



Da quali parti è composto uno script .Rmd?

Un file Rmarkdown(.Rmd) è un file di testo formato dai seguenti elementi:

YAML metadata - YAML è un linguaggio leggibile in chiaro utilizzato per la serializzazione di dati che viene spesso impiegato per la scrittura dei file di configurazione. Nel file .Rmd esiste sempre all'inizio dello script una sezione in formato YAML delimitata da tre trattini (---) che serve a definire gli elementi di configurazione e i metadati quali ad esempio il titolo e il formato di output.

Plain Text - Possono essere inseriti nello script parti "narrative" formattate in markdown o in vari linguaggi di markup (es. html o latex) rispettando alcune regole di sintassi.

Code Chunks - "blocchi di codice" nei diversi linguaggi di programmazione supportati che devono essere tradotti da knitr. Sono racchiusi all'interno di blocchi con una marcatura particolare con la quale possono essere definite anche le opzioni e le modalità di esecuzione del "chunk".

YAML header

<https://yaml.org/> - <https://github.com/vubostat/r-yaml>

%YAML 1.2

YAML: YAML Ain't Markup Language™

What It Is:

YAML is a human-friendly data serialization language for all programming languages.

YAML Resources:

YAML Specifications:

- YAML 1.2:
 - [Revision 1.2.2](#) # Oct 1, 2021 *New*
 - [Revision 1.2.1](#) # Oct 1, 2009
 - [Revision 1.2.0](#) # Jul 21, 2009
- [YAML 1.1](#)
- [YAML 1.0](#)



YAML Headers Example

The screenshot shows the RStudio interface. On the left, the code editor displays the YAML header and R Markdown content:

```
home > michele > Scrivania > prova.Rmd
1
2   title: "Esempio"
3   author: "Michele Bonsignore"
4   date: "`r Sys.Date()`"
5   output:
6     html_document:
7       toc: true
8       number_sections: true
9       toc_float: true
10
11  ## Paragrafo 1
12  - R Markdown
13  - Including plots
14
15  ## Paragrafo2
16  - Quant'è bello R Markdown
```

Below the code editor is a terminal window showing the command to run R Markdown:

```
michele@T480MICHELE:~/Scrivania$ cd /home/michele/Scrivania
michele@T480MICHELE:~/Scrivania$ Rscript -e "rmarkdown::render('prova.Rmd')"
```

On the right, the browser window displays the generated HTML document:

Esempio

Michele Bonsignore

2023-10-22

0.1 Paragrafo 1

- R Markdown
- Including plots

0.2 Paragrafo2

- Quant'è bello R Markdown



YAML argomenti più comuni

- `title` -> Titolo della pagina
- `subtitle` -> Sottotitolo
- `author` -> Specifica l'autore o gli autori (es. `author: "Mike B., John Doe"`)
- `output` -> è possibile specificare uno o più formati validi per esportare il documento (es. `html_document`, `html_notebook`, `pdf_document`, `word_document`)

Attributi associati al formato `html_document`

- `toc`, `number_sections`, `toc_float` -> argomenti e attributi della "table of contents" del documento
- `theme` -> supporto ai temi, vedi galleria <https://www.datadreaming.org/post/r-markdown-theme-gallery/>
- `code_folding` -> **code_folding: show** aggiunge un pulsante che mostra o nasconde il codice dei "chuncks". L'attributo **hide** nasconde sempre il codice.

Attributi associati al formato `word_document`

- `reference-docx` -> con output `word_document` modifica l'aspetto del documento associandovi un file di template.



Esempio output multiplo (html + docx)

GNU nano 7.2

prova.Rmd

```
---
```

```
title: "Esempio"
author: "Michele Bonsignore"
date: "`r Sys.Date()`"
output:
  html_document:
    toc: true
    number_sections: true
    toc_float: true
  word_document:
    reference_docx: word_template.docx
---
## Paragrafo 1
- R Markdown
- Including plots

## Paragrafo2
- Quant'è bello R Markdown
```



Esempio output multiplo (html + docx)

```
File Modifica Visualizza Terminale Schede Aiuto
michele@T480MICHELE:~/Scrivania$ Rscript -e "rmarkdown::render('prova.Rmd',output_format='all')"
esecuzione dello script con "output_format= 'all'

processing file: prova.Rmd

output file: prova.knit.md

/usr/bin/pandoc +RTS -K512m -RTS prova.knit.md --to html4 --from markdown+autolink_bare_uris+tex_math_single_backslash --output prova.html --lua-filter /home/michele/R/x86_64-pc-linux-gnu-library/4.3/rmarkdown/rmarkdown/lua/pagebreak.lua --lua-filter /home/michele/R/x86_64-pc-linux-gnu-library/4.3/rmarkdown/rmarkdown/lua/latex-div.lua --self-contained --variable bs3=TRUE --section-divs --table-of-contents --toc-depth 3 --variable toc_float=1 --variable toc_selectors=h1,h2,h3 --variable toc_collapsed=1 --variable toc_smooth_scroll=1 --variable toc_print=1 --template /home/michele/R/x86_64-pc-linux-gnu-library/4.3/rmarkdown/rmd/h/default.html --no-highlight --variable highlightjs=1 --number-sections --variable theme=bootstrap --mathjax --variable 'mathjax-url=https://mathjax.rstudio.com/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML' --include-in-header /tmp/RtmpderhHir/rmarkdown-str41a6390dd3e.html

Output created: prova.html  <-- output ".html"
-
processing file: prova.Rmd

output file: prova.knit.md

/usr/bin/pandoc +RTS -K512m -RTS prova.knit.md --to docx --from markdown+autolink_bare_uris+tex_math_single_backslash --output prova.docx --lua-filter /home/michele/R/x86_64-pc-linux-gnu-library/4.3/rmarkdown/rmarkdown/lua/pagebreak.lua --highlight-style tango --reference-doc word_template.docx

Output created: prova.docx  <-- output ".docx"
michele@T480MICHELE:~/Scrivania$
```

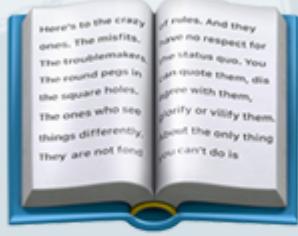


Esempio output multiplo (html + docx)

The screenshot illustrates the process of generating multiple outputs from R Markdown. On the left, the LibreOffice Writer application is open with a document titled "prova.docx". The document contains a red header "Esempio", author information "Michele Bonsignore", and a date "2023-10-22". It features two sections: "Paragrafo 1" and "Paragrafo2", each with a bulleted list. On the right, a web browser window shows the generated HTML output at the URL <https://home/michele/Scrivania/prova.html>. The HTML page has a blue header "Esempio", author "Michele Bonsignore", and date "2023-10-22". It also displays the two sections with their respective bullet points.

https://zsmith27.github.io/rmarkdown_crash-course/lesson-4-yaml-headers.html

PLAIN TEXT



FREE WORLD
WWW.FSF.ORG



Plain Text

In uno script Rmarkdown è possibile aggiungere, in alternativa a testo in formato 'Markdown', anche testo in altri formati di mark-up. Questo ci permette per esempio di aggiungere direttamente contenuti in html (es. filmati, iframe, ecc.) o formule matematiche in Latex previa opportuna marcatura, come da esempi sotto.

aggiunta di un video

```
<video width="760" controls>
<source src="img/ioslides.mp4"
type="video/mp4">
Your browser does not support
the video tag.
</video>
```

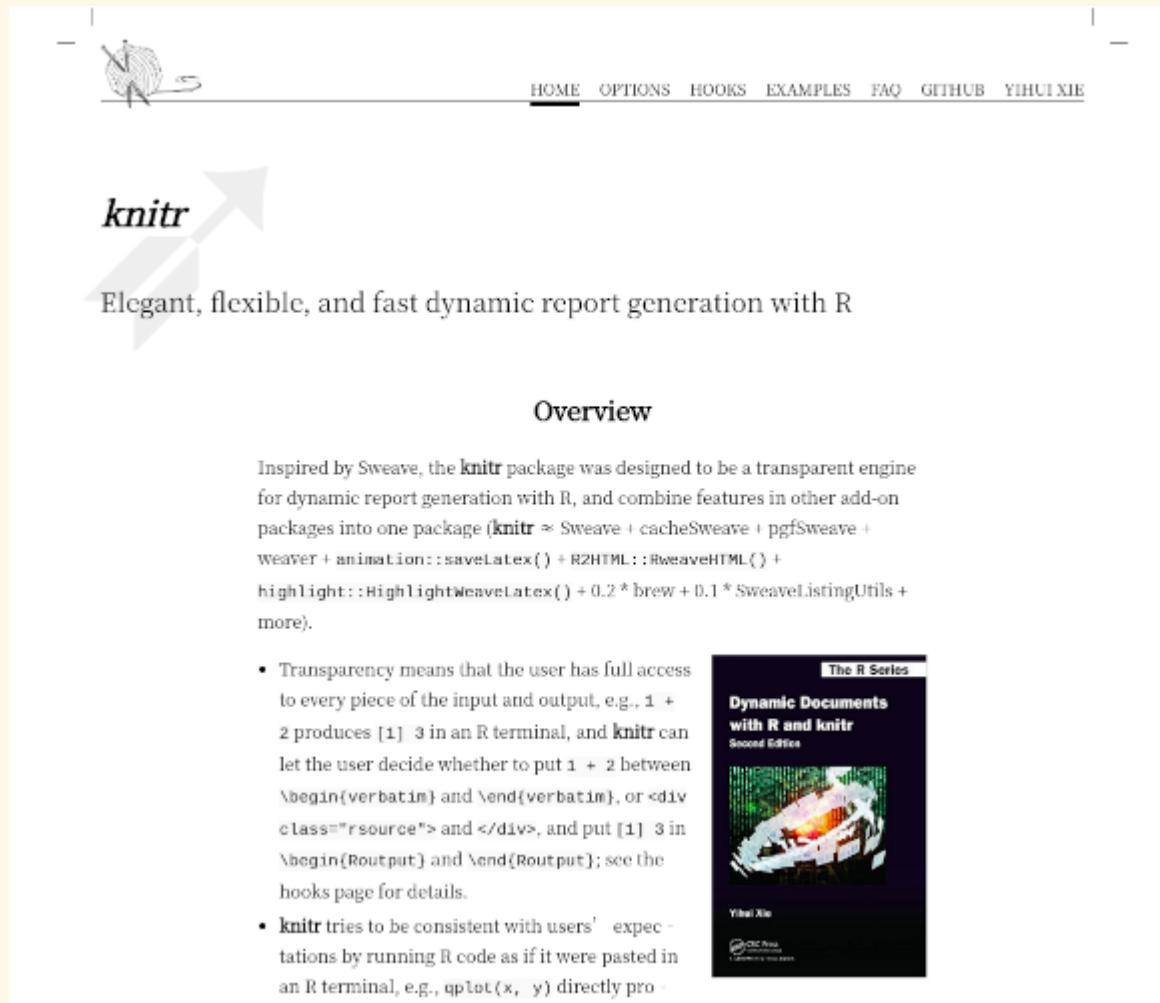
Formule matematiche LaTex

$$\Theta = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

CODE CHUNKS



FREE WORLD
WWW.FSF.ORG



The screenshot shows the homepage of the knitr package. At the top, there's a decorative header with a skein of yarn and a needle, followed by a navigation bar with links: HOME (underlined), OPTIONS, HOOKS, EXAMPLES, FAQ, GITHUB, and YIHUI XIE. Below the navigation is a large title 'knitr' in a bold, italicized font. A grey triangular graphic is positioned to the left of the title. The main text below the title reads: 'Elegant, flexible, and fast dynamic report generation with R'. Underneath this, there's a section titled 'Overview' with a short paragraph about the package's history and features. To the right of the text, there's a small image of the book 'Dynamic Documents with R and knitr' by Yihui Xie, which is part of The R Series.

Inspired by Sweave, the **knitr** package was designed to be a transparent engine for dynamic report generation with R, and combine features in other add-on packages into one package (**knitr** ≈ Sweave + cacheSweave + pgfSweave + weaver + animation:::saveLatex() + R2HTML:::RweaveHTML() + highlight:::HighlightweaveLatex() + 0.2 * brew + 0.1 * SweaveListingUtils + more).

- Transparency means that the user has full access to every piece of the input and output, e.g., `1 + 2` produces `[1] 3` in an R terminal, and **knitr** can let the user decide whether to put `1 + 2` between `\begin{verbatim}` and `\end{verbatim}`, or `<div class="rsource">` and `</div>`, and put `[1] 3` in `\begin{Rout}` and `\end{Rout}`; see the hooks page for details.
- **knitr** tries to be consistent with users' expectations by running R code as if it were pasted in an R terminal, e.g., `plot(x, y)` directly pro-

<https://yihui.org/knitr/>



knitr can execute code in many language engines

```
```{r, label='my-chunk', engine='xx'}
#il tuo codice qui
````
```

Linguaggi supportati

```
[1] "awk" "bash" "coffee" "gawk" "groovy" "haskell"
[7] "lein" "mysql" "node" "octave" "perl" "php"
[13] "psql" "Rscript" "ruby" "sas" "scala" "sed"
[19] "sh" "stata" "zsh" "asis" "asy" "block"
[25] "block2" "bslib" "c" "cat" "cc" "comment"
[31] "css" "ditaa" "dot" "embed" "eviews" "exec"
[37] "fortran" "fortran95" "go" "highlight" "js" "julia"
[43] "python" "R" "Rcpp" "sass" "scss" "sql"
[49] "stan" "targets" "tikz" "verbatim" "glue" "glue_sql" [55] "gluesql"
```



<https://yihui.org/knitr/demo/engines/>

```
```{r hellopython, echo=FALSE,
 comment=", engine='python'}
Python engine test
x = 'hello python world!'
print(x.split(' '))
```
```

['hello', 'python', 'world!']

```
```{r test-c, engine='c'}
void square(double *x) {
 *x = *x * *x;
}
```

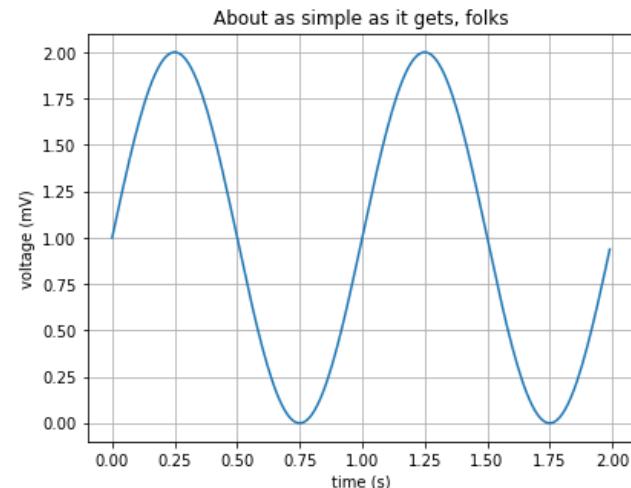
```{r, results='markup',comment=""}
.C('square', 123)
```
```

[[1]]
[1] 15129



Un esempio più complesso in python

```
```{python echo=FALSE, message=TRUE}
import matplotlib.pyplot as plt
import numpy as np
Data for plotting
t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2 * np.pi * t)
fig, ax = plt.subplots()
ax.plot(t, s)
ax.set(xlabel='time (s)', ylabel='voltage (mV',
 title='About as simple as it gets, folks')
ax.grid()
fig.savefig("test.png")
plt.show()
```
```





Un esempio con PHP

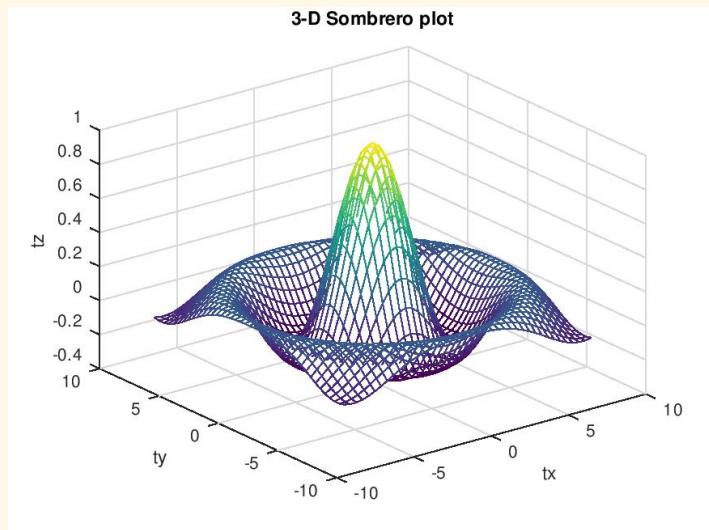
```
```{php echo=FALSE, comment=""}
$indice = 3;
do
{
 // esegue questo almeno una volta
 echo "Indice: ".$indice."\n";
 $indice--;
}
while ($indice > 0);
````
```

Indice: 3.
Indice: 2.
Indice: 1.



Octave example (syntax like MATLAB®)

```
```{octave}
graphics_toolkit('fltk')
figure (1);
clf ();
tx = ty = linspace (-8, 8, 41)';
[xx, yy] = meshgrid (tx, ty);
r = sqrt (xx .^ 2 + yy .^ 2) + eps;
tz = sin (r) ./ r;
mesh (tx, ty, tz);
xlabel ("tx");
ylabel ("ty");
zlabel ("tz");
title ("3-D Sombrero plot");
print -djpeg figure1
```
```





Un esempio banale in Perl

```
```{perl echo=FALSE, comment=""}
#!/usr/bin/perl
use strict;
use warnings;

print "hi Pearl Learner\n";
```
```

hi Pearl Learner



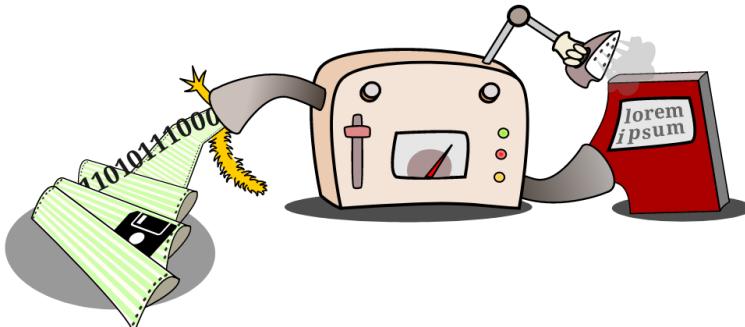
Pandoc - The Universal Converter

<https://pandoc.org/>

Pandoc a universal document converter

Donate Sponsor Please help Ukraine!

