



02 - Chunck di codice

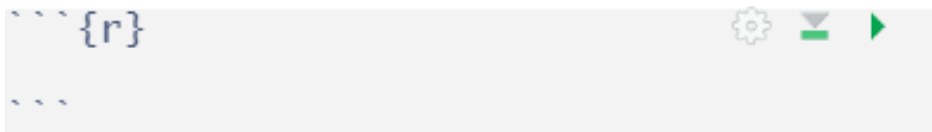
Ottavia M. Epifania

Cosa sono

Sono quello che rende `RMarkdown` così “speciale”.

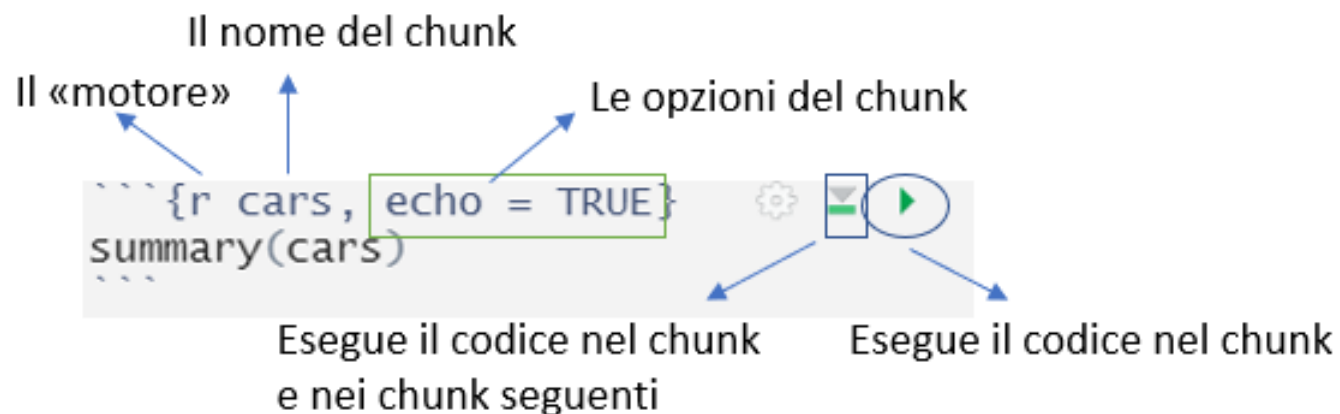
Permettono di integrare i documenti con i risultati delle analisi e con i codici usati per ottenerle (tutti i miei materiali supplementari alle riviste sono prodotti con `RMarkdown`)

Per crearli, basta usare la combinazione di tasti `Ctrl + Alt + i` (`Cmd + Alt + i` se si ha il Mac):



(si può anche usare l'icona )

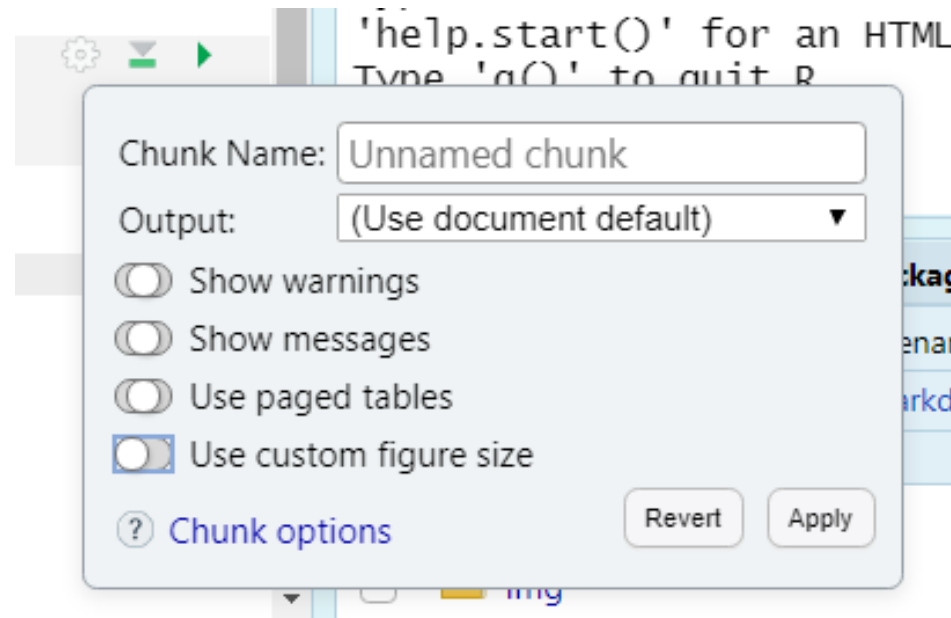
Il file di default



Tutte le opzioni che si vogliono dare al chunk (e.g., come mostrare i risultati, come mostrare le figure, se mostrare o meno le figure ecc.) vanno separate dalla virgola e scritte a mano

