r.l. Stefano Palma © 2015 pTabs

Variabili di pannello:

Statistiche:

Net:

Esporta:

PDF:

pTabsWeb 2 - Test

Tabella 5 - Quanto è d'accordo con la frase ...?

Totale intervistati	TOTALE	Area geografica				Condizione lavorativa				
		Nord Ovest	Nord Est	Centro	Sud	Lavoratore a tempo pieno	Lavoratore part-time	Studente	In cerca di occupazione	Casalanga
Totale	100,0%	18,1%	27,3%	27,3%	27,3%	49,4%	13,8%	30,3%	5,6%	1,0%
NON RISPONDE	7,7%	6,8%	6,3%	5,8%	11,6%	6,6%	8,0%	7,0%	14,9%	38,9%
NEGATIVO	38,6%	53,0%	51,4%	34,1%	20,9%	39,6%	36,8%	34,2%	54,6%	61,1%
POSITIVO	53,7%	40,3%	42,3%	60,0%	67,5%	53,8%	55,2%	68,8%	30,5%	-

Pesi: non risponde = n.a., per niente d'accordo = 1, poco d'accordo = 2, abbastanza d'accordo = 3, molto d'accordo = 4

Company Demo S.r.l. Stefano Palma © 2015 pTabs

Variabili di pannello:

Statistiche:

Net:

Esporta:

PDF:

Note

Utilità da linea di comando

36.1 dft2pt

dft2pt converte i file **dft** in file **pt**.

```
1 > dft2pt nome_file.dft
```

Verrà creato il file `nome_file.dft.pt`.

36.2 cleanQ

cleanQ normalizza i file di testo dei questionari. Salvare il file Word in formato testo e processarlo con *cleanQ*.

```
1 > cleanq nome_file_questionario.txt
```

Verrà creato il file `nome_file_questionario.txt.clr`.

I file di risorse di default possono essere estratti da pTabs e copiati nella cartella corrente tramite lo switch `rs`:

```
ptabs -rs
```

37.1 File di configurazione di pTabs

```
ptabs.yml
```

37.2 Fogli stile

```
ptabs.css
```

```
ptabs_pdf.css
```

37.3 Header e footer PDF

```
pdf_header.html
```

```
pdf_footer.html
```

37.4 Installazione librerie R

```
install_packages.R
```

Risoluzione dei problemi

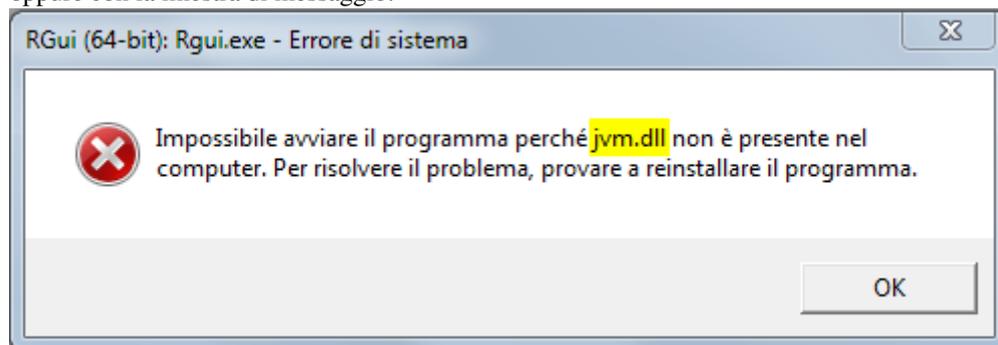
38.1 pTabs

38.1.1 Virtual Machine Java

Errore: pTabs termina con un errore relativo alla Virtual Machine Java:

- can't create Java VM
- Error occurred during initialization of VM
- Could not reserve enough space for object heap

oppure con la finestra di messaggio:



Suggerimento: Configurare i percorsi della *Virtual Machine Java* e del *Java Development Kit*

Se l'errore si verifica con qualsiasi script che cerca di produrre un report in formato Excel, è probabile che si tratti di un problema di configurazione.

Controllare il contenuto del proprio file di configurazione `ptabs.yml` verificando le voci:

```
:jsdk_home_32:
:jre_home_32:
:jre_home_64:
```

Queste voci indicano dove pTabs cerca la *Virtual Machine Java* e il *Java Development Kit*.

Eseguendo pTabs verificare se i percorsi indicati nella finestra di output sono corretti, per esempio:

```
--- Java SE: c:/stat/java/jre7
--- Java SDK: c:/stat/java
```

Verificare l'installazione consultando *Installazione*.

Errore: Parse error on eval:options(width=150, scipen=5)

Suggerimento: Configurare i percorsi della *Virtual Machine Java* e del *Java Development Kit*

Si tratta probabilmente anche in questo caso di un problema di configurazione della *Virtual Machine Java* e del *Java Development Kit*.

38.1.2 error: cannot delete

Errore: All'inizio dell'esecuzione, pTabs stampa questi errori:

```
error: cannot delete old C:/stat/ptabs/lib/pt/postinst.rb
error: cannot delete old C:/stat/ptabs/lib/pt/win32/open3.so
error: cannot delete old C:/stat/ptabs/lib/R/pt/libraries.R
```

Suggerimento: Modificare i permessi delle cartelle in cui è installato pTabs

pTabs non ha i permessi per gestire i file e le cartelle del programma. E' necessario aumentare i permessi di accesso alle cartelle e ai file presenti nel percorso di installazione di pTabs.

