



# 00-Introduzione

Ottavia M. Epifania

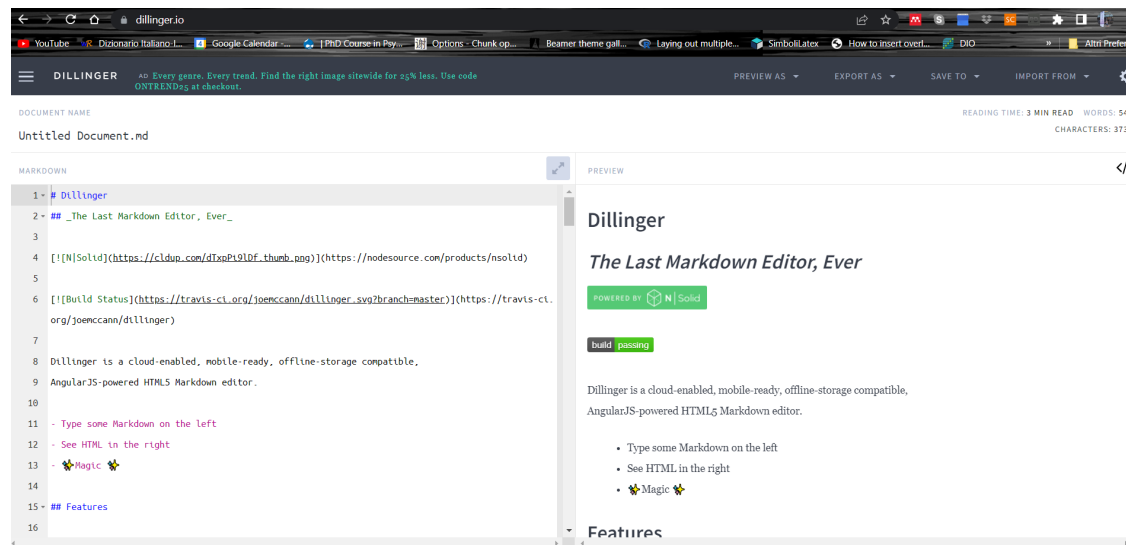
**RMarkdown**

# Di cosa stiamo parlando?

markdown è un markup language, ossia un linguaggio di programmazione dove la formattazione è gestita da pezzi di codice ("tags").

L'estensione dei markdown è un file .md

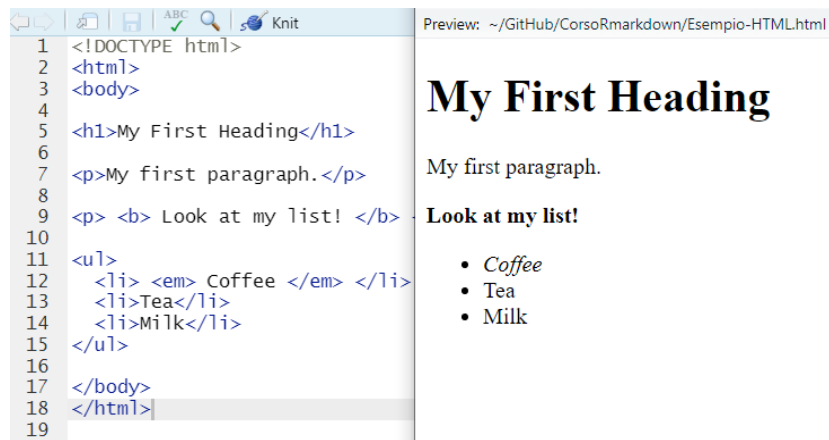
Sono disponibili diversi software per redigere documenti in .md:



<https://dillinger.io/>

# Markup languages

HTML è un classico esempio di markup language:



The screenshot shows a Knit editor window with a file named "Esemplio-HTML.html". The left pane displays the HTML source code, and the right pane shows the rendered output. The code includes a DOCTYPE declaration, html and body tags, a heading, a paragraph, and a list. The rendered output shows the heading "My First Heading", the paragraph "My first paragraph.", and a bolded list "Look at my list!" containing "Coffee", "Tea", and "Milk".