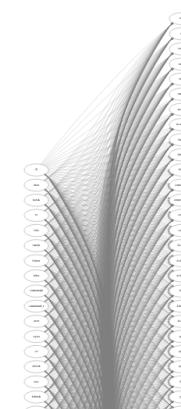
About Installing Demos ▾ Documentation Help Extras Releases Search



If you need to convert files from one markup format into another, pandoc is your swiss-army knife. Pandoc can convert between the following formats:

(← = conversion from; → = conversion to; ↔ = conversion from and to)

| | |
|---|---|
| Lightweight markup formats | Word processor formats |
| → Markdown (including CommonMark and GitHub-flavored Markdown)
↔ reStructuredText
→ AsciiDoc
↔ Emacs Org-Mode
↔ Emacs Muse
→ Textile
→ Markua
↔ txt2tags | ↔ Microsoft Word docx
↔ Rich Text Format RTF
↔ OpenOffice/LibreOffice ODT |
| HTML formats | Interactive notebook formats |
| ↔ (X)HTML 4
↔ HTML5 | ↔ Jupyter notebook (ipynb) |
| | Page layout formats |
| | ↔ InDesign ICML
↔ Typst |
| | Wiki markup formats |
| | ↔ MediaWiki MW |



<https://pandoc.org/diagram.svg?r=20230606212820>



Installazione del software

Preparazione dell'ambiente di sviluppo



Installiamo RStudio

L'ambiente di sviluppo non è in bundle con pertanto va installato a parte.

Installiamo ora l'ambiente di sviluppo **vivamente consigliato** [rstudio-desktop](#). Se siamo in ambiente Linux multiutente possiamo eventualmente valutare anche la possibilità di installare [rstudio-server](#) (l'IDE in questo caso è accessibile da browser in localhost:8787)

- In alternativa sono disponibili altri IDE dai repository della propria distribuzione come [Rkward](#) o [R Commander](#) ma il paragone con RStudio non regge.
- RStudio è multipiattaforma quindi disponibile come pacchetto per Linux [.deb](#) o [rpm](#) e per windows come [installer](#) binario.
- Esiste inoltre la possibilità per i diversi O.S. di avviare **Rstudio** da "Tarball", ovvero utilizzandolo in modo simile ad una *portable app*.
- Per gli utilizzatori occasionali, senza installare nulla, è inoltre possibile avviare da browser un'istanza gratuita Rstudio in cloud <https://posit.cloud/plans/free>



RStudio Server (Login Utente)

The screenshot shows a web browser window with the title "R Studio Server". Below the title is a "Sign in to RStudio" form. The form has two input fields: "Username" and "Password", both of which are currently empty. Below the password field is a checkbox labeled "Stay signed in when browser closes". A note below the checkbox states: "You will automatically be signed out after 60 minutes of inactivity." At the bottom of the form is a blue "Sign in" button.



RStudio Desktop

The screenshot shows the RStudio desktop interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The main workspace has three tabs open: "my-slides.Rmd", "Topic1Sezione1.Rmd", and "xaringanNewFile.vtt". The left pane contains a code editor with R Markdown code, including sections like "Installiamo RStudio" and "Pacchetti di base". The right pane shows the "Environment" tab with variables "content" and "titolo" defined. Below the environment is a "Values" table. The bottom pane features a "Presentation" tab displaying a slide titled "Installiamo RStudio" with the following content:

L'ambiente di sviluppo non è in bundle con R pertanto va installato a parte.
Installiamo ora l'ambiente di sviluppo vivamente consigliato [rstudio-desktop](#). Se siamo in ambiente Linux multitone possiamo eventualmente valutare anche la possibilità di installare [rstudio-server](#) (l'IDE in questo caso è accessibile da browser in localhost:8787)

I contenuti prodotti da [VIMELUG](#) sono integralmente rilasciati sotto licenza [Creative Commons v. 4.0 CC BY-NC-SA](#)



RStudio Cloud

All Plans [Free](#) Basic Standard Instructor Student Log In Sign Up

Cloud Free Features

| | | |
|---------------------|-------------|---|
| SHARED SPACES | 1 | i |
| PROJECTS | 25 | i |
| OUTPUTS | 25 | i |
| COMPUTE HOURS | 25 | i
included per month |
| MAX RAM | 1 GB | i |
| MAX CPU | 1 CPU | i |
| MAX EXECUTION | 1 hour | i |
| CONCURRENT PROJECTS | 3 | i |
| DATA CONNECTIONS | 1 per space | i |
| PROJECT TEMPLATES | 1 per space | i |
| ONLINE SUPPORT | No | i |
| BETA FEATURES | No | i |
| SSO | No | i |

Sign Up for Cloud Free

\$0 / forever

If you make limited, occasional use of Posit Cloud, have access to an organization account, or just want to explore Cloud to see if it's right for you, our Cloud Free plan is all you need.

[Sign Up](#)



Pacchetti di base

Per maggiori dettagli vedi il TaskView specifico del repository CRAN <https://cran.r-project.org/web/views/ReproducibleResearch.html> sezioni **HTML** e **Markdown**. Installiamo ora i pacchetti necessari: `markdown`, `rmarkdown`, `xaringan`, `xaringanExtra`, `knitr`, `tidyverse`, `tinytex`.

The screenshot shows the RStudio interface. The top menu bar has 'Tools' highlighted. A dropdown menu is open under 'Tools' with the following options: 'Install Packages...', 'Check for Package Updates...', 'Version Control', 'Shell...', 'Terminal', 'Jobs', 'Addins', 'Keyboard Shortcuts Help' (Alt+Shift+K), 'Modify Keyboard Shortcuts...', 'Show Command Palette' (Ctrl+Shift+P), 'Project Options...', and 'Global Options...'. The main workspace shows R code being run, including the installation of the 'expm' package.