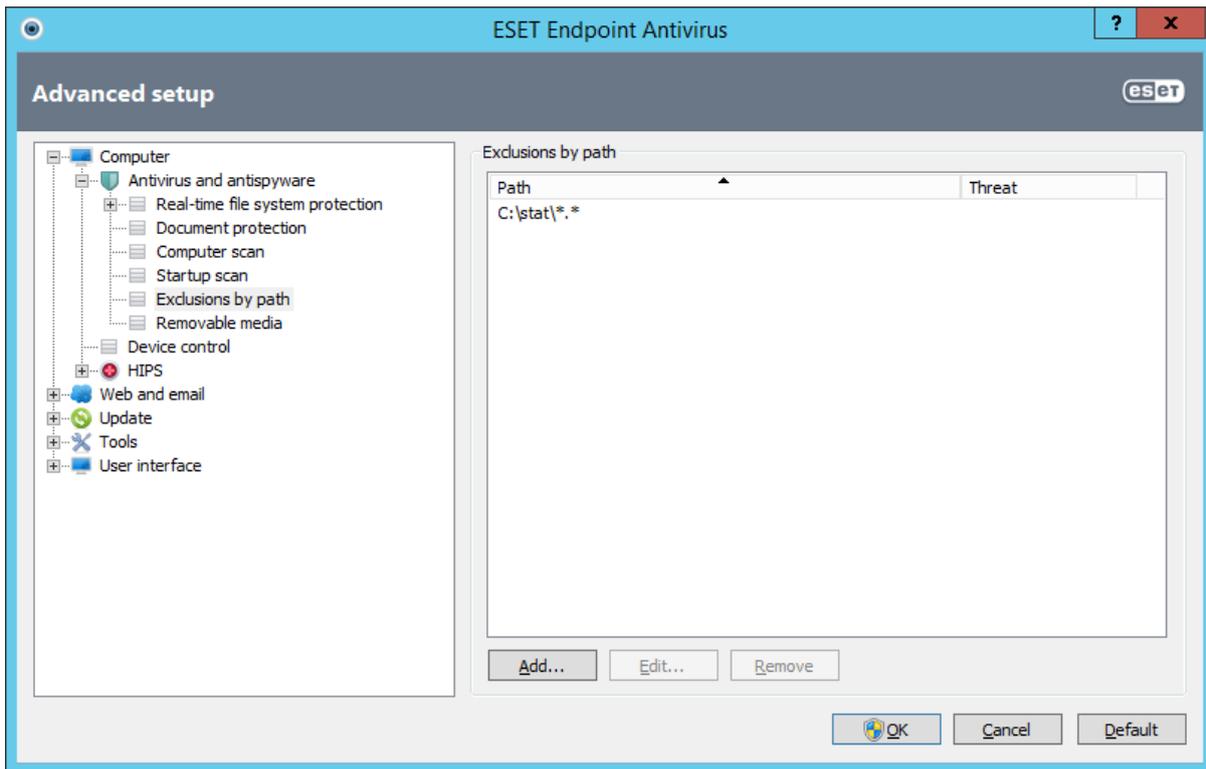
38.1.3 CreateFile() error

Errore: All'inizio dell'esecuzione, pTabs stampa questo errore:

```
CreateFile() error 32 when trying set file time
```

Suggerimento: Escludere la cartella di installazione di pTabs dai percorsi dell'antivirus

Si tratta di un errore riscontrato dopo l'installazione di **NOD32 Antivirus**. Si può indicare all'antivirus di escludere la cartella di pTabs dal controllo.



38.1.4 user_function.rb

Errore: L'esecuzione di pTabs si arresta quasi subito con l'errore:

- Il file "user_functions.rb" non esiste. Controlla la configurazione. (NoResourceError)

Suggerimento: Verificare il puntamento della variabile d'ambiente PTABS_USER

Il file `user_function.rb` viene copiato durante l'installazione di pTabs nella cartella `user` nel percorso di installazione di pTabs, per esempio: `c:/stst/pTabs/user`. Se invece è stata impostata la variabile d'ambiente `PTABS_USER`, questa deve puntare a una cartella contenente il file `user_function.rb`.

Consultare le istruzioni nel paragrafo **Personalizzazioni utente** in *Installazione ed esecuzione*.

38.1.5 Variabile non trovata

Errore: pTabs non trova una variabile, ma sei certo che esista.

Suggerimento: Verificare la corretta scrittura maiuscolo/minuscolo del nome della variabile

I nomi delle variabili in pTabs sono sensibili al maiuscolo/minuscolo, ma di default tutti i nomi delle variabili vengono convertiti in minuscolo.

Pertanto un eventuale nome di variabile con caratteri maiuscoli deve essere utilizzato in minuscolo, oppure è necessario disattivare il parametro di configurazione `vars_to_lower_case`:

```

1 vars_to_lower_case false
2 # oppure
3 vars_to_lower_case_

```

38.2 ptKey

Errore: Cliccando sul pulsante **Test connessione** compare una finestra con l'errore:
Impossibile connettersi: FTP: ...

Suggerimento:

- Configurare il **firewall** per consentire le connessioni FTP dal proprio PC
- Connettersi temporaneamente a Internet tramite una **Internet Key**

E' probabile che un firewall blocchi le connessioni **FTP**. Potrebbe trattarsi di un firewall software sul proprio PC oppure di un firewall hardware nella rete aziendale.

38.3 SciTE

38.3.1 SciTE apre dei file vuoti

Errore: Cliccando o premendo invio su un file **.pt** si apre SciTE, ma invece di caricare il file, apre una serie di file vuoti.

Suggerimento: Nel registro di sistema aggiungere le virgolette attorno al percorso nella voce dell'associazione tra l'estensione **.pt** e **SciTE**

Il problema è generato dal fatto che i nomi di cartelle del percorso di lavoro contengono degli spazi e questo disturba alcune versioni di Windows.

In alcune versioni di Windows, l'associazione di un tipo di file con SciTE non funziona se il percorso contiene spazi. Per risolvere questo problema, la variabile del percorso %1 deve essere circondata da virgolette doppie. Questo può essere fatto direttamente nel *Registro di sistema* o attraverso *Esplora risorse: Strumenti → Opzioni cartella → Tipo di file (Selezionare l'estensione) → Avanzate → open → Modifica*. Modificare il campo "Applicazione utilizzata per eseguire l'azione" in modo simile a:

```
"C:\path\to\SciTE\SciTE.exe" "%1"
```

Le associazioni tra estensioni e programmi possono essere facilmente gestite con un programma come [FileTypesMan](#).

38.3.2 SciTE non sembra configurato per pTabs

Errore: Sembra che alcune funzioni di supporto a pTabs non siano installate:

- i caratteri non sono allineati
- non compare il menu Tools > pTabs
- non compaiono i tooltips
- F1 non apre le pagine di help

Suggerimento: Impostare la variabile d'ambiente `SciTE_HOME`

SciTE non trova la corretta cartella di configurazione. Verificare che la variabile d'ambiente `SciTE_HOME` sia impostata e punti nella sottocartella `pSciTE` all'interno del percorso di installazione di pTabs.

Si può accedere alle variabili d'ambiente tramite il *Pannello di controllo → Sistema → Impostazioni di sistema avanzate → Variabili d'ambiente*.