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>My First Heading</h1>
6
7 <p>My first paragraph.</p>
8
9 <p> <b> Look at my list! </b>
10
11 <ul>
12   <li> <em> Coffee </em> </li>
13   <li>Tea</li>
14   <li>Milk</li>
15 </ul>
16
17 </body>
18 </html>
19
```

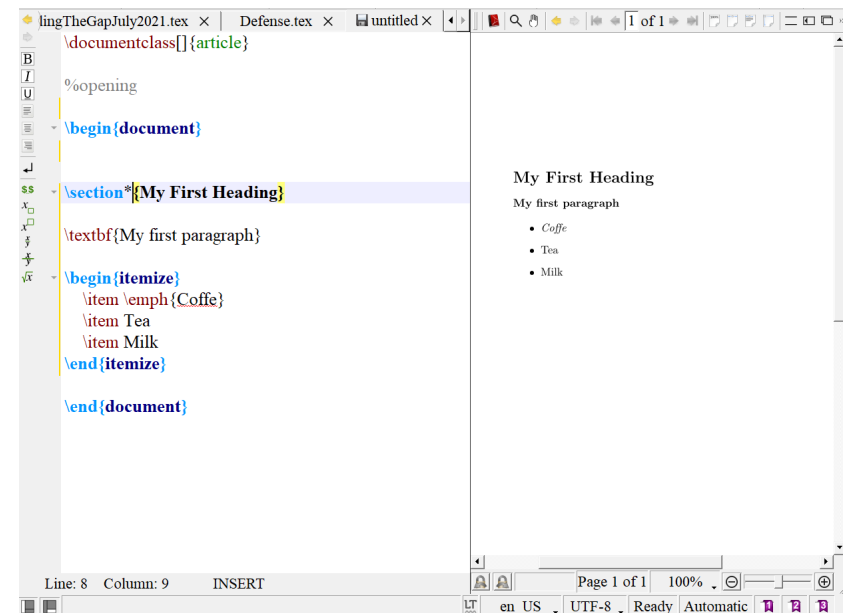
My First Heading

My first paragraph.

**Look at my list!**

- *Coffee*
- Tea
- Milk

Anche LaTeX è un markup language the best



The screenshot shows a LaTeX editor window with a file named "lingTheGapJuly2021.tex". The left pane displays the LaTeX source code, and the right pane shows the rendered output. The code includes a documentclass declaration, an opening section, a document begin/end block, a section command, a text block, an itemize list, and a document end block. The rendered output shows the heading "My First Heading", the paragraph "My first paragraph.", and a bulleted list containing "Coffee", "Tea", and "Milk".

```
\documentclass[article]{article}

%opening

\begin{document}

\section*[My First Heading]

