

RMarkdown: le basi

RMarkdown è integrato nativamente in RStudio, per consentire la creazione di report riproducibili, a partire dai notebook. Basato sulla sintassi di Markdown, consente, a partire da un file di testo, di produrre report in vari formati, combinando testo formattato, codice e risultati.

Vedi anche:

- [Per iniziare: RMarkdown](#)
- [Da uno script R a RMarkdown, e viceversa](#)

Un primo documento di prova

Vediamo come creare ed esportare il primo documento.

1. Creazione del file

In RStudio, il menu è File > New File > RMarkdown.

Si aprirà una finestra di dialogo che chiederà di scegliere, fra le altre cose, il formato del documento (ad esempio, HTML, PDF, Word, ecc.).

Il documento appena creato si presenterà come segue:

```
---
title: "Documento RMarkdown"
author: "Nome Cognome"
date: "Oggi"
output: html_document
---
This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When
you execute code within the notebook, the results appear beneath the
code.

Try executing this chunk by clicking the *Run* button within the
chunk or by placing your cursor inside it and pressing
*Ctrl+Shift+Enter*.

```{r}
plot(cars)
```
```

Il testo integrato fornisce le istruzioni (in inglese) per iniziare a usare questi documenti.

2. Modifica del testo e salvataggio

Possiamo iniziare a scrivere, ad esempio:

```
Questo è un testo normale, che uso per commentare  
il blocco di codice eseguibile R, che è quello che segue.
```

```
```{r}  
questo è un commento in un blocco di codice
print("Ciao!")
```
```

Il cuore del documento è ovviamente rappresentato dal codice, che va inserito in un **blocco di codice**, come quello che segue.

```
```{r}  
questo è un commento in un blocco di codice
print("Ciao!")
```
```

In RStudio, il codice contenuto nei blocchi **può essere eseguito** mentre si sta scrivendo, cliccando sulla freccia verde in alto a sinistra rispetto al blocco (*run*, simile a *play*). Questo consente di analizzare / visualizzare i dati e commentarli contemporaneamente.

Per la formattazione del testo, vedi [Guida alla sintassi Markdown](#).

Anche se sembra banale ... **Salvare il file**, altrimenti il file di output non verrà generato. Dopo il primo salvataggio, il file verrà salvato sempre, prima del rendering.

3. Rendering

Il processo che trasforma un file .Rmd in un documento formattato (HTML, PDF, Word, ecc.) si chiama *rendering*.

Cliccando su “Knit” (indicato con l'icona del gomitolo, o di un block notes per i notebook) si genererà il documento di output¹⁾.

Nella figura che segue, il rendering è quello predefinito per i notebook, ovvero l'anteprima in HTML.

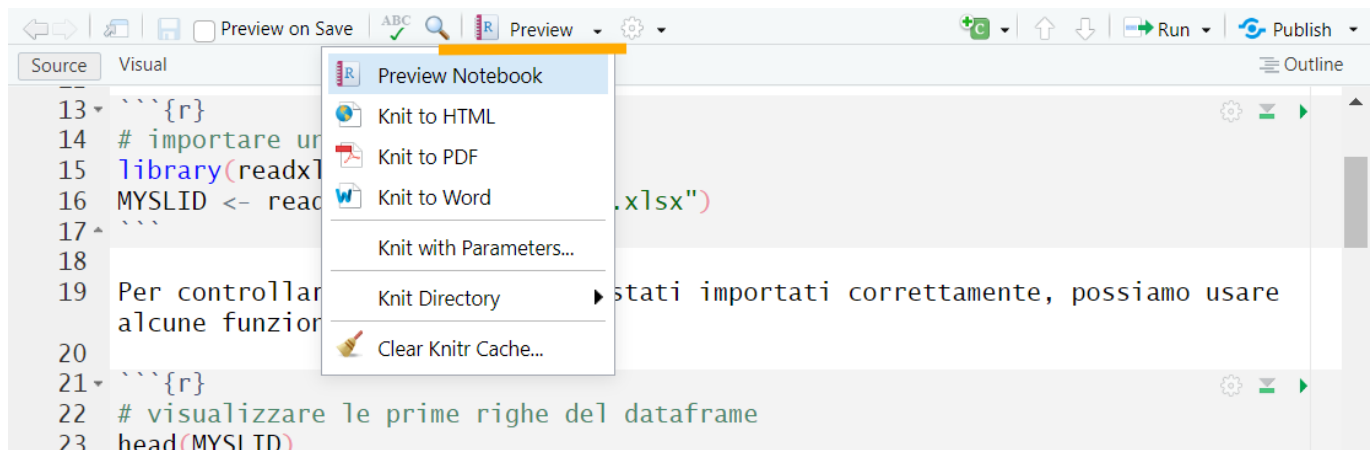


Fig. 1: Rendering del documento o del notebook

Il *rendering* può essere eseguito anche da riga di comando, o script:

```
rmarkdown::render(file, output_format = "word_document")
```

Il front matter

La parte iniziale del file .Rmd, delimitata da ---, si chiama **front matter**:

```
---
title: "Documento RMarkdown"
author: "Il tuo nome"
date: "Oggi"
output: html_document
---
```

Il front matter è un elemento di fondamentale importanza in tutti i file markdown, in quanto contiene i metadati, quali il titolo del documento, l'autore, la data e soprattutto il tipo di output desiderato.

Alcune di queste informazioni, a seconda del formato di output scelto, verranno inserite nel documento: ad esempio, il titolo indicato nel front matter sarà il titolo principale del documento finale generato.

Il tipo di output viene scelto al momento della creazione del file, e il *rendering* avverrà automaticamente per quel formato. È possibile però cambiarlo o aggiungerne un altro (dal menu a tendina, o a mano), e quindi produrre da uno stesso file più tipi di documenti.

```
---
title: "Documento RMarkdown"
author: "Il tuo nome"
date: "Oggi"
output:
  word_document: default
```

```
html_notebook: default
```

I formati accettati sono svariati (anche se non sono inclusi esplicitamente nel menu): oltre a “word_document”, “pdf_document”, “html_document”, possiamo indicare “odt_document”, “powerpoint_presentation”, e altri ancora (vedi [I formati di RMarkdown](#)).

I blocchi di codice

I blocchi di codice (*code chunks*) sono la componente fondamentale di un documento RMarkdown, in quanto permettono di inserire ed eseguire codice R (o anche altri linguaggi, ma R è quello predefinito).

In generale i blocchi di codice sono delimitati da:

- Tre *backtick* `` `
- e tre *backtick* alla fine: `` `

I blocchi di codice eseguibili

Quelli eseguibili contengono {r} all'inizio:

```
`` `{r}
# commento
codice
`` `
```

Fortunatamente, in RStudio esiste un pulsante per inserirli senza doverli scrivere, ovvero quello evidenziato nell'immagine che segue.

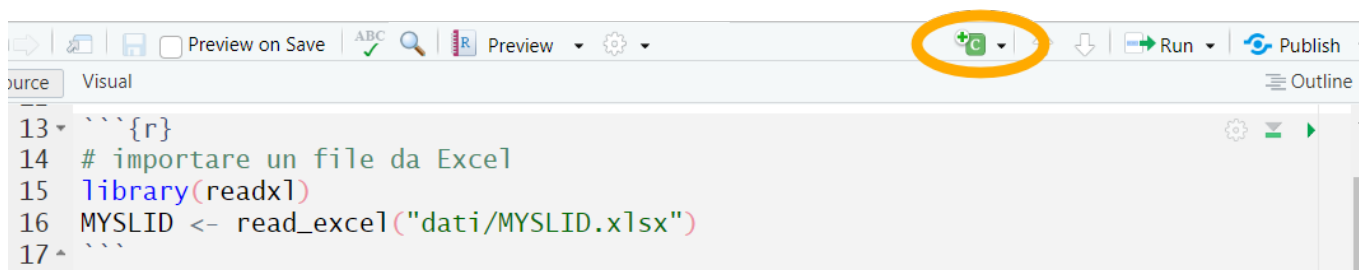


Fig. 2: Il menu per inserire i blocchi di codice

Quando il documento RMarkdown viene compilato, il codice all'interno di questi blocchi viene eseguito e i risultati (come tabelle, grafici, output testuali) vengono inseriti nel documento finale, insieme al testo normale scritto al di fuori dei blocchi.

Opzioni dei blocchi

I blocchi possono essere personalizzati con **opzioni** che controllano cosa viene mostrato nel documento finale.

| Opzioni | |
|----------------------------------|---|
| echo = TRUE | mostra il codice nel documento |
| eval = FALSE | non esegue il codice |
| include = FALSE | esegue il codice ma non mostra né codice né risultati |
| message = FALSE, warning = FALSE | nasconde messaggi e avvisi |

Tab. 1: Opzioni dei blocchi di codice (1)

Queste opzioni si scrivono nella testata del blocco²⁾, ad esempio:

```
```${r grafico, echo=FALSE}
non mostra il codice, mostra solo l'output
```
```

RStudio semplifica la gestione dei blocchi con un **menu a tendina** che appare cliccando sulla rotellina in alto a destra di ogni chunk.



Fig. 3: Il menu con le opzioni per i blocchi di codice

Dati per gli esempi successivi

```
library(LabRS)
data(MYSLID)
tab1 <- table(MYSLID$Lingua)
tab3 <- table(MYSLID$Lingua, MYSLID$Genere)
```

Le tabelle

All'interno di un report di ricerca, le tabelle sono un elemento fondamentale. Markdown permette di costruire tabelle che hanno qualche limitazione (per la formattazione base, vai a [Markdown: le tabelle](#)).

Per ovviare a tali limitazioni, esistono pacchetti specializzati, che in ogni caso richiedono di conoscere la sintassi di base e il modo in cui *rmarkdown* tratta (passando da *knitr* e *Pandoc*) le tabelle, le didascalie, la numerazione delle tabelle e i riferimenti incrociati.

Tabelle dagli output di R

Questa che segue è una tabella normale, così come la vediamo nella console di R, e incollata in un file di testo.

| | Donna | Uomo |
|----------|-------|------|
| Inglese | 2999 | 2717 |
| Francese | 262 | 235 |
| Altro | 564 | 527 |

Utilizzando la funzione `kable()` del pacchetto *knitr* (comunque usato per il *rendering* dei documenti RMarkdown:

```
```{r}
kable(tab3, caption = "didascalia")
```
```

la tabella verrà prima trasformata in markdown (in background, a meno che non eseguiate il comando in uno script o in console):

```
|          | Donna | Uomo |
|-----:|-----:|
| Inglese | 2999 | 2717 |
| Francese | 262 | 235 |
| Altro   | 564 | 527 |

: didascalia
```

e poi trasformata (con Pandoc, e sempre in background) in una tabella compatibile con il formato scelto.

I grafici

Per la formattazione dei grafici nel file di Word, possono essere utili le seguenti opzioni dei blocchi di codice:

| Opzioni | |
|-----------------|--|
| fig.cap = "..." | Didascalia |
| dpi | Risoluzione del grafico |
| out.width | Spazio della pagina occupato dal grafico |
| fig.width | Dimensioni del grafico |
| fig.height | |

Tab. 2: Opzioni dei blocchi di codice (2)

Ad esempio:

```
```${r fig.cap="Esempio di grafico a barre"}
barplot(
 tab1,
 main = "Lingue"
```
```

Per visualizzare solo il grafico senza il codice:

```
```${r echo=FALSE, fig.cap="Esempio di grafico a barre"}
barplot(
 tab1,
 main = "Lingue"
```
```

Un file di esempio

Questo file si può scaricare, ed usare per generare un file html (notebook) o un file Word.

[primo_documento.Rmd](#)

```
---
title: "Primo documento RMarkdown"
author: "Nome Cognome"
date: "`r Sys.Date()`"
output:
  word_document: default
  html_notebook: default
```

I blocchi possono avere un nome.

```
```{r pacchetti}
library(LabRS)
library(knitr)
```
```

```
```{r dati}
data(MYSLID)
tab1 <- table(MYSLID$Lingua)
tab3 <- table(MYSLID$Lingua, MYSLID$Genere)
```
```

Questa è una tabella con didascalia

```
```{r}
kable(tab3, caption = "didascalia")
```
```

Questo è un grafico con didascalia, ma nel documento non si vedrà il codice

```
```{r echo=FALSE, fig.cap="Esempio di grafico a barre"}
barplot(
 tab1,
 main = "Lingue"
)
```
```

[markdown, RStudio](#)

¹⁾
Nel caso dei notebook, viene “restituito” di default solo l’output del codice eseguito esplicitamente.

²⁾
La sintassi di RMarkdown e Quarto è qui diversa, ma Quarto interpreta correttamente i blocchi di Rmd

From:

<https://www.agnesevardanega.eu/wiki/> - **Ricerca Sociale con R**

Permanent link:

https://www.agnesevardanega.eu/wiki/r/scrittura_accademica/rmarkdown-basi

Last update: **20/10/2025 13:58**