38.4 Ottenere assistenza

Per ottenere assistenza relativamente a errori verificatisi durante l'utilizzo di *pTabs*, inviare:

- il contenuto completo del **log** di pTabs salvato in formato testo
- lo script **pt** che genera l'errore
- (potrebbe inoltre essere richiesto) un **set di dati** che unitamente allo script riproduce l'errore

Version History

Consiglio: Per aggiornare *pTabs* consultare le istruzioni alla pagina [Aggiornamento e rinnovo](#).

39.1 Beta

39.1.1 1.5.3 (07-02-2016)

- **nets:** nuova sintassi per definire i *net* nelle multiple
- **doc:** output in HTML, PDF e pTabdWeb
- **doc:** ora accetta anche markup [Markdown](#) e [Haml](#)

39.1.2 1.5.1 (01-02-2016)

- **break:** esegue il codice contenuto nel blocco impostando un nuovo filtro a ogni ripetizione. *break* genera dei gruppi di tavole che vengono riconosciuti da *pTabsWeb* e trattati come un unico oggetto con un filtro selezionabile (vedi [Selezione delle osservazioni](#))
- **map:** produce una mappa geografica visualizzando la posizione di ciascun record e/o di coordinate esterne
- **xtab** e **xtab_u** calcolano tabelle mono e bivariate e restituiscono i dati (vedi [Funzioni informative](#))
- **get_data** restituisce dei vettori con i dati delle variabili indicate (vedi [Funzioni informative](#))
- **chart:** nuova istruzione per produrre grafici in modo molto più semplice a partire dalle variabili del file di dati

39.2 Ultima

39.2.1 1.4.2 (26-11-2015)

- *Pairwise Comparisons Test* con più livelli di significatività
- **doc:** inserisce una pagina contenente testo in formato HTML
- **all:** nelle md aggiunge il conteggio delle osservazioni che hanno attivato almeno un item
- **net.new** e **net.none:** possibilità di indicare il codice da contare e quello da generare
- **net.new** e **net.none:** vengono formattati come net se nella tabella sono presenti altri net
- rientri negli item dei net nell'output Excel
- **header:** inserisce un'etichetta immediatamente sotto il pannello

- **.render :ptweb**: genera il database del report per **pTabsWeb**
- **stratum**: etichetta dello strato delle tabelle seguenti (solo *pTabsWeb*)
- **period**: etichetta del periodo delle tabelle seguenti (solo *pTabsWeb*)

39.3 Precedenti

39.3.1 1.3.4 (6-01-2015)

- **sqlserver**: gestisce database Microsoft SQL Server
- **jet**: gestisce file Access nel formato 2003
- **ace**: gestisce file Access nel formato 2007-2010
- **query**: interroga la fonte di dati e restituisce il risultato. Non carica la fonte di dati nel data frame attivo

39.3.2 1.3.2 (21-09-2014)

- **if_none**: crea una nuova variabile uguale a 1 se le variabili indicate non contengono un dato valore
- **if_any**, **if_all**, **if_none**, **if_only** anche per variabili factor
- **weight!** permette di specificare un variabile di ponderazione differente per ciascuna variabile di pannello
- **why** sottocomando delle tabelle che permette di indicare la/le variabile/i di ponderazione di una specifica tabella
- **list** stampa nel log i valori delle variabili dei record del data frame
- **ptKey**: connessione FTP in modalità passiva

39.3.3 1.3.1 (18-06-2014)

- **factor**: controllo sui valori effettivamente esistenti nella variabile
- **factor**: parametro **fill** per generare automaticamente tutti i valori della sequenza numerica
- gestione della dimensione dei font nei file dei temi
- **theme** accetta anche un blocco di istruzioni
- **img** e **img_foot** oltre che un file locale, accettano anche un link web a un'immagine
- **img** e **img_foot** anche per report in formato XLSX
- **img_link** e **img_foot_link** permettono di specificare un link web che si attiva cliccando sulle immagini
- **image** inserisce un'immagine in qualsiasi punto del report
- metodi **multi!** e **single!** dei set per modificare il tipo di set
- **r.source** per eseguire file R
- **pcp_pvalues_print** e **pcm_pvalues_print**: stampa i p-value nel log
- **pc_test_level**: nella nota della tabella cambia tra "livello di confidenza" (default) e "livello di significatività"

39.3.4 1.3.0 (24-03-2014)

- **topline**: report *Top-line* in formato HTML
- **autotab**: produce automaticamente le tabelle per tutte le variabili del file di dati
- **sets.find** ricerca gruppi di variabili singole e multiple
- report anche nel **nuovo formato di Excel** (*Excel Open XML Format Spreadsheet*)
- **callbacks**: funzioni ridefinibili dall'utente che vengono richiamate durante la generazione dei report per inserire dei comportamenti specifici
- gestione **colori** e **temi** per report HTML, PDF e Excel
- **odin.open**: legge i file di dati NIPO Odin usando il file .VAR di NIPO Diana
- **na_to_level**: ricodifica i casi mancanti di una variabile factor creando un nuovo livello se necessario
- **info** restituisce un oggetto con le informazioni relative alla variabile
- **keep** tiene solo le variabili elencate ed eliminare le altre
- nuove e più complete istruzioni per l'installazione e l'aggiornamento di *pTabs*

Avvertimento: Solo se si vuole utilizzare **open.odin** o **fwf.open** è necessario installare la seguente libreria di R (*istruzioni*):

```
install.packages("ffbase")
```

39.3.5 1.2.3 (12-02-2014)

- controllo automatico degli aggiornamenti (configurabile con il parametro *check_new_version_hh*)
- **stack** concatena set di variabili di un data frame
- **unstack** affianca gruppi ripetuti di variabili
- **aggregate** aggrega un data frame applicando le funzioni specificate
- **get_values** restituisce i valori di una variabile

Avvertimento: E' necessario installare le seguenti librerie di R (*istruzioni*):

```
install.packages("DBI")
install.packages("gsubfn")
install.packages("RSQLite")
install.packages("RSQLite.extfuns")
install.packages("sqldf")
```

39.3.6 1.2.2 (17-12-2013)

- *Finestre di dialogo* per ricevere l'input dall'utente e mostrare messaggi: **ask**, **yesno**, **message**, **warning**, **error**
- Metodo **name** per modificare il nome degli output del report
- **panel(n).rowbase** per impostare una base differente per il calcolo delle percentuali di riga
- **Mediana**, **quartili**, **scarto interquartile** e **range** nelle statistiche riassuntive
- **rake**: **:pw** specifica una variabile peso di partenza
- **parse**, **parse!** e **parse?** espande e controlla le espressioni

- **get_selection** e **get_selection_u** restituisce il numero di casi pesati selezionati dall'espressione

39.3.7 1.2.1 (30-09-2013)

- **split_vars**: duplica le variabili in base ai livelli di un'altra variabile
- **derotate**: effettua la de-rotazione di gruppi di variabili
- **ddup**: elimina i record duplicati
- **factor?**: controlla se la variabile è una variabile factor
- **.open!**: permette di aprire un file di dati, eseguire un blocco di istruzioni e salvare un file RData se non esiste, di aprire direttamente il file RData, senza eseguire il blocco, se esiste
- **value**: assegna alle variabili i valori specificati
- *value_if* anche con più di un valore
- *labels*: maggiori controlli nel caricamento delle etichette da un file esterno
- nuovi parametri di configurazione *index_link_html* e *index_link_xls*

39.3.8 1.2.0 (14-05-2013)

- **charts**, **chr** produce grafici nell'output HTML e PDF
- funzioni per estrarre i dati dalle tabelle
- **xcat** produce le singole tabelle *cat* per un set di variabili
- **mresp**: risoluzione dell'errore con variabili di pannello vuote
- **mresp** è ora disattivato per default
- **i_conc**: risoluzione dell'errore con tabelle con solo la colonna di totale
- **low_values**: determina se per le basse numerosità di tabelle e celle devono essere considerati i valori grezzi (:raw) o ponderati (:weight)
- **tcount_u**: totale valori assoluti non ponderati
- **count_u**: valori assoluti non ponderati
- *.render :xls stat_col*: permette di aggiungere una colonna con le etichette delle statistiche

Avvertimento: Installare la libreria R **RColorBrewer**:

```
install.packages("RColorBrewer")
```

39.3.9 1.1.1 (8-04-2013)

- *filter*: modificatori **_not**, **_and**, **_or**
- *select*: modificatore **_not**
- **net_from_groups**: genera i net di una variabile in base a un'altra variabile di raggruppamento
- **move_before**: sposta dei livelli di una variabile factor prima di un determinato livello
- **move_after**: sposta dei livelli di una variabile factor dopo di un determinato livello
- **string**: converte una variabile factor in stringa
- **delete**: cancella le variabili

- **usedvars**: restituisce l'elenco di tutte le variabili utilizzate nelle tavole
- *excel.save* e *excelx.save*: aggiunta delle opzioni **:codebook** per il foglio delle etichette e **:names** per le intestazioni delle colonne
- **none**: nelle tabelle singole con *base_pct :all* aggiunge il complemento di casi
- **netstyle**: attiva/disattiva la differenziazione grafica dei net
- **corner**: imposta un'etichetta nel corner delle tavole. E' indipendente rispetto all'etichetta dei filtri
- **base_pos**: definisce dove deve comparire l'etichetta del filtro: **:corner**, **:title**, **:note**, **:tnote**

39.3.10 1.1.0 (20-03-2013)

- *rake*: **:if** per generare il peso solo su un sottoinsieme dei casi
- *rake*: **:if_not** valore da assegnare al peso per i casi per cui non viene calcolato
- *rake*: **:if_na** comportamento in caso di mancate risposte nelle variabili di ponderazione
- *rake*: valori di ritorno da utilizzare nello script
- *.render :db*: opzione **:clear** per eliminare i dati già presenti con stesso *id* e *period*
- *.render :xls*: **:all** in *stats* per includere tutte le statistiche.
- **table_note** imposta una nota per tutte le tabelle seguenti
- parametro di configurazione **add_load_path** per aggiungere percorsi di ricerca per i file di risorse
- **deffun** e **defpfun** definiscono delle nuove funzioni dello script
- **defvar** e **defpvar** definiscono delle nuove variabili dello script
- metodi **ydump** e **jdump** per salvare dati su file *YAML* e *JSON*
- metodi **yload** e **jload** per leggere dati da file *YAML* e *JSON*
- metodo **p** per gli Array, per stampare nel log il contenuto: `s (:d1_, 10) .p`
- *filter, select*: condizioni in **OR**

39.3.11 1.0.12 (12-03-2013)

- *net*: possibilità di calcolare i net in base a **espressioni**
- *net*: parametro **:stats** per limitare le statistiche da mostrare per i net
- *accentuazioni*: è possibile calcolare le accentuazioni in base **all'indice di concentrazione**
- nuovi metodi **weight!** e **weight_** per attivare e disattivare la ponderazione
- il nome dello script e il percorso possono contenere **spazi**
- **\$p** contiene un riferimento al report corrente. Per eseguire funzioni del report all'interno di altri contesti

39.3.12 1.0.11 (4-02-2013)

- **no_panel** elimina il pannello dalle *hjoin* con solo il totale e aggiunge un'etichetta alle colonne
- **htjoin** facilita la creazione di *hjoin* con solo il totale (sostanzialmente delle grid di tabelle generiche)
- **indice di concentrazione**
- **analisi delle corrispondenze**

Avvertimento: Installare la libreria R **ca**:

```
install.packages("ca")
```

39.3.13 1.0.10 (19-01-2013)

- *pTabsDB*:
 - riga con le basi nelle tabelle
 - etichetta della statistica nelle tabelle
 - identificativo della tabella e della domanda nel nome del foglio dati
 - copia dei dati della tabella negli appunti (invece di produrre la tabella)
 - possibilità di aggiungere una nuova serie di dati per produrre tabelle e grafici di trend
 - possibilità di modificare le etichette delle colonne
- *report.render :db*: switch (:version o :v) per produrre un DB nel vecchio formato
- il default per i parametri *pcm_test_minbase*, *pcp_test_minbase*, *phl_min* e *mhl_min* è ora impostato a **0**

39.3.14 1.0.9 (7-01-2013)

- **pcp_test**: Pairwise Comparisons sulle percentuali
- **spss.time** e **spss.date** convertono le variabili data/tempo di SPSS
- **chron** converte stringhe in data/tempo ed estrae informazioni dalle date
- pagina Twitter: <http://twitter.com/ptabsinfo>

39.3.15 1.0.8 (16-12-2012)