Oppure

Oppure si selezionano le opzioni base desiderate dal menu che appare cliccando sulla rotellina del chunk stesso:



A [questa pagina](#) è disponibile una lista esaustiva degli argomenti dei chunk di codice

Stessa cosa ma più schematica disponibile [qui](#)

Setup chunk

Qui vanno specificate tutte le opzioni di default che vogliamo applicare al nostro documento, i pacchetti che pensiamo di usare per le nostre analisi, i dati, gli environment (per fare prima) eccetera.

Ad esempio, il setup chunk di queste slide è:

```
```${r} setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE,
 fig.align = "center",
 out.width = "50%")

library(emoji)
library(knitr)
```
```

A meno che non venga specificato altrimenti, ogni chunk del documento seguirà le opzioni di default specificate nel setup chunk

Your turn!

- Importate il vostro dataset nel setup chunk e assegnatelo a un oggetto:

```
> data(nome_dataset)
> dati = nome_dataset
```

- Create un nuovo chunk dove esplorate il dataset (`head(dati)`)

ADVANCED

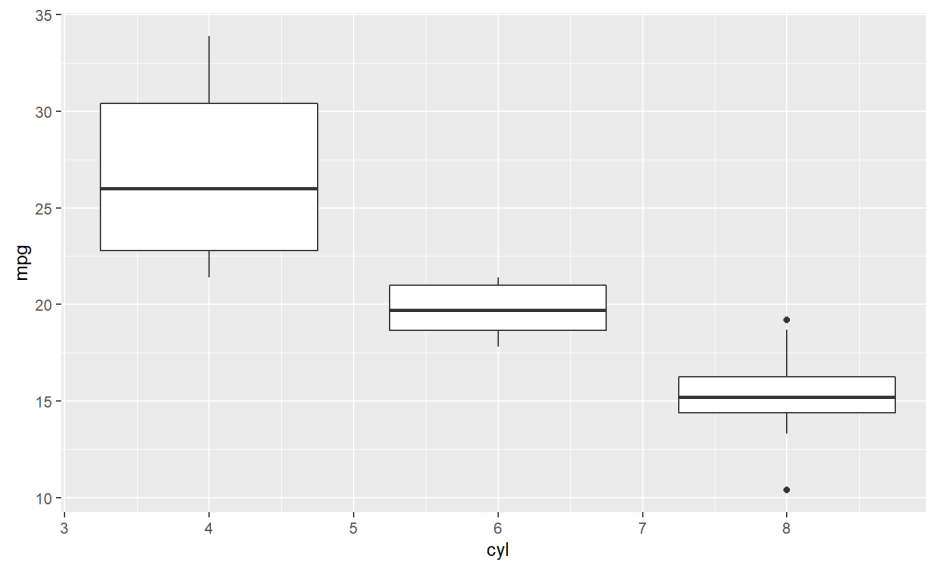
- Fate sparire il codice di \mathbb{R} dall'output
- Salvate l'environment di \mathbb{R} e richiamatelo nel setup chunk (hint: `load()`)

Codice e risultati

Codice eseguito e risultati mostrati

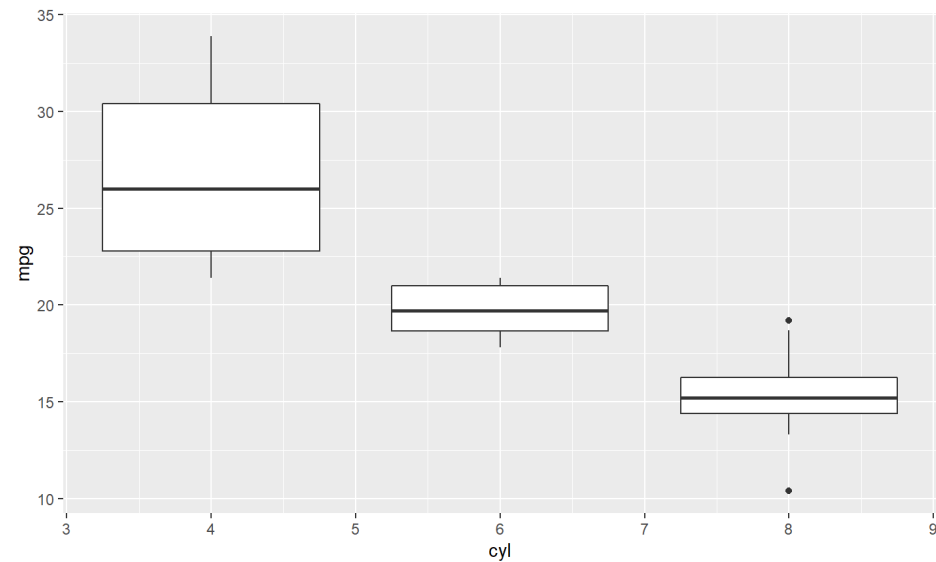
```
`r, eco=TRUE, eval=TRUE}  
ggplot(mtcars,  
       aes(x=cyl,y=mpg, group=cyl)) + geom_boxplot()  
`
```

```
> ggplot(mtcars, aes(x = cyl, y = mpg, group = cyl)) + geom_boxplot()
```