\textbf{My first paragraph}

\begin{itemize}
\item \emph{Coffee}
\item Tea
\item Milk
\end{itemize}

\end{document}
```

My First Heading

My first paragraph

- *Coffee*
- Tea
- Milk

# WYSIWYG system

Word (o Google docs o Libre Office o Pages) è un sistema **What You See Is What You Get** → il prodotto finale è uguale a quello che si vede sullo schermo

Il testo viene direttamente modificato attraverso dei comandi built-in, con combinazioni di tasti o attraverso il menu di formattazione.

## My First Heading

My first paragraph

Look at my list!

- *Coffee*
- Tea
- Milk

# Ma allora perché...?

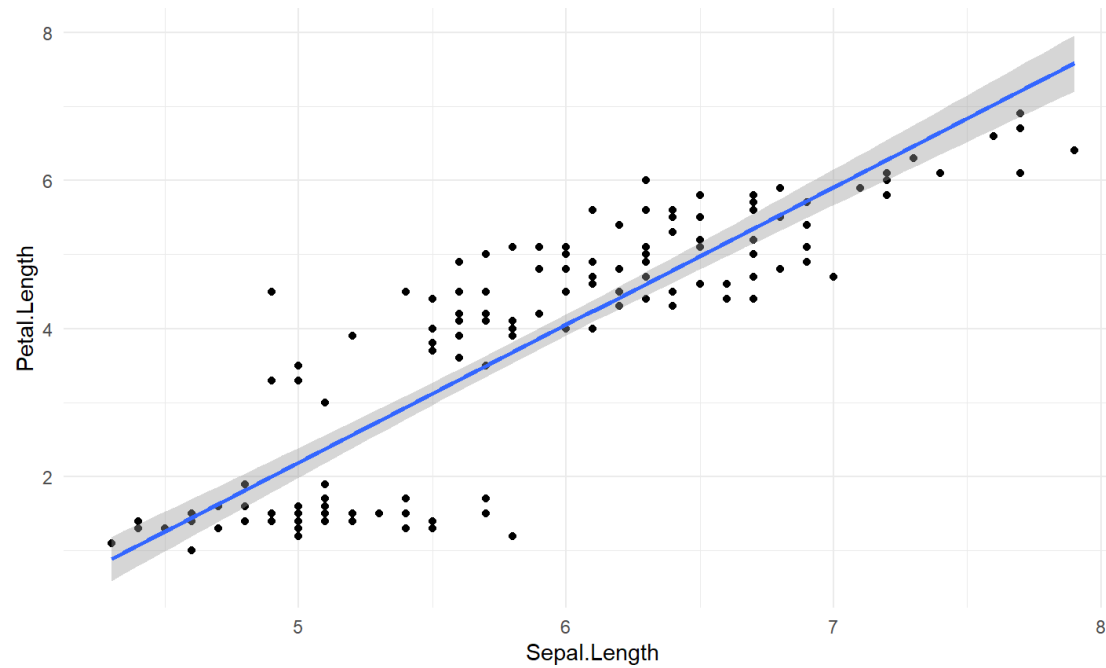


# Non solo

- RMarkdown permette di inserire **direttamente nel file di testo** i risultati delle analisi svolte, inclusi i grafici!
- Il codice usato per le analisi è dentro il file di report o la presentazione stessa: non si rischia di perdere qualche pezzo per strada
- 🎵 All you need is `knit` 🎵

# Un esempio

Questo grafico:



Ma vi viene chiesto di cambiare colore alla retta di regressione (Le cose importanti) !



# Cosa dovrete fare

In Word:

- Ripredere il codice che avete usato
- Fare le modifiche al codice
- Creare il grafico
- Salvare il grafico
- Metterlo dentro al file Word, al posto di quello vecchio

In RMarkwown all you need is knit 🎵:

- Prendere il codice usato per generare il grafico:

```
library(ggplot2)
ggplot(iris,
       aes(x=Sepal.Length, y = Petal.Length)) +
  geom_point() + theme_minimal()
geom_smooth(method="lm")
```

- Aggiungere il colore che volete  
(geom\_smooth(method="lm",  
color="red"))
- Cliccare knit

# Cosa può fare RMarkdown?

- Produrre testi in HTML e PDF (volendo anche in .docx ma è **molto** sconsigliato)
- Produrre presentazioni in HTML e PDF (volendo anche in .pptx ma è **molto** sconsigliato)
- Produrre file PDF usando il motore di LaTeX
- Le slides di questo corso

## MA SOPRATTUTTO

Permette la riproducibilità dei dati e delle analisi svolte sugli stessi tenendo in unico file report, dati, codici e risultati

# Cosa faremo in questo corso

- Baby steps: Comandi basici di `RMarkdown`
- Chunk di codice: Dalle immagini ai grafici passando per i codici
- Creazione di un documento PDF con bibliografia
- Creazione di una presentazione in HTML
- LaTeX e `RMarkdown` insieme
- Piccoli assaggi di `xaringan` e `quarto`

# Piccola esercitazione finale

Due file:

- Un PDF con almeno:
  - Un elenco, parole con diversa formattazione, una tabella, un grafico, un chunk di codice, un'equazione e risultati di  $\mathbb{R}$  nel testo
- Una presentazione in HTML con almeno 3 slide e:
  - Colonne, tabella, testo progressivo, un chunk di codice, una tabella con i risultati del modello e un plot

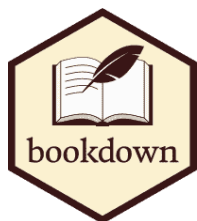
In [questa cartella](#) vi ho preparato degli esempi di file (tra cui quello per la tesi)

[Qui](#) trovate il questionario da compilare a fine corso per dire se il corso vi è piaciuto

# Cosa non faremo in questo corso



[Pagedown](#): Rende più facile il rendering in PDF



[bookdown](#): Come sopra

Let's get started 🥳

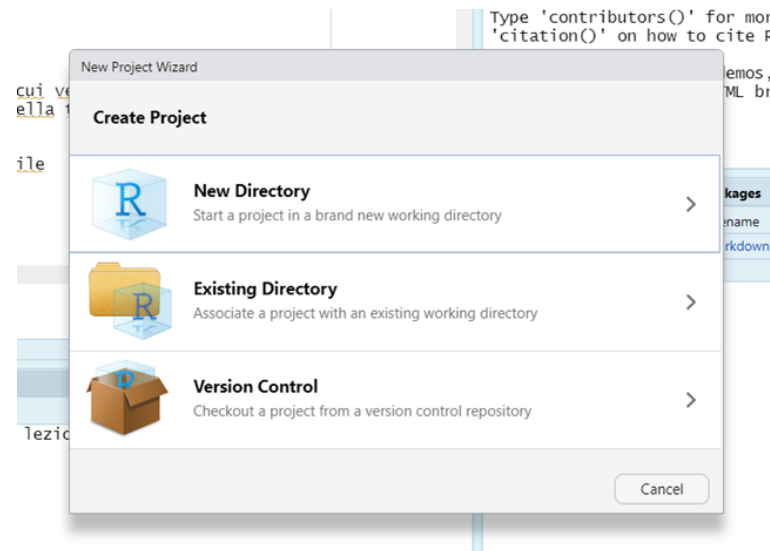
# Ordine ordine ordine ordine (e ancora ordine)

# Rstudio Project

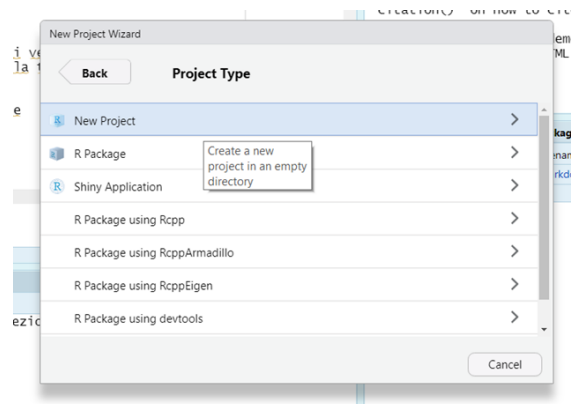
Consente di creare una directory e di organizzarla in subdirectory all'interno di cui vengono messi tutti i file (immagini, dati, bibliografia) necessari per la creazione della tesi/report/presentazione che si sta facendo

Mantiene tutto in ordine e facilmente controllabile

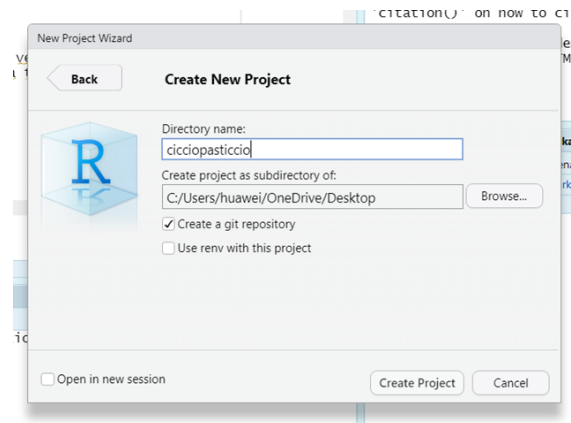
File → New project:



Dire cosa si vuole fare



Dove lo si vuole fare e come lo si vuole chiamare





Con cosa lavoriamo?

# I dataset di R

Useremo dei datasets interni ad R per fare qualche analisi e qualche grafico (tranquilli, vi darò io i codici 😁)

A [questa pagina](#) trovate un doc con la lista dei vari datasets con una piccola descrizione delle variabili

Ad ognuno viene assegnato un dataset e dovrà lavorare su quello per tutto il corso



Assegnazione dataset in corso

(Se avete dei dataset vostri su cui volete lavorare, be my guest, a patto che siate abbastanza autonomi sul codice)

# Your turn!

- Create un progetto sul vostro desktop (è il progetto su cui lavorerete per questo corso, scegliete un bel nome 😊)
- Aprite un file di R
- “Giocate” con il dataset che vi è stato assegnato (`head()`, `str()`)

## ADVANCED

- Salvate il dataset in un oggetto chiamato `data` (tip: `<-`)