- nuova libreria R per file Excel funzionante anche su Windows 64-bit
- **union** unisce i casi di due file di dati
- **excelx**: importazione ed esportazione dati in formato *Excel 2007*
- **csv**: importazione ed esportazione dati in formato delimitato (*csv*)
- *xport*: anche esportazione dati in formato *XPORT (SAS)*
- *rake*: anche esportazione pesi in formato *XPORT (SAS)* e in formato *Excel 2007*
- attivazione sottocomando *titles* anche per tabelle *tbox*
- **time**: un oggetto con la data e l'ora corrente
- *pSciTE*: voce Tools > pt: istruzioni report per una bozza di script
- *pSciTE*: voce Tools > pt: istruzioni render per una bozza di istruzioni *render*

Avvertimento: E' necessario installare le seguenti librerie di R:

```
install.packages("SASxport")
install.packages("XLConnect")
install.packages("reshape")
```

39.3.16 1.0.7 (29-11-2012)

- *pSciTE*: esecuzione parziale (selezione) dello script con ALT+F5: Tools > Go (Selection)
- *pSciTE*: apertura di istanze multiple per esecuzione contemporanea di più script

39.3.17 1.0.6 (26-11-2012)

- correzione problema interazione *filter* e *base* nelle multiple
- *numeric* accetta anche una singola variabile e non solo liste
- **xord** produce le singole tabelle *ord* per un set di variabili
- **sord** e **ords** producono una tabella *scale* di un set di variabili più le singole tabelle *ord*
- **tbox** produce tabelle riassuntive di top-box
- **xport**: caricamento dati in formato *XPORT* (SAS)
- **get_max_level** restituisce il numero totale di livelli di una variabile
- **varnames** restituisce un vettore di nomi di variabili partendo dall'elenco di variabili dato ed espandendo gli eventuali net
- **size** restituisce il numero di elementi partendo dall'elenco di variabili dato ed espandendo gli eventuali net
- **ids** restituisce un vettore con solo gli elementi con gli indici indicati
- *scale*: le tabelle *scale* sono ora in formato più compatto. Il vecchio formato è riattivabile con `scale_style :long`
- *accentuazioni*: nei livelli per le accentuazioni delle medie si può ora definire qualsiasi multiplo o frazione di deviazione standard (la sintassi è cambiata, consultare il paragrafo specifico)
- *rake*: stampa nel log dell'istruzione SPSS per il match del file dei pesi
- l'istruzione per il *character encoding* dello script (`# coding: iso-8859-1`) è stata trasferita nel file di configurazione

39.3.18 1.0.5 (6-11-2012)

- correzione problemi *sort* con *cat* e *ord*
- istruzioni **down** e **up** negli ordinamenti
- **md_to_panel** per utilizzare delle variabili *multidicotomy* nel pannello
- **label_to_level** copia l'etichetta della variabile come etichetta di un livello
- **na_if** e **value_if** per ricodificare le variabili
- **add_levels** aggiunge livelli alle variabili
- **get_last_id** restituisce l'id dell'ultima tavola generata
- **css.table** imposta gli attributi css per una sola tavola
- **rowlevel** stampa solo righe di livello minore o uguale (1 = righe senza rientro; 2 = righe con un rientro, ecc.)
- **pTabsDB**: template grafici
- nuovi esempi nell'help per l'istruzione *recode*

39.3.19 1.0.4 (19-10-2012)

- parametro **:auto** nel comando **fill** per generare distribuzioni proporzionali alla distribuzione dei casi validi
- parametri **:xhp1**, **:xhp2**, **:xhnote** per specificare l'altezza delle due righe del pannello e della riga delle note

39.3.20 1.0.3 (17-10-2012)

- **create** genera nuove variabili vuote
- **numeric** crea variabili numeriche da variabili factor
- **level_to_label** copia l'etichetta di livello come etichetta della variabile
- aggiunta del nome delle variabili da creare in **expand**
- nuove funzioni per i titoli: **\$vl()**, **\$vn()**, **\$tn()**, **title_template**, **title_prefix**

39.3.21 1.0.2 (14-10-2012)

- **fill** genera dati casuali per simulazioni e test di tavole.
- variabile d'ambiente **PTABS_USER** per definire una cartella con le configurazioni dell'utente
- possibilità di specificare più comandi **net** in una tabella
- **netonly** mostrare solo i net di una tabella
- **mc_to_md** trasforma un set di variabili *multicategory* in un set di variabili *multidicotomy*
- personalizzazione dei colori delle accentuazioni

39.3.22 1.0.1 (8-10-2012)

- **pTabsDB**: output su database *Access* con interfaccia grafica
- **missed** e **omitted** permettono di specificare dei livelli da escludere nelle tabelle
- nuove opzioni per accentuazioni: **:pos**, **:sd3**, **:sd2**, **:sd**
- **exist?** e **existing** per controllare se le variabili esistono e generare elenchi di sole variabili esistenti
- **valid?** e **missing?** per controllare se la variabile contiene dei casi validi o solo casi mancanti
- sottocartella **user** per i file di configurazione e le funzioni dell'utente
- moduli **User::Ptabs** e **User::Report** per le funzioni definite dall'utente
- correzione errore tabella con variabile vuota e senza etichette dei valori
- ricalcolo percentuali di riga su net di variabili singole

39.3.23 1.0.0 (20-10-2012)

- **ptabs.new** senza nessun argomento utilizza come nome dei report il nome dello script
- **get_label**, **get_labels**, **get_levels**
- **get_base**, **get_base_u**, **get_count**, **get_count_u**

39.3.24 0.1.29

- **accentuazioni**
- **t-test** delle medie
- script per normalizzare i file di testo dei questionari: **cleanQ**
- opzione **matrix** e **mx** per esportazioni tabelle in matrici
- formato sintetico nelle espressioni (filter, select, base): `v1 => :valid->:v1 e v1 => :na-> "!v1"`
- sottocomando **base** per specificare una base differente per il calcolo delle percentuali nelle md
- opzione **:rev** in `sort` per invertire l'ordine degli items

39.3.25 0.1.28

- Formato **data** e **tempo**
- script per convertire i file **dft** in file **pt**: **dft2pt**

39.3.26 0.1.27

- **Somma** nelle statistiche riassuntive (`sum`)

39.3.27 0.1.26

- **ptGen**: input in formato **Triple-S**
- legenda dei pesi con etichette anche per tabelle tipo *scale*