Risultati senza codice

```
`r, eco=FALSE, eval=TRUE}  
ggplot(mtcars,  
       aes(x=cyl,y=mpg, group=cyl)) + geom_boxplot()  
`
```



Codice senza risultati (non eseguito)

```
```{r, eco=TRUE, eval=FALSE}  
ggplot(mtcars,
 aes(x=cyl,y=mpg, group=cyl)) + geom_boxplot()
```
```

```
> ggplot(mtcars, aes(x = cyl, y = mpg, group = cyl)) + geom_boxplot()
```

Codice eseguito, nessun risultato, nessun codice

```
```{r, include=FALSE}  
library(ggplot2)
```
```

Non appare niente, non sono prodotti risultati, ma è quello che serve come base del file per funzionare.

Messaggi di errore, warning, messaggi

- Molti pacchetti di R mostrano dei messaggi quando vengono caricati o quando viene usata qualche funzione. Per toglierli: `message=FALSE`
- Per togliere i warning (ad esempio quando usiamo `lme4` e fittiamo modelli molto complessi): `message=FALSE`
- Per togliere i messaggi di errore (e.g., viene chiamato un oggetto che non esiste): `error=FALSE`

```
` `{r, message=FALSE, error=FALSE, warning=FALSE}  
ggplot(mtcars, aes(x=cyl,y=mpg, group=cyl)) + geom_boxplot()  
` `
```

⚠ i messaggi di errore/warning sono spariti, gli errori che li causano esistono ancora!!!

Risultati un po' più carini

Di default, i risultati vengono mostrati così:

```
##      speed dist
## 1         4    2
## 2         4   10
## 3         7    4
```

Gli asterischi si possono togliere con l'argomento `comment`, settato uguale a quello che più vi piace:

```
```{r, comment=" "}  
cars[1:3,]
```
```

```
      speed dist  
1         4    2  
2         4   10  
3         7    4
```

`eval=FALSE` **VS.** `results='hide'`

`eval=FALSE`

- Il codice **non** viene eseguito
- Se ci sono degli errori, non ve ne accorgete
- Particolarmente utile quando si vuole spiegare un codice molto complesso (ma dovete essere sicuri che funzioni)

`results='hide'`

- Il codice **viene** eseguito
- Se ci sono degli errori il codice non va
- Particolarmente utile quando si vuole spiegare un codice più semplice

Your turn!

- Create un nuovo chunk per il summary dei dati `summary(dati)` **senza** codice e con “NA” al posto degli hashtag nell’output
- Nuovo chunk dove eseguite il codice del vostro dataset che trovate [qui](#) (non fate la regressione e il grafico) **ma non mostrate né il codice né i risultati**
- Nuovo chunk dove eseguite il codice del grafico `plot(data$y ~ data$x)` **senza** codice
- Nuovo chunk dove mostrate il codice del grafico `plot(data$y ~ data$x)` **senza** risultati

Le tabelle

La sintassi classica (Don't panic)

```
Colonna 1	Colonna 2	Colonna 3
1	2	3
4	5	6
```

Che mi dà:

| Colonna 1 | Colonna 2 | Colonna 3 |
|-----------|-----------|-----------|
| 1 | 2 | 3 |
| 4 | 5 | 6 |

Ci sono dei compilatori automatici che ci risparmiano la fatica, ma non è comunque il massimo

Stessa sintassi in `html` e `PDF`

Integrare con R!

Ci sono diversi pacchetti che ci permettono di creare tabelle direttamente dai dataset senza fare fatica:

- [xtable](#)
- [stargazer](#) ❤️
- [sjPlot](#) (per le tabelle di correlazione non c'è niente di meglio)
- [kable e kableExtra](#)

In pratica, noi facciamo le nostre analisi e il pacchetto scelto le sistema in tabelle pronte all'uso!