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ - Go to file/function Addins
Console Terminal Jobs ~/
Type 'demo()' for some demos, 'help()' for on-line help,
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("expm")
WARNING: Rtools is required to build R packages but is not
version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/users/bhagi/Documents/R/win-
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib
Content type 'application/zip' length 239023 bytes (233
downloaded 233 KB

package 'expm' successfully unpacked and MD5 sums checked
The downloaded binary packages are in
```

Tinytext distribuzione *LaTeX* basata su *TeXLive* molto leggera e perfettamente funzionante



Pacchetti opzionali

Anche se non siamo obbligati a farlo vi consiglio di aggiungere alla vostra lista di tools anche i seguenti pacchetti, installandoli da console o da menu grafico:

```
install.packages( c("fontawesome", "emoji", "leaflet") )
```

👉 Oppure: Tutti pacchetti in un colpo solo!

Il package manager **PACMAN** ci viene in aiuto verificando se i pacchetti sono già installati e caricandoli automaticamente:

http://trinker.github.io/pacman/vignettes/Introduction_to_pacman.html

```
install.packages("pacman")
# la funzione pacman::p_load() ci permette di caricare i pacchetti e anche
# di installarli al volo se necessario
pacman::p_load(fontawesome, xaringan, xaringanExtra, knitr, tidyverse, emoji, leaflet)
```



Facciamo un po' di pratica!!!

- - - Documents

HTML document
Notebook
PDF Document
Word document
OpenDocument Text
Rich Text Format
Markdown document
Package Vignette

- - Presentations

ioslides presentation
Slidy presentation
Beamer presentation
PowerPoint presentation
reveal.js presentation
xaringan Presentation

- Other Formats

Community formats
Websites
R packages HTML documentation
Books
Journals
Interactive Tutorials
Shiny documents

<https://bookdown.org/yihui/rmarkdown/>

HTML document from Markdown



YAML Metadata

Per creare un documento HTML da Rmarkdown è necessario specificare `html_document` come output format negli YAML metadata del proprio documento:

```
--  
title: Habits  
author: John Doe  
date: March 22, 2005  
output: html_document  
--
```

È possibile aggiungere al documento la tabella dei contenuti personalizzandone aspetto e caratteristiche, come nel seguente esempio:

```
--  
title: Habits  
output:  
  html_document:  
    toc: true  
    toc_float: true  
--
```



```
--  
title: "prova"  
output: html_document  
date: "2023-07-09"  
  
## Dati Trimestrali {.tabset}  
#### Per prodotto  
(tabbed content)  
#### Per regione  
(tabbed content)
```

Tabbed Section

The screenshot shows a web browser window with the title bar 'prova.html | Open in Browser | Find'. Below the title bar, there's a header with the text 'prova' and the date '2023-07-09'. Underneath the header, the title 'Dati Trimestrali' is displayed. Below the title, there are two tabs: 'Per prodotto' and 'Per regione'. The 'Per regione' tab is highlighted with a blue border, indicating it is the active tab. Below the tabs, the placeholder text '(tabbed content)' is visible, suggesting where the content for the selected tab would be displayed.



title: "Theme and highlights"

output:

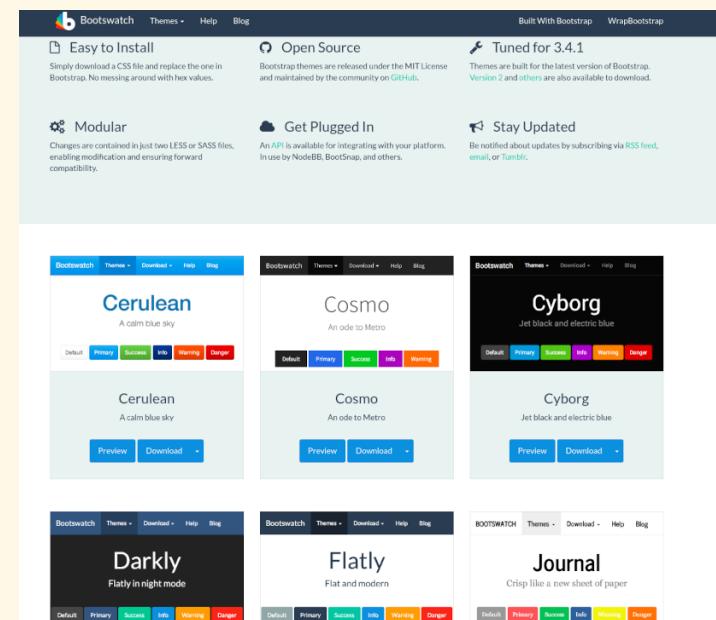
html_document:

theme: united

highlight: tango

L'opzione **theme** specifica il tipo di tema **Bootstrap** basato sulla libreria [bootswatch](#). Keyword valide: default, bootstrap, cerulean, cosmo, darkly, flatly, journal, lumen, paper, readable, sandstone, simplex, spacelab, united, yeti.

L'opzione **highlight** specifica il tipo di evidenziazione del codice (default, tango, pygments, kate, monochrome, espresso, zenburn, haddock, breezedark, textmate)





```
--  
title: "Paged Printing"  
output:  
  html_document:  
    df_print: paged  
--
```

```
```{r cars}  
mtcars
```
```

| | mpg | cyl | disp | hp | drat | wt | qsec | vs | vs | vs | vs |
|-------------------|-------|-------|-------|-------|------------|-------|-------|-------|-------|-------|-------|
| | <dbl> | <dbl> | <dbl> | <dbl> | <dbl><dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> | <dbl> |
| Mazda RX4 | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.620 | 16.46 | 0 | 0 | 0 | 0 |
| Mazda RX4 Wag | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.875 | 17.02 | 0 | 0 | 0 | 0 |
| Datsun 710 | 22.8 | 4 | 108.0 | 93 | 3.85 | 2.320 | 18.61 | 1 | 1 | 1 | 1 |
| Hornet 4 Drive | 21.4 | 6 | 258.0 | 110 | 3.08 | 3.215 | 19.44 | 1 | 1 | 1 | 1 |
| Hornet Sportabout | 18.7 | 8 | 360.0 | 175 | 3.15 | 3.440 | 17.02 | 0 | 0 | 0 | 0 |
| Valiant | 18.1 | 6 | 225.0 | 105 | 2.76 | 3.460 | 20.22 | 1 | 1 | 1 | 1 |
| Duster 360 | 14.3 | 8 | 360.0 | 245 | 3.21 | 3.570 | 15.84 | 0 | 0 | 0 | 0 |
| Merc 240D | 24.4 | 4 | 146.7 | 62 | 3.69 | 3.190 | 20.00 | 1 | 1 | 1 | 1 |
| Merc 230 | 22.8 | 4 | 140.8 | 95 | 3.92 | 3.150 | 22.90 | 1 | 1 | 1 | 1 |
| Merc 280 | 19.2 | 6 | 167.6 | 123 | 3.92 | 3.440 | 18.30 | 1 | 1 | 1 | 1 |

1-10 of 32 rows | 1-9 of 12 columns

Previous [1](#) [2](#) [3](#) [4](#) [Next](#)



Espressioni matematiche

Per produrre espressioni matematiche possiamo utilizzare la sintassi LaTex quotando il codice con il simbolo `$` o con `$$`.

```
$f(k) = {n \choose k} p^k (1-p)^{n-k}
```

$$f(k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

```
$$\Theta = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}
```

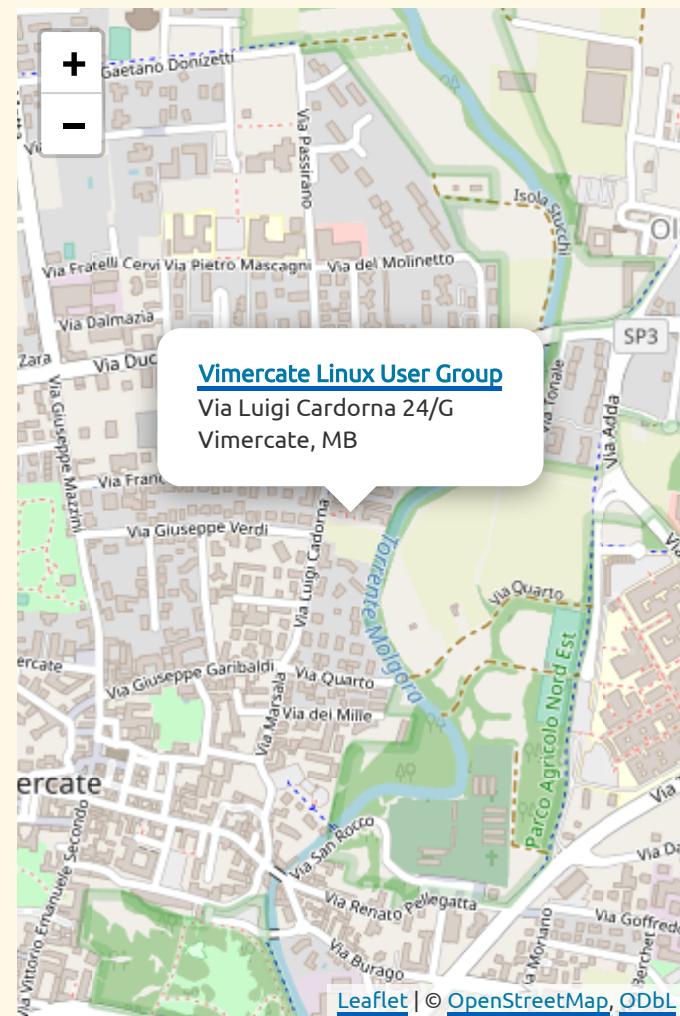
$$\Theta = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$



Html Widgets (leaflet)

```
```{r echo=FALSE, out.width="95%"}  
content <- paste(sep = "
",
 "
 Vimercate Linux User Group","
 "Via Luigi Cardona 24/G",
 "Vimercate, MB"
)
leaflet() %>% addTiles() %>%
addPopups(lat=45.61804,lng=9.37691,
 content, options =
 popupOptions(closeButton=FALSE)
)
...``
```

<https://rstudio.github.io/leaflet/>



Leaflet | © OpenStreetMap, ODbL



# FONT AWESOME ICONS & EMOJI



## Il package fontawesome

Easily and flexibly insert 'Font Awesome' icons into 'R Markdown' documents.

- Cerchiamo l'icona per keyword da console R

```
pacman::p_load(fontawesome, tidyverse) # carico i pacchetti
fa_metadata()$icon_names %>% as_tibble() %>% filter(grepl("cow",value))
```

**value**

<chr>

cow

hat-cowboy

hat-cowboy-side

3 rows

- Stampiamo le icone corrispondenti alla keyword

```
`r fa("cow", fill = "#005EB8")`
`r fa("hat-cowboy", fill = "red")`
`r fa("hat-cowboy-side", fill = "green")`
```





# Il package emoji

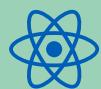
## Data and Function to Work with Emojis

```
cows <- emoji_find("cow") %>% ## cerco emoji per keyword
 rename(unicode = emoji) ## rinomino la colonna emoji del tibble in unicode
estraggo la codifica unicode emoji
cows$unicode <- sapply(cows$unicode,function(x) utf8splain::runes(x)$rune)
cows # stampo il risultato della ricerca
```

<b>name</b>	<b>unicode</b>
<b>&lt;chr&gt;</b>	<b>&lt;chr&gt;</b>
cow	U+1F42E
cow2	U+1F404
2 rows	

```
Stampo gli emoji
`r emoji('cow')`
`r emoji('cow2')`
```