39.3.28 0.1.25

- Caricamento dati in formato **Triple-S**
- **compress** genera una variabile factor da più variabili. Il caso più comune è la ricostruzione di una variabile singola a partire da variabili dummy.
- il sottocomando **w** accetta:
 - `:legend` per inserire una nota con la descrizione dei pesi (sostituisce il comando `autonote`)
 - un template per inserire l'indicazione del peso nell'etichetta
- **ptGen**: correzione errore con question usate come PAGE

39.3.29 0.1.24

- Migliore integrazione di **SciTE**
- Correzione errore hole count con item multipla senza nessun caso attivo
- Eliminazione warning standard deviation

39.3.30 0.1.23

- Output Excel con più fogli con differenti impostazioni di tavole
- Switch **-l** per salvare il log in un file

39.3.31 0.1.22

- Nuovo output Excel molto più veloce. Usa JExcelAPI (Java) invece di Spreadsheet (Ruby)

39.3.32 0.1.21

- correzione del calcolo del numero medio di risposte nelle multiple
- correzione dei problemi con pannello senza totale

39.3.33 0.1.20

- net in grassetto negli excel
- drop_levels con più variabili anche senza parentesi quadre
- **hide**: elimina righe dalla tabella. Rispetto a *drop* agisce al momento della *stampa* della tabella. Per esempio può eliminare l'unico item di un net con una sola voce.
- **hcm**: hole count per variabili multiple (anche in *ptGen*)

39.3.34 0.1.19

- excel formattato
- unione labels uguali delle variabili in pannello e in fiancata ora anche nei file excel
- tabelle **hjoin**: unione orizzontale di più tavole con la stessa fiancata
- **net_only**: variabile net che verifica la presenza esclusiva di una valore in un set di item rispetto a tutti gli altri

39.3.35 0.1.18

- nuovo programma di installazione
- **ptKey** per la gestione delle licenze

39.3.36 0.1.17

- **ptGen**: generazione istruzioni *hole count*
- **zerorow**: attiva/disattiva la presenza delle righe con casi uguali a zero nelle frequenze delle variabili (solo item, non totali)
- correzione del problema dei link degli indici in excel in presenza di nomi di foglio con caratteri particolari e spazi
- nuova gestione dei **formati** con la possibilità di modificare il separatore dei decimali e quello delle migliaia
- **xls_round** e **xls_matrix**: determinare se negli output excel e a matrice i dati devono essere arrotondati secondo il formato di stampa o devono invece mantenere tutti i decimali

39.3.37 0.1.16

- **ss**: genera liste di liste di variabili: `ss ("v11.#2.#1", 24, 40)`
- migliore gestione del totale nelle tabelle con più variabili (per es. la tabella campione)

39.3.38 0.1.11

- **interaction**: combina i livelli di due o più fattori in un'unica variabile
- **dummies**: genera un set di variabili dummy da una variabile factor
- **switch**: scambia due livelli di una variabile factor
- **reorder**: riordina i livelli di una variabile factor

39.3.39 0.1.10

- Merge data frame
- Esportazione **tabelle** come matrice di dati nei formati csv, Excel e Access
- Indice con link alle tabelle nei report Excel
- Esportazione **dati** in SPSS con etichette
- Esportazione **dati** in Excel con conversione factor in numeriche e foglio con le etichette
- Esportazione **dati** in formato testo con conversione factor in numeriche, file con tracciato e file etichette

39.3.40 0.1.9

- Input dati in formato testo fisso
- Output dati in formato Excel e testo
- rake: possibilità di salvare i pesi anche in formato testo
- logo nell'output PDF
- s/seq cerca le variabili nel file di dati

39.3.41 0.1.8

- Output report in formato testo
- Corretto errore nei net delle singole

39.3.42 0.1.7

- Cambiata la sequenza interna di esecuzione del programma: per ciascuna tabella vengono subito eseguiti i calcoli in R e caricati i risultati in Ruby
- il metodo `run` del report non è più necessario
- `fixed`: negli ordinamenti, specifica gli item che devono rimanere in fondo
- ID tabella
- Sdoppiamento `filter`, selezione temporanea, da `select`, selezione definitiva.
- Procedura per la ponderazione dei dati: `rake`

Avvertimento: Installare la libreria R `survey`:

```
install.packages("survey")
```

39.3.43 0.1.6

- HTML per IE parzialmente corretto (non ancora i rientri nei net)
- Sintassi per i net nelle multidicotomy completamente cambiata:

```
net.new :n7_1, "TOT. CITAZIONI", s(:d7_,1..3,:n7_11,:n7_12)
net.none
```
- Net a più livelli
- Nelle multidicotomy totale risposte e media risposte: `mresp`, `nresp`
- `noresp`: variabili che non devono essere conteggiate nel numero delle risposte (non risponde, non sa, ecc.)
- `rename` delle variabili
- `rep` anche per hash

39.3.44 0.1.5.2

- `last` oltre a `first`
- `keep` e `drop` per tenere solo o eliminare le righe indicate
- `sort` alfabetico
- `report` e grafici `missing` (`show_missing`)

Avvertimento: Installare altre 2 librerie di R:

```
install.packages("mice")
install.packages("VIM")
```

Problemi risolti

- PDF: raddoppio dell'indice se la tabella va in un'altra pagina
- Nel PDF se una domanda va su più pagine riportare la riga del totale
- Selezione su valori validi/valori mancanti: `=> :na` e `=> :valid`

39.3.45 0.1.5

- Ci sono un po' di cambiamenti su alcuni elementi generali per cercare di avere una maggiore chiarezza, per esempio: `ptabs.new_report` (o `ptabs.new`) e non `new_report` (che comunque funzionano ancora)
- Mette (di default) le % di riga a totale, il vecchio comportamento è invece ottenibile con:

```
trowpct_
tcolpct!
```
- Trasformazione in `factor` anche senza dichiarazione dei livelli
- Multiple su `factor` (si/no) ok ma occorre dichiarare i valori

39.3.46 0.1.4

- `net_after` non esiste più, è sostituito da:
 - `net_pos :before` (il default)
 - `net_pos :after`

- net_pos :top
- net_pos :bottom

39.3.47 0.1.3

Avvertimento: Installare un'altra libreria di R:

```
install.packages("RODBC")
```