Tabelle senza fatica

Semplicemente l'output di R:

```
> (summary_cars = mtcars %>%  
+   group_by(cyl) %>%  
+   summarise(mean = mean(mpg), sd = sd(mpg), min = min(mpg), max = max(mpg)))
```

```
## # A tibble: 3 x 5  
##   cyl mean    sd  min  max  
##   <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1     4  26.7  4.51  21.4  33.9  
## 2     6  19.7  1.45  17.8  21.4  
## 3     8  15.1  2.56  10.4  19.2
```

😁 rapido, efficace, colpo d'occhio

😓 Nella tesi non sta molto bene

stargazer

- 😊 Fa in automatico i summary dei dataset
- 😊 Sistema in tabella i risultati dei modelli
- 😊 Stessa sintassi per html e PDF
- 😊 Perfetto per model comparison
- 😿 Non va bene per riportare le tabelle di summary preparate da noi.
- 😿 è ottimizzato per LaTeX (ossia per produrre i PDF).

FONDAMENTALE:

```
```{r, results='asis'}  

```
```

stargazer summary

```
`r, results='asis')
library(stargazer)
stargazer(mtcars, type="latex", summary = TRUE,
          title= "Tabella di summary", digits = 2, header=FALSE)
`
```

```
##
## \begin{table}[!htbp] \centering
##   \caption{Tabella di summary}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lcccccc}
## \hline
## \hline \hline
## Statistic & \multicolumn{1}{c}{N} & \multicolumn{1}{c}{Mean} & \multicolumn{1}{c}{St.
## \hline \hline
## mpg & 32 & 20.09 & 6.03 & 10 & 15.4 & 22.8 & 34 \hline
## \hline \hline
```

stargazer model

```
> m1 = lm(mpg ~ cyl, data = mtcars)
> stargazer(m1, type = "html", summary = TRUE, title = "Risultati del modello", digits = 2,
+   header = FALSE)
```

| Risultati del modello | |
|-------------------------|-------------------------------------|
| | <i>Dependent variable:</i> |
| | mpg |
| cyl | -2.88***
(0.32) |
| Constant | 37.88***
(2.07) |
| Observations | 32 |
| R ² | 0.73 |
| Adjusted R ² | 0.72 |
| Residual Std. Error | 3.21 (df = 30) |
| F Statistic | 79.56*** (df = 1; 30) |
| Note: | $p < 0.1$; $p < 0.05$; $p < 0.01$ |

stargazer model comparison

```
> m0 = lm(mpg ~ 1, data = mtcars)
> m1 = lm(mpg ~ cyl, data = mtcars)
> stargazer(m0, m1, type = "html", title = "Model comparison", digits = 2, intercept.top = TRUE,
+   intercept.bottom = FALSE, header = FALSE)
```

| Model comparison | | |
|-------------------------|-------------------------------------|--------------------|
| | <i>Dependent variable:</i> | |
| | mpg | |
| | (1) | (2) |
| Constant | 20.09***
(1.07) | 37.88***
(2.07) |
| cyl | | -2.88***
(0.32) |
| Observations | 32 | 32 |
| R ² | 0.00 | 0.73 |
| Adjusted R ² | 0.00 | 0.72 |
| Residual Std. Error | 6.03 (df = 31) | 3.21 (df = 30) |
| F Statistic | 79.56*** (df = 1; 30) | |
| Note: | $p < 0.1$; $p < 0.05$; $p < 0.01$ | |

Your turn!

- Tabella di summary del vostro dataset con 3 decimali
- Tabella di summary del vostro modello di regressione
- Caption per ognuna delle tabelle

ADVANCED

- Tabella di model comparison con modello nullo m_0 ($m_0 = \text{lm}(y \sim 1, \text{data} = \text{dati})$)

Codice (e risultati) nel testo

Avete visto che nelle mie slide appaiono degli elementi scritti con la formattazione del codice di R

Ottenerli è molto semplice: ``R`` mi restituisce R

Ma molto di più: ``r mean(mtcars$mpg)`` mi restituisce 20.090625 (Provateci!)

Se in un chunk precedente avessimo assegnato `x = mean(mtcars$mpg)`, avremmo potuto semmplicemente scrivere ``r x``

Codice, equazioni e codice nelle equazioni

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n}$$

Per mpg:

$$\bar{X} = \frac{\sum_{i=1}^n x_i}{n} = \frac{642.9}{32} = 20.09$$

```
$\frac{\text{round}(\text{sum}(\text{mtcars}$mpg$), 2)}{\text{nrow}(\text{mtcars})} = \text{round}(\text{round}(\text{sum}(\text{mtcars}$mpg$), 2) / \text{nrow}(\text{mtcars}), 2)$
```