# IOSLIDES PRESENTATION



## loslides in action

Vimercate Linux User Group

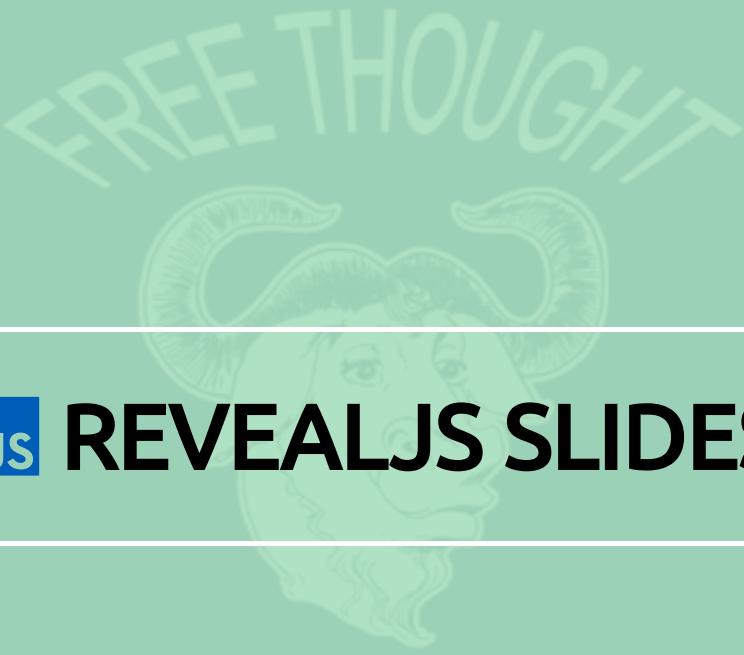
Michele Bonsignore  
July 11, 2023



File Modifica Cerca Visualizza Documento Aiuto

```
1 ---
2 title: "Vimercate Linux User Group"
3 author: Michele Bonsignore
4 date: July 11, 2023
5 output:
6 ioslides_presentation:|
7 widescreen: true
8 logo: LPI-Training.png
9 transition: slower
10 ---
11
12 # Open source & Open hardware per tutti!
13
14 ## COME ISCRIVERSI (in 3 semplici mosse)
15
16 - Versa la quota associativa al seguente: IBAN: IT97D0503434073000000001954
17 - Compila in ogni sua parte la DOMANDA DI ISCRIZIONE
18 - Invia la domanda all'indirizzo mail admt@vimelug.org
19
20 ## 5 BUONI MOTIVI PER ASSOCIARTI
21
22 - Posso parlare di tecnologia e fare esperienze
23 - Posso proporre un progetto o partecipare ad un progetto
24 - Posso sempre contare sul supporto di altri soci
25 - Posso conoscere argomenti nuovi
26 - Posso contribuire con le mie conoscenze alla crescita formativa del gruppo
27
28 ## Dove siamo
29
```

---



# **js REVEALJS SLIDES**

---



## RevealJS in action

# VIMERCATE LINUX USER GROUP

## MICHELE BONSIGNORE

JULY 11, 2023



0:00 / 0:34



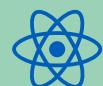


home ▶ michele ▶ Documenti ▶ Project ▶ project\_R ▶ NotebookExample ▶ revealDoc.Rmd

```
1 ---
2 title: "Vimercate Linux User Group"
3 author: Michele Bonsignore
4 date: July 11, 2023
5 output:
6 revealjs::revealjs_presentation:
7 theme: moon
8 incremental: true
9 ---
10
11 # COME ISCRIVERSI (in 3 semplici mosse)
12
13 ## PRIMA MOSSA
14 - Versa la quota associativa al seguente: IBAN: IT97D05034340730000000001954
15
16 ## SECONDA MOSSA
17 - Compila in ogni sua parte la DOMANDA DI ISCRIZIONE
18
19 ## TERZA MOSSA
20 - Invia la domanda all'indirizzo mail admt@vimelug.org
21
22 # 5 BUONI MOTIVI PER ASSOCIARTI
23
24 - Posso parlare di tecnologia e fare esperienze
25 - Posso proporre un progetto o partecipare ad un progetto
26 - Posso sempre contare sul supporto di altri soci
27 - Posso conoscere argomenti nuovi
28 - Posso contribuire con le mie conoscenze alla crescita formativa del gruppo
29
30 # DOVE SIAMO
31
32
33
```



FREE THOUGHT



# SLIDY PRESENTATION

FREE WORLD  
[WWW.FSF.ORG](http://WWW.FSF.ORG)



## Slidy in action

The screenshot shows a presentation slide with a green gradient background. A watermark of a ram's head with the words "FREE THOUGHT" above it is centered on the slide. Below the watermark, the text "Vimercate Linux User Group" is displayed in a dark blue font. At the bottom of the slide, there is a video player interface. It includes a play button, a progress bar showing "0:00 / 0:20", the name "Michele Bonsignore", the date "July 11, 2023", and three small icons for volume, full screen, and more options.



The screenshot shows the RStudio interface with the following details:

- Title Bar:** The title is "slidyPres.Rmd" located at the top center. Below it, the path is shown as "~/Documenti/Project/R/NotebookExample".
- Toolbar:** Standard RStudio icons for file operations (New, Open, Save, etc.) are visible.
- Code Area:** The main content area displays an R Markdown document. It includes YAML front matter, a section titled "## COME ISCRIVERSI (in 3 semplici mosse)", a bulleted list of steps, a section titled "## 5 BUONI MOTIVI PER ASSOCIARTI", and a bulleted list of reasons.

```

```

```
title: "Vimercate Linux User Group"
author: Michele Bonsignore
date: July 11, 2023
output:
 slidy_presentation:
 footer: "VimeLug - contenuti rilasciati sotto Creative Commons"
 css: ["w3c-blue.css", "my-style.css"]

COME ISCRIVERSI (in 3 semplici mosse)

- Versa la quota associativa al seguente: IBAN: IT97D05034340730000000001954
- Compila in ogni sua parte la DOMANDA DI ISCRIZIONE
- Invia la domanda all'indirizzo mail admτ@vimelug.org

5 BUONI MOTIVI PER ASSOCIARTI

- Posso parlare di tecnologia e fare esperienze
- Posso proporre un progetto o partecipare ad un progetto
- Posso sempre contare sul supporto di altri soci
- Posso conoscere argomenti nuovi
- Posso contribuire con le mie conoscenze alla crescita formativa del gruppo
```



# BEAMER PRESENTATION



# Beamer in action

```
--
title: "Vimercate Linux User Group"
author: Michele Bonsignore
date: July 11, 2023
output:
 beamer_presentation:
 theme: "AnnArbor"
 colortheme: "dolphin"
 fonttheme: "structurebold"
--
```

The screenshot shows a Beamer presentation slide. At the top, there is a navigation bar with icons for back, forward, search, and file operations. The slide itself has a blue header bar. The main content area contains the title "Vimercate Linux User Group" in blue, the author's name "Michele Bonsignore" in black, and the date "July 11, 2023" in black. At the bottom, there is a footer bar with three sections: "Michele Bonsignore", "Vimercate Linux User Group", and "July 11, 2023". The footer also includes a page number "1 / 5".

