

Un documento Quarto con codice integrato

Domenico Vistocco

2023-10-24

INDICE

Introduzione	2
Primo paragrafo	2
Secondo paragrafo: elenchi	3
Terzo paragrafo (“tecnico”)	3
Un sottoparagrafo con <code>chunk</code> di codice R	3
Un sottoparagrafo con alcuni grafici	4
Un sottoparagrafo matematico	5
Un sottoparagrafo “tabellare”	6
Tabelle statiche	6
Tabelle dinamiche	6
Alcuni dettagli	8
Citazioni bibliografiche	8
Note a piè di pagina	8
Riferimenti incrociati	8
Riferimenti bibliografici	9

i Nota

Puoi scaricare [qui il progetto RStudio in formato compresso](#) con tutto il necessario per compilare sul tuo PC i report Quarto (codice R integrato, codice R esterno, report parametrico con codice R integrato, report parametrico con codice R esterno).

Introduzione

Ecco una breve *introduzione*. Ricorda che puoi passare ad un differente formato del documento di output senza cambiare questo documento sorgente: basta scegliere il formato desiderato dal sottomenu del tasto **Render**, disponibile quando si specificano più formati di output nella sezione **YAML** del documento.

Primo paragrafo

Per la composizione del report puoi sfruttare la sintassi del linguaggio **Markdown** syntax. Per dettagli puoi consultare il materiale su [questo link](#), dove trovi una guida dettagliata al sistema **Quarto** organizzata per sezioni. Una guida sulla sintassi base di **Markdown** è consultabile nella sezione **Authoring** seguendo il [link Markdown Basics](#).

Questo è un esempio di citazione (Tolstoj ?):

Tidy datasets are all alike but every messy dataset is messy in its own way. –
Hadley Wickham

Ricorda anche di consultare i capitoli 29 e 30 del libro **R for Data Science** (2nd edition) (Wickham and Grolemund 2017), disponibile [online qui](#). E' senza dubbio la guida di riferimento al sistema:



Figure 1: Questa è un esempio di didascalia

Secondo paragrafo: elenchi

Puoi inserire elenchi puntati:

- punto 1 di un elenco puntato (non numerato)
- punto 2
- punto 3

oppure elenchi numerati:

1. punto 1 di un elenco numerato
2. punto 2 di un elenco numerato
3. punto 3 di un elenco numerato

Puoi anche innestare più elenchi come di seguito:

1. punto 1 di un elenco numerato
 - primo punto di un elenco puntato innestato
 - secondo punto di un elenco puntato innestato
2. punto 1 di un elenco numerato

Puoi anche utilizzare una `definition list`, ovvero elenchi in cui i punti sono associati ad una voce:

Punto 1 Questo è il primo punto di una `definition list`.

Punto 2 E questo è il secondo.

Terzo paragrafo (“tecnico”)

Un sottoparagrafo con chunk di codice R

Sfruttando le opzioni `echo` ed `eval` dei chunk di codice puoi decidere cosa includere nel documento in output. Ecco un esempio di un chunk che è inserito in output ma non valutato (`echo: true` e `eval: false`):

```
fit <- lm(dist ~ speed + I(speed^2), data = cars)
b <- coef(fit)      # coefficienti del modello di regressione
                     # (che sfrutto dopo per mostrare l'uso del codice inline)
summary(fit)        # sintesi del modello di regressione
```

Impostando invece `echo: false` e `eval: true` nel report è inserito solo l'output del codice ma non il codice sorgente R:

Call:

```
lm(formula = dist ~ speed + I(speed^2), data = cars)
```

Residuals:

Min	1Q	Median	3Q	Max
-28.720	-9.184	-3.188	4.628	45.152

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.47014	14.81716	0.167	0.868
speed	0.91329	2.03422	0.449	0.656
I(speed^2)	0.09996	0.06597	1.515	0.136

Residual standard error: 15.18 on 47 degrees of freedom

Multiple R-squared: 0.6673, Adjusted R-squared: 0.6532

F-statistic: 47.14 on 2 and 47 DF, p-value: 5.852e-12

Il codice è inserito in un riquadro in tutti i formati di output del documento.

Questo testo sembra un chunk di codice ma ne utilizza solo la stessa formattazione.

E' ottenuto senza specificare {r} nella testa del chunk.

Un sottoparagrafo con alcuni grafici

I grafici creati dal codice R sono inseriti direttamente nel report in output ed è possibile impostarne le opzioni di visualizzazione:

```
par(mfrow = c(2, 2), pch = 20, mar = c(4, 4, 2, .1), bg = 'white')  
plot(lm_fit)
```

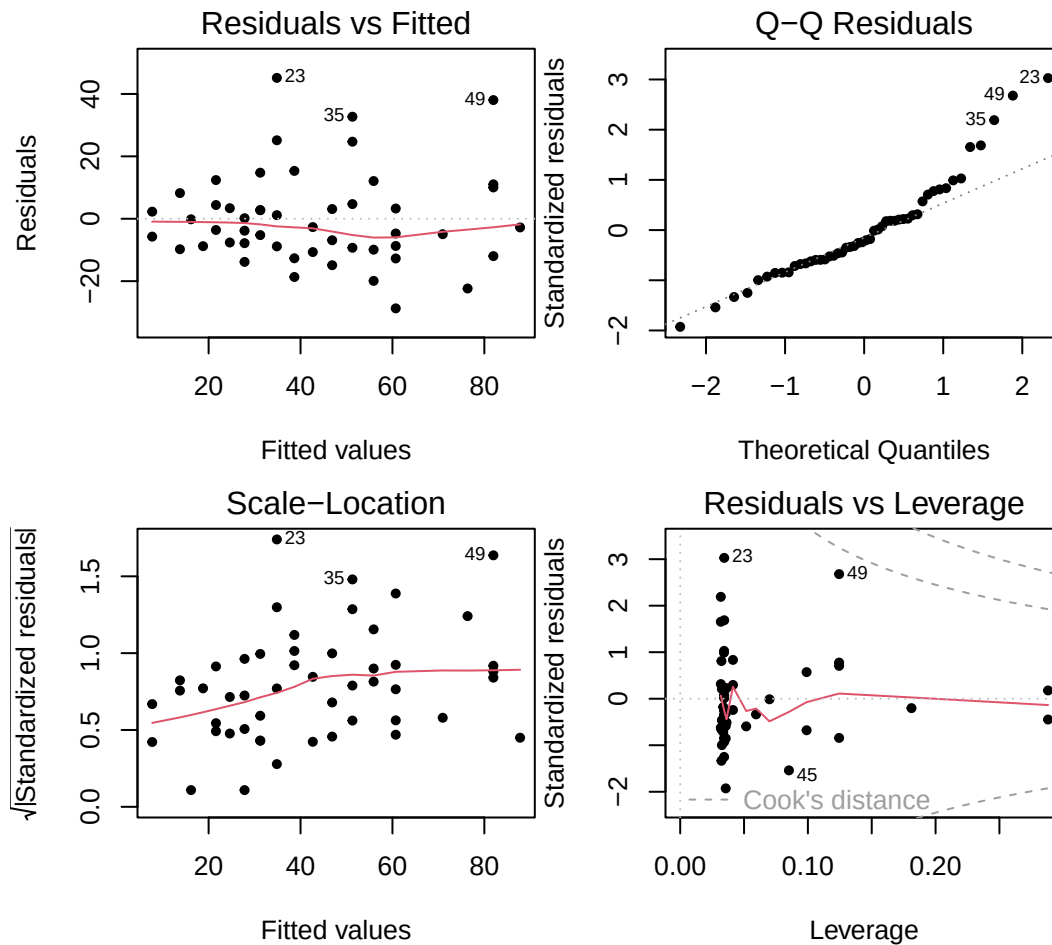


Figure 2: Grafici diagnostici di regressione

Un sottoparagrafo matematico

Ecco un esempio di equazione in modalità `display`, il famoso modello di regressione:

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

E' possibile scrivere anche un'equazione in modalità `inline`. Ecco l'equazione di regressione che sfrutta anche codice R inline (coefficienti del modello di regressione salvati in un chunk di codice precedente): $\hat{y} = 2.4701378 + 0.9132876 x$.

Ed ecco un esempio di equazione complessa (copiata da una pagina di Wikipedia relativa al

LaTeX):

$$\frac{1}{1 + \frac{1}{2 + \frac{1}{3 + x}}} + \frac{1}{1 + \frac{1}{2 + \frac{1}{3 + x}}}$$

Un sottoparagrafo “tabellare”

Tabelle statiche

Puoi inserire tabelle statiche in un report sfruttando la sintassi di composizione **Markdown**. Di seguito un esempio di una semplice tabella.