\$tn(), 128
\$vl(), 128
\$vn(), 128
%include, 38
NA, 81

accentuazioni, 143
Access, 49
access, 49
ace, 49
add_levels, 99
add_load_path, 13
aggiornamento, 10
all, 67, 87
Analisi delle corrispondenze, 148
ask, 176
attivazione, 1
autotab, 180

base, 13, 67
base_multi, 13, 67
base_pct, 13, 67
base_pos, 13, 87
base_u, 13
break, 87
break_char, 13

ca, 148
ca_arrows, 13
ca_contrib, 13
ca_db, 13
ca_dims, 13
ca_mass, 13
ca_out, 13
ca_plot_frm, 13
ca_plots, 13
Callbacks, 180
 after_report, 180
 before_exit, 180
 before_report, 180
 render_title, 180
camp, 67
cases, 87
cat, 67
cbyc, 67

cchart, 13
chart, 156
charts, 156
check_new_version_hh, 13
Chi-quadrato, 143
chisq_formats, 13
chisq_labels, 13
chisq_test, 13, 143
chisq_warning_cel_freq, 13
chr, 156
chr_caption, 13
chr_gauge_caption, 13
chr_gauge_height, 13
chr_gauge_label, 13
chr_gauge_options, 13
chr_gauge_width, 13
chr_height, 13
chr_legend_pos, 13
chr_width, 13
chron, 119
colori, 13
 theme#.yml, 13
colors, 156
colpct, 13
column, 67
columns, 67
common, 122
compress, 99
configurazione, 13
coord, 67
copy, 99
copy_label, 99
copy_levels, 99
corner, 13, 67
count, 13
count_u, 13
create, 99
css, 122, 174
css.chart, 122
css.load, 122
css.map, 122
css.table, 122
css.html, 13
css_pdf, 13
css_tl, 13

CSV, 49
csv, 49
csv.open, 49
csv.save, 49
csv1, 49
csv1.open, 49
csv1.save, 49
csv2, 49
csv2.open, 49
csv2.save, 49
csv_col_sep, 13
csv_dec_sep, 13
cvars, 67

DB2ppt, 13
ddup, 87
dec_sep, 13
deffun, 180
defvar, 180
delete, 99
delete_table_objects, 13
derotate, 99
df?, 128
Diana, 49
doc, 174
dot_to_underscore, 13
drop, 67
drop_levels, 99

empty_table, 13
empty_table_msg, 13
encoding, 13
error, 176
esecuzione, 1
espressioni, 112
etichette, 99
Excel, 49
excel, 49
excel.open, 49
excel.save, 49
excelx, 49
excelx.open, 49
excelx.save, 49
exist?, 128
existing, 128
expand, 99

factor, 99
fake_weight_varname, 13
file
 install_packages.R, 13, 203
 pdf_footer.html, 13, 203
 pdf_header.html, 13, 203
 ptabs.css, 13, 203
 ptabs.yml, 13, 203
 ptabs_pdf.css, 13, 203
fill, 99
filter, 87
first, 67
fixed, 67
footer, 13, 42
footnote, 67
frm_base, 13
frm_base_u, 13
frm_coord, 13
frm_count, 13
frm_count_u, 13
frm_i_conc, 13
frm_iqr, 13
frm_mean, 13
frm_median, 13
frm_mresp, 13
frm_nresp, 13
frm_pct, 13
frm_q0, 13
frm_q1, 13
frm_q2, 13
frm_q3, 13
frm_q4, 13
frm_range, 13
frm_sd, 13
frm_se, 13
frm_sum, 13
ftp, 180
fwf, 49
fwf.open, 49
fwf.save, 49

get_base, 128
get_base_u, 128
get_codes_and_levels, 128
get_counts, 128
get_label, 128
get_labels, 128
get_last_id, 128
get_levels, 128
get_max_level, 128
get_selection, 128
get_selection_u, 128
grafici, 156
grid, 67

haml, 174
hc, 59
hc_maxlev, 13
header, 67
headers, 67
hide, 67
hjoin, 67
htjoin, 67
html, 174
html_print_footer, 13
html_print_index, 13
html_print_nav, 13
hyphen_char, 13

i_conc, 13
ids, 122

if_all, 99
 if_any, 99
 if_only, 99
 image, 172
 img, 13, 172
 img_foot, 13, 172
 img_foot_link, 13
 img_link, 13
 in_groups_of, 67, 122
 inc, 42
 index_link_html, 13
 index_link_xls, 13
 index_title, 13
 installazione, 1
 interaction, 99
 ipf, 137
 iqr, 13

 java_heap_xms, 13
 java_heap_xmx, 13
 java_home, 13
 jdump, 122
 jet, 49
 jload, 122
 join_col_labels, 13
 join_row_labels, 13
 jre_home_32, 13
 jre_home_64, 13
 jsdk_home_32, 13

 keep, 67, 99

 label, 99
 label_base, 13
 label_base_u, 13
 label_campione, 13
 label_colpct, 13
 label_conf_level, 13
 label_corr, 13
 label_count, 13
 label_count_u, 13
 label_ctotal, 13
 label_freq_count, 13
 label_freq_pct, 13
 label_i_conc, 13
 label_iqr, 13
 label_mean, 13
 label_median, 13
 label_mresp, 13
 label_note_pcm, 13
 label_note_pcp, 13
 label_nresp, 13
 label_one_tail, 13
 label_pcm_test, 13
 label_pcp_test, 13
 label_q0, 13
 label_q1, 13
 label_q2, 13
 label_q3, 13
 label_q4, 13
 label_range, 13
 label_rowpct, 13
 label_rtotal, 13
 label_sd, 13
 label_se, 13
 label_sig_level, 13
 label_sum, 13
 label_to_level, 99
 label_two_tail, 13
 label_yates, 13
 labels, 67
 last, 67
 level_to_label, 99
 levels, 99
 librerie, 1, 10
 list, 128
 low_cell, 13
 low_cell_count, 13
 low_cell_label, 13
 low_table, 13
 low_table_count, 13
 low_table_msg, 13
 low_values, 13
 lprefix, 99
 lsuffix, 99

 map, 165
 map_active_layer, 13
 map_bound_serie, 13
 map_center, 13
 map_clustered, 13
 map_data_name, 13
 map_height, 13
 map_layers, 13
 map_layers_control, 13
 map_marker_icons_fa, 13
 map_marker_icons_mk, 13
 map_marker_size, 13
 map_marker_type, 13
 map_overlays, 13
 map_point_label, 13
 map_scale, 13
 map_search, 13
 map_search_text, 13
 map_show_position, 13
 map_show_series_info, 13
 map_width, 13
 map_zoom_control, 13
 map_zoom_slider, 13
 markdown, 174
 matrix, 13
 matrix_round, 13
 mc, 67
 mc_to_md, 99
 md, 67
 md_false_value, 13
 md_ft, 67