FREE THOUGHT

---

 Word Document

 PowerPoint Presentation

---

FREE WORLD  
WWW.FSF.ORG



## Word output

```
—
title: "Vimercate Linux User Group"
author: Michele Bonsignore
date: July 11, 2023
output: word_document
—
```

Si può modificare l'aspetto del file referenziando un template .docx nell'header in formato YAML.

```
—
title: "Vimercate Linux User Group"
author: Michele Bonsignore
date: July 11, 2023
output:
 word_document:
 reference_docx: my-styles.docx
—
```

The screenshot shows a Microsoft Word document window. At the top, there's a toolbar with icons for back, forward, search, and file operations. Below the toolbar, the page number '1 / 2' is displayed. The main content area contains the following text:

**Vimercate Linux User Group**

Michele Bonsignore

July 11, 2023

**COME ISCRIVERSI (in 3 semplici mosse)**

- Versa la quota associativa al seguente: IBAN: IT97D05034340730000000001954
- Compila in ogni sua parte la DOMANDA DI ISCRIZIONE
- Invia la domanda all'indirizzo mail [admt@vimelug.org](mailto:admt@vimelug.org)

**5 BUONI MOTIVI PER ASSOCIARTI**

- Posso parlare di tecnologia e fare esperienze
- Posso proporre un progetto o partecipare ad un progetto
- Posso sempre contare sul supporto di altri soci
- Posso conoscere argomenti nuovi
- Posso contribuire con le mie conoscenze alla crescita formativa del gruppo



## PowerPoint output

```
--
title: "Vimercate Linux User Group"
author: Michele Bonsignore
date: July 11, 2023
output: powerpoint_presentation
```

Si può modificare l'aspetto del file referenziando un template .pptx nell'header in formato YAML.

```
--
title: "Vimercate Linux User Group"
author: Michele Bonsignore
date: July 11, 2023
output:
 powerpoint_presentation:
 reference_docx: my-styles.pptx
```

The screenshot shows a Microsoft PowerPoint slide. At the top, there's a navigation bar with icons for back, forward, search, and file operations. The slide itself has a light blue header with the title "Vimercate Linux User Group". Below the title, the author's name "Michele Bonsignore" is displayed. A small date stamp "July 11, 2023" is visible at the bottom left of the slide area. The main content area contains the text "COME ISCRIVERSI (in 3 semplici mosse)" followed by a bulleted list: "• Versa la quota associativa al seguente: IBAN:".



# Interactive Tutorials - package LearnR

---



# Tutorial interattivi con il package "learnr"

```
install.packages("learnr")
```

## Tutorial Quiz Questions in learnr

[Overview](#)[Basic Question Types](#)[Custom Question Types](#)[Start Over](#)

### Formatting and Math

You can use markdown to format text within questions, answers, and custom messages. You can also include embedded LaTeX math using the `$` delimiter. For example:

```
```{r math, echo=FALSE}
x <- 42
question(sprintf("Suppose  $x = %s$ . Choose the correct statement:", x),
  answer(sprintf("$\\sqrt{x} = %d", x + 1)),
  answer(sprintf("$x^2 = %d", x^2), correct = TRUE),
  answer("$\\sin x = 1$")
)
```

```

Note the use of a double-backslash (`\\"`) as the prefix for LaTeX macros. This is necessary to “escape” the single-backslash so that R doesn’t interpret it as a special character. Here’s what this example would look like within a tutorial:

Suppose  $x = 42$ . Choose the correct statement:

- $\sqrt{x} = 43$
- $x^2 = 1764$
- $\sin x = 1$

Submit Answer



## quiz {learnr} R Documentation

# Tutorial quiz questions

## Description

Add interactive quiz questions to a tutorial. Each quiz question is executed within a **shiny runtime** to provide more flexibility in the types of questions offered. There are four default types of quiz questions:

`learnr_radio` - Radio button question. It allow for a single answer submission by the user.

`learnr_checkbox` - Check box question. It allow for one or more answers to be submitted by the user.

`learnr_text` - Text box question. It allow for free form text to be submitted by the user.

`learnr_numeric` - Numeric question. It allow for a number to be submitted by the user.

Note, the print behavior has changed as the **runtime is now Shiny based**.



## Usage

```
quiz(..., caption = rlang::missing_arg())

question(
 text,
 ...,
 type = c("auto", "single", "multiple", "learnr_radio", "learnr_checkbox",
 "learnr_text", "learnr_numeric"),
 correct = "Correct!",
 incorrect = "Incorrect",
 try_again = NULL,
 message = NULL,
 post_message = NULL,
 loading = NULL,
 submit_button = rlang::missing_arg(),
 try_again_button = rlang::missing_arg(),
 allow_retry = FALSE,
 random_answer_order = FALSE,
 options = list()
)
```



## example

```
quiz(
 question("What number is the letter A in the alphabet?",
 answer("8"),
 answer("14"),
 answer("1", correct = TRUE),
 answer("23"),
 incorrect = "See [here](https://en.wikipedia.org/wiki/English_alphabet) and try again.",
 allow_retry = TRUE
,

 question("Where are you right now? (select ALL that apply)",
 answer("Planet Earth", correct = TRUE),
 answer("Pluto"),
 answer("At a computing device", correct = TRUE),
 answer("In the Milky Way", correct = TRUE),
 incorrect = paste0("Incorrect. You're on Earth, ",
 "in the Milky Way, at a computer.")
)
)
```