Tabella: Un esempio della sintassi di una tabella in **Markdown**.`{#tbl-tabella-statica}`

Table 1: Didascalia della tabella

Right	Left	Center	Default
1	2	3	4
5	6	7	8
9	10	11	12

Tabelle dinamiche

Puoi inserire una “spartana” tabella prodotta dal codice **R** semplicemente stampandola e questa verrà formattata così come è stampata nella console di **R**:

```
head(cars, 7)
```

```
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
7    10   18
```

Sono disponibili diversi package che ti permettono con poca fatica di formattare in maniera professionale delle tabelle. Ecco un esempio della funzione `kable()` disponibile nel package **knitr**:

```
library(knitr)
kable(head(cars, 7))
```

Table 2: Una tabella sfruttando `kable` {`knitr`}

speed	dist
4	2
4	10
7	4
7	22
8	16
9	10
10	18

E' importante sottolineare che per sfruttare una funzione `R` disponibile in un package è necessario caricare il package in un **chunk** di codice prima di sfruttare la funzione. In altre parole il workspace dedicato al report **Quarto** sfrutta un ambiente di lavoro dedicato. E' naturalmente possibile sfruttare la sintassi che permette di utilizzare una funzione specificando il **namespace** in cui questa è definita senza caricare il package in memoria. Ecco un esempio che sfrutta la funzione `gt`, disponibile nell'omonimo package, per un altro tipo di formattazione dell'output tabellare:

```
gt::gt(head(cars, 7))
```

Table 3: Una tabella sfruttando `gt` {`gt`}

speed	dist
4	2
4	10
7	4
7	22
8	16
9	10
10	18

Alcuni dettagli

Citazioni bibliografiche

I riferimenti bibliografici in un documento Quarto sono gestiti utilizzando la sintassi del BibTeX. Esistono molte guide online anche molto dettagliate. Una tra queste è [The quick BibTeX guide])(<https://www.bibtex.com>). In realtà per essere operativo ti basta partire dal file `.bib` associato a questo report e consultare le differenti tipologie di fonti che puoi citare su [questa pagina](#) della suddetta guida.

Una volta preparato il file `.bib` con le citazioni, per inserirle nel testo si tratta davvero di un gioco da ragazzi. Non ci credi? Ecco le citazioni del package **knitr** (Xie 2015) e del manuale R (R Core Team 2013) generate automaticamente.

Note a piè di pagina

Ci sono due modi di inserire note a piè di pagina. Il primo consiste nell'inserire nel testo un richiamo con un indice che viene collegato automaticamente alla nota¹, come in questo caso.

In alternativa sono possono inserire anche note sfruttando una sintassi `inline`, come nell'esempio di questa seconda nota².

Nel testo in output non c'è alcuna differenza tra i due tipi di nota.

Riferimenti incrociati

I riferimenti incrociati in un documento possono essere di diversa tipologia. Ecco un esempio di utilizzo in un elenco (`examples` in Pandoc):

- (1) Questo è un primo esempio $2 + 2$.
- (2) Questo è un secondo esempio $2 - 2$. Stessi numeri ma risultati differenti?

Sebbene (1) sia molto semplice, ti invito a pensare in cosa differisce dal secondo (2).

Un altro tipo molto utile di riferimenti incrociati sono quelli relativi a tabelle e figure. Ti invito a guardare il codice in corrispondenza della Figure 1 per un esempio relativo ad un riferimento ad una figura nel testo, e a quello delle Table 1, Table 2 e Table 3 per i riferimenti alle tabelle.

E' importante sottolineare che le etichette per far riferimento alle figure devono iniziare necessariamente con la stringa `fig-`, mentre quelle relative alle figure con la stringa `tbl`. Per cambiare le etichette e le impostazioni di default relativi ai riferimenti incrociati puoi sfruttare

¹Ciao, sono la prima nota a piè di pagina.

²Ciao di nuovo, sono un'altra nota.

la voce `crossref` nella sezione `YAML`. Ecco ad esempio le impostazioni per un documento in italiano (i valori di default sono relativi alla lingua inglese):

```
crossref:
  fig-prefix: Figura    # (default: "Figure")
  tbl-prefix: Tabella   # (default: "Table")
  ref-hyperlink: false # (default: true)
```

Maggiori dettagli sui riferimenti incrociati sono disponibili sulla [corrispondente pagina](#) della documentazione `Quarto`.

Per chiudere, puoi fissare il titolo della sezione in cui verranno inseriti i riferimenti bibliografici semplicemente inserendo una sezione vuota (solo titolo) in calce al documento.

Riferimenti bibliografici

- R Core Team. 2013. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <http://www.R-project.org/>.
- Wickham, Hadley, and Garrett Grolemund. 2017. *R for Data Science*. O'Reilly. <http://r4ds.had.co.nz/>.
- Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. <http://yihui.name/knitr/>.