md_tf, 67
 md_to_panel, 99
 md_true_value, 13
 mean, 13
 median, 13
 merge, 93
 message, 176
 mhl, 13
 mhl_lev, 13
 mhl_min, 13
 mhl_minus, 13
 min_split_chars, 13
 missing, 99, 153
 missing?, 128
 move_after, 99
 move_before, 99
 mresp, 13, 67
 multiple_sheets, 13
 mx, 13

 n, 87
 na_if, 99
 net, 81
 net_from_groups, 81
 net_pos, 13, 81
 netonly, 13, 81
 nets, 81
 netstyle, 13
 next, 122
 nil_weight_label, 13
 nil_weight_legend, 13
 no_panel, 67
 none, 67, 81
 noresp, 67, 81
 note, 67
 note_missed, 13
 note_omitted, 13
 now, 128
 nresp, 13, 67
 numeric, 99

 ODBC, 49
 odbc, 49
 ODIN, 49
 omitted, 67
 ord, 67
 ords, 67

 p, 122
 Pairwise Comparisons, 143
 palette, 13, 156
 panel, 64
 panels, 64
 parse, 112
 pc_sep_chr, 13
 pc_skip_first, 13
 pc_test_level, 13
 pcm_pvalues_print, 13
 pcm_test, 13, 143
 pcm_test_adj, 13
 pcm_test_minbase, 13
 pcm_test_note, 13
 pcm_test_sig, 13
 pcm_test_tails, 13
 pcp_pvalues_print, 13
 pcp_test, 13, 143
 pcp_test_adj, 13
 pcp_test_minbase, 13
 pcp_test_note, 13
 pcp_test_sig, 13
 pcp_test_tails, 13
 pcp_yates_corr, 13
 pdf_encoding, 13
 pdf_footer, 13
 pdf_header, 13
 pdf_page_row_number, 13
 pdf_zoom, 13
 period, 44, 87
 phl, 13
 phl_ic, 13
 phl_lev, 13
 phl_min, 13
 phl_minus, 13
 post-stratificazione, 137
 print, 42
 PSPP, 49
 pt, 40
 ptabs, 40
 csv, 40
 inc, 40
 new, 38, 40
 new_report, 40
 open_report, 40
 pTabsDB, 193
 pTabsWeb, 44, 199

 q0, 13
 q1, 13
 q2, 13
 q3, 13
 q4, 13
 qrt, 13

 R, 1, 10, 49
 r, 49
 r.exec, 122
 r.open, 49
 r.save, 49
 r.source, 122
 r_log_width, 13
 rake, 137
 range, 13
 recode, 99
 rename, 99
 render, 38, 42, 44
 db, 44
 Excel, 44
 Html, 44

Pdf, 44
 ptweb, 44
 text, 44
 reorder, 99
 rep, 122
 rev, 122
 rinnovo, 10
 rotazioni, 99
 row, 67
 rowlevel, 13
 rowpct, 13
 rows, 67
 rtotal, 13
 Ruby, 35
 rvars, 67

 s, 67, 122
 sample, 87
 SAS, 49
 save, 42
 scale, 67
 scale_style, 13
 schart, 13
 SciTE, 1
 sd, 13
 se, 13
 seed, 99
 select, 87
 selection, 87
 seq, 122
 sets, 62, 115
 sets_all, 13
 sets_display, 13
 sets_pref, 13
 sets_sep, 13
 sets_start, 13
 sheet_index_name, 13
 show_dataframe, 128
 show_missing, 153
 single_panel, 13
 size, 128
 sord, 67
 sort, 67
 split_vars, 99
 SPSS, 49
 spss, 49
 spss.date, 119
 spss.open, 49
 spss.save, 49
 spss.time, 119
 spss_max_levels, 13
 spss_string_as_factor, 13
 spss_trim_factor_names, 13
 spss_trim_values, 13
 spss_use_missings, 13
 spss_use_value_labels, 13
 SQL Server, 49
 sqlserver, 49

 ss, 67, 122
 sseq, 122
 SSS, 49
 stack, 93
 start, 122
 stats, 81
 stop, 122
 stratum, 44, 87
 string, 99
 styles, 156
 subtitle, 42
 sum, 13
 switch, 99
 -rs, 13, 203
 -sc, 38
 system, 180

 t-test, 143
 tabelle, 67
 autotab, 67
 camp, 67
 cat, 67
 cbyc, 67
 grid, 67
 join, 67
 mc, 67
 md, 67
 ord, 67
 ords, 67
 scale, 67
 sord, 67
 tbox, 67
 xcat, 67
 xord, 67
 table_continues, 13
 table_note, 13
 tbox, 67
 tcolpct, 13
 tcount, 13
 tcount_u, 13
 Text, 49
 theme, 13
 ths_sep, 13
 time, 128
 title, 42, 67
 title_prefix, 13
 title_template, 13
 titles, 67
 titolo report, 42
 titolo tabella, 67, 180
 tl_alpha, 13
 tl_file_sx, 13
 tl_highlight_base, 13
 tl_index, 13
 tl_index_title, 13
 tl_join, 13
 tl_maxlev, 13
 tl_nodata, 13

tl_notitle, 13
today, 128
top-line, 62
total_after, 13
total_varname, 13
totals, 13
Triple-S, 49
trowpct, 13
txt_cell_size, 13
txt_net_id, 13
txt_net_indent, 13
txt_page_height, 13
txt_page_width, 13
txt_side_size, 13

underscore_to_dot, 13
union, 93
usedvars, 128

valid?, 128
value, 99
value_if, 99
VAR, 49
var, 81
varlabel, 13
varnames, 128
vars, 67
vars_to_lower_case, 13
ViewSav, 49
vjoin, 67

w, 67
warning, 176
wby, 67
weight, 137
weight_, 137
weight_legend_label, 13
weights, 67
weights_vars_action, 13
wkhtmltopdf, 1, 13
workspaces, 93
ws, 93

xcat, 67
xhnote, 13
xhp1, 13
xhp2, 13
xls_round, 13
xls_width_data, 13
xord, 67
XPORT, 49
xtab, 128
xtab_u, 128

ydump, 122
yesno, 176
yload, 122

zero, 13
zero_xls, 13
zerorow, 13