

**R toolkit**

**R, RStudio, tidyverse, R Markdown**

**Matilde Trevisani**

# Toolkit e obiettivi del corso

## Toolkit

- R come linguaggio di programmazione:
  - R
  - RStudio
  - tidyverse
  - R Markdown

## Goals

- Doing Data Science:
  - ottenere **informazioni, idee, intuizioni** (*insight*) **dai dati**
  - in maniera **riproducibile**
  - usando **strumenti e tecniche moderne di programmazione**

# Riproducibilità: checklist

Cosa significa per un'analisi dei dati essere "riproducibile"?

goals di breve periodo:

- Le tabelle e le figure sono riproducibili dal codice e dai dati?
- Il codice fa effettivamente quello che tu pensi che faccia?
- Oltre a ciò che è stato fatto, è chiaro *perché* è stato fatto?

goals di lungo periodo:

- Il codice può essere usato per altri dati?
- Puoi estendere il codice per fare altre cose?

# Toolkit per la riproducibilità

- Scriptability → R
- Literate programming (codice, narrazione, output in un solo posto) → R Markdown in RStudio
- Version control → Git / GitHub

# R and RStudio

# R and RStudio



- R è un **linguaggio di programmazione** statistico open-source
- R è anche un ambiente per il calcolo e i grafici (statistici)
- R è facilmente estendibile con i *packages*



- RStudio è una comoda interfaccia per R detta **IDE** (integrated development environment), e.g. *"I write R code in the RStudio IDE"*
- RStudio non è necessario per programmare con R, ma è usato molto comunemente da programmatori di R e *data scientists*

Download di R e RStudio da <https://posit.co/download/rstudio-desktop/>

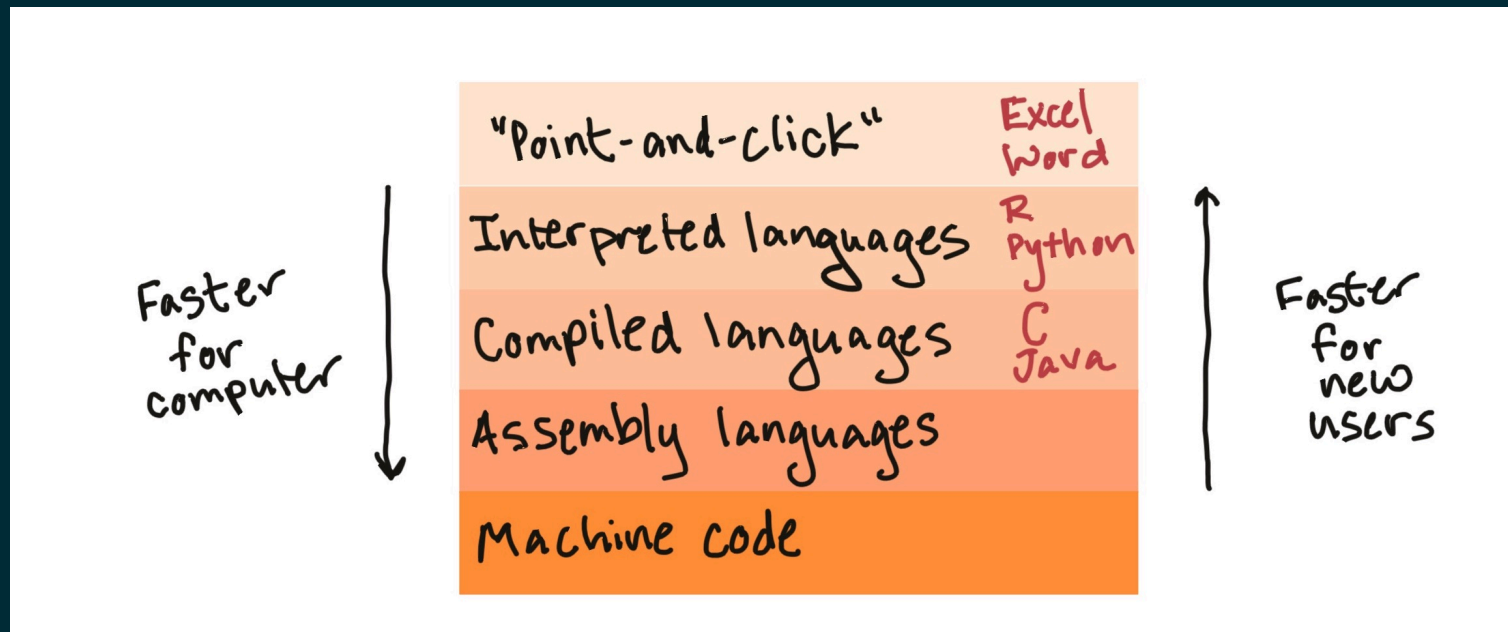
# R e altri linguaggi di programmazione

R sta diventando sempre più utile per una varietà di attività di programmazione. Tuttavia, dove brilla davvero è nel lavorare con i dati e fare analisi statistiche. R è attualmente popolare in numerosi campi, tra cui:

- Statistica
- Machine Learning
- Analisi dei dati

Altri linguaggi di programmazione popolari per il calcolo statistico includono:

- SAS
- SPSS
- Matlab
- Python



### Broad types of software programs.

- R è un **interpreted language**. Ciò significa che si può comunicare con esso in modo interattivo, da una riga di comando. Altri linguaggi interpretati sono Python e Perl.
- Programmi 'point-and-click' (come Excel e Word) sono spesso più facili per un nuovo utilizzatore all'inizio, ma più lenti per il computer e limitati nelle funzionalità offerte.
- Linguaggi compilati (come C e Java), *assembly languages*, e *machine code* sono più veloci per il computer e offrono possibilità più vaste di creazione, ma possono richiedere più tempo per il *coding* (programmazione) e a un nuovo utilizzatore per l'apprendimento.



# R versus Python

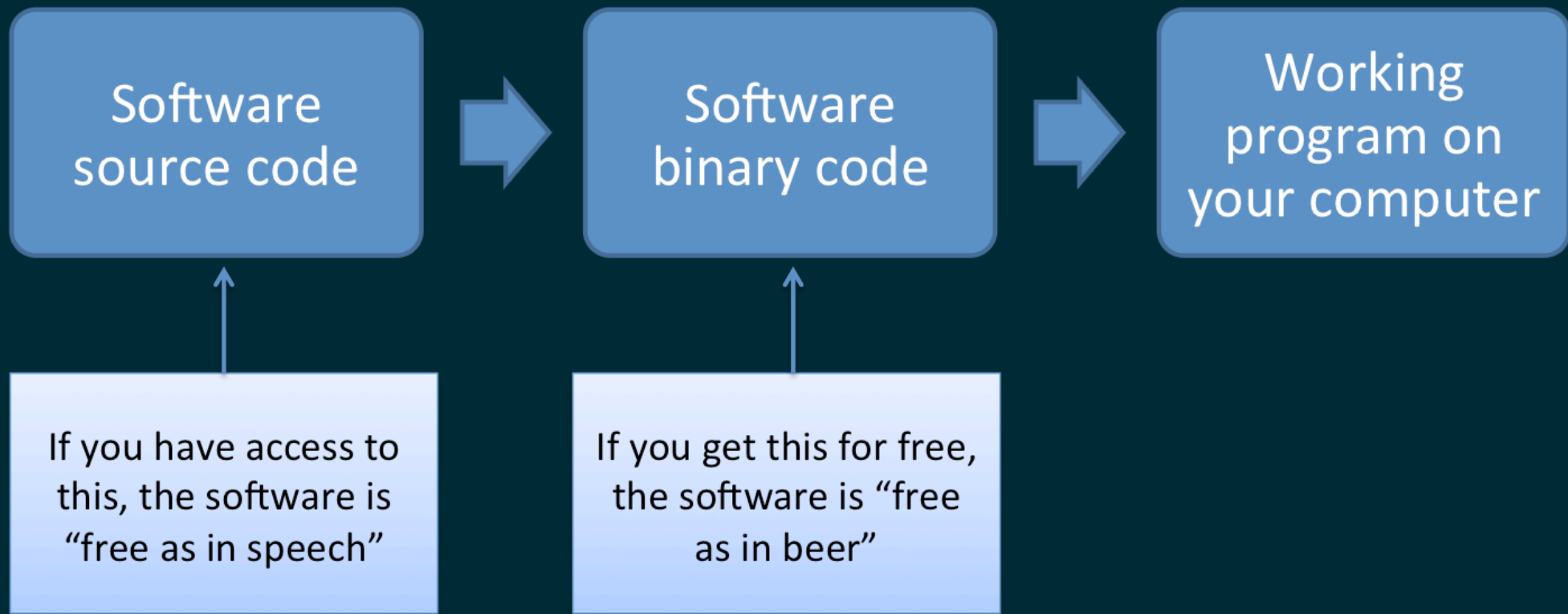
R ha alcuni degli stessi punti di forza (veloce e facile nel *coding*, si interfaccia bene con altri linguaggi, facile da lavorare in modo interattivo) e punti deboli (più lento dei linguaggi compilati) di Python.

Per le attività relative ai dati, R e Python sono relativamente testa a testa.

Tuttavia, R è ancora la prima scelta degli statistici nella maggior parte dei campi, quindi si può sostenere che R abbia un vantaggio se si desidera avere accesso a metodi statistici più avanzati.

"The best thing about R is that it was developed by statisticians. The worst thing about R is that... it was developed by statisticians."

Bo Cowgill, Google, at the Bay Area R Users Group



An overview of how software can be each type of free. For software programs developed using a compiled programming language, the final product on your computer is run by machine-readable binary code. A developer can give you this code for free without sharing any of the original source code. By contrast, open-source software is software for which you have access to the human-readable code that was used as input in creating the software binaries.

# Open source software

| "Life is too short to run proprietary software." -- Bdale Garbee

R è un software open source. Molti altri popolari linguaggi di programmazione statistica, al contrario, sono proprietari.

R è gratuito ed è forte la tentazione di pensare al software open source semplicemente come al "software gratuito".

È utile considerare alcuni significati diversi della parola "free". "Free" può significare:

- *Gratis* ("zero price"): *Free beer* come birra gratis
- *Libre* (with no or few restrictions): *Free speech* come libertà di parola



Free beer or free speech?

# *Free come libre*

Il software open source è di tipo *libre* di *free*. Ciò significa che, con il software open source, puoi:

- Accedere a tutto il codice che compone il software
- Modificare il codice come preferisci per le tue applicazioni
- Sviluppare il codice con le tue estensioni
- Condividere il software e il suo codice, nonché le tue estensioni, con altri

Significa anche che puoi creare il tuo software *on the top* di R esistente con le sue estensioni.

Questa natura open source di R (e di altri linguaggi, incluso Python) ha creato una vasta comunità di persone in tutto il mondo che sviluppano e condividono estensioni di R.

Di conseguenza, puoi contribuire con pacchetti che ti consentono di fare ogni genere di cose in R, come analizzare sondaggi complessi, adattare modelli di ML, visualizzare Tweet, e molte altre cose interessanti.

## Open source meno vulnerabile

"Despite its name, open-source software is less vulnerable to hacking than the secret, black box systems like those being used in polling places now. That's because anyone can see how open-source systems operate. Bugs can be spotted and remedied, deterring those who would attempt attacks. This makes them much more secure than closed-source models like Microsoft's, which only Microsoft employees can get into to fix." --

Woolsey and Fox. *To Protect Voting, Use Open-Source Software*. New York Times. August 3, 2017.

# Confronto con ChatGPT & co.

Tra i modelli linguistici generativi la tecnologia più famosa è ChatGPT, creata da OpenAI, basata sul LLM (Large Language Model) **GPT** (Generative Pre-trained transformer). Altri sono Claude (Anthropic), Gemini (Google), Perplexity AI, etc.

chat bot, noto anche come chatterbot o robot di conversazione, è un software progettato per simulare una conversazione con un essere umano

Tutte queste piattaforme sono proprietarie o fanno capo a grandi aziende.

Il software proprietario, noto anche come software **closed-source**, è un software di proprietà privata e limitato dal suo creatore o editore. Il codice sorgente del software proprietario non è condiviso con il pubblico, il che significa che gli utenti non possono visualizzarlo, modificarlo o distribuirlo.

In general, ChatGPT e altri chatbot sono "free" da usare. Ma se si vuole avere accesso ad un uso illimitato e agli ultimi modelli di AI si deve pagare una sottoscrizione (ad esempio, i piani a pagamento di ChatGPT sono Go, Plus, Pro, Team ed Enterprise).

## Alternative Open Source a ChatGPT

Un progetto **Open Source** è associato ad una licenza "permissiva" che consente a chiunque disponga delle necessarie competenze di studiare, modificare, adottare e redistribuire il codice sorgente di un applicativo.

Le alternative Open Source a ChatGPT vengono implementate nel contesto di community interessate al loro miglioramento e alla loro sicurezza più che a logiche di profitto.

Un altro vantaggio risiede nel fatto che a monte di tali soluzioni non vi sono grandi aziende. Questo significa che non si deve sottostare alle loro policy e a clausole contrattuali spesso di carattere puramente commerciale.

Il codice aperto garantisce inoltre maggiore trasparenza e le modalità con le quali vengono trattati i dati elaborati dai modelli sono note.

Si hanno inoltre un grado più elevato di personalizzazione e di adattamento alle singole esigenze e maggiori opportunità di risparmio in termini di investimento.

# Perchè R

- A differenza della maggior parte degli altri software progettati specificamente per l'insegnamento della statistica, R è
  - gratuito
  - open source
  - potente, flessibile e uno strumento per andare oltre i corsi introduttivi di statistica.
- Gli argomenti contro l'uso e l'insegnamento di R, soprattutto a livello statistico introduttivo, generalmente si concentrano attorno ai seguenti due punti:
  - insegnare la programmazione oltre ai concetti statistici è impegnativo
  - la riga di comando può spaventare un po' i principianti rispetto all'interfaccia utente grafica (GUI) della maggior parte dei softwares di tipo *point-and-click*.



- MA noi non vogliamo fornire sempre agli studenti tabelle di dati piccole e ordinate (per cui non ci sarebbe bisogno di software statistico oltre al foglio di calcolo o alla calcolatrice). Sarebbe un disservizio per gli studenti. Vogliamo fornire **dati grezzi** se è un corso moderno di statistica.
- Un'altra soluzione consiste nell'utilizzare i tradizionali softwares *point-and-click* per l'analisi dei dati. L'argomento tipico è che la GUI è più facile da imparare per gli studenti che quindi possono dedicare più tempo ai concetti statistici. Tuttavia, ciò ignora il fatto che questi strumenti hanno anche curve di apprendimento non banali. In effetti,
  - insegnare attività specifiche di analisi dei dati utilizzando tali software spesso richiede lunghe istruzioni passo passo, con schermate annotate, per la navigazione nei menu e in altri elementi dell'interfaccia.
  - Inoltre, non è raro che le istruzioni per un compito non si estendano facilmente ad un altro.
  - Sostituire tali istruzioni con poche righe di codice R rende effettivamente i materiali didattici più concisi e meno pesanti.

- Molti nella comunità statistica sono favorevoli all'insegnamento di R (o qualche altro linguaggio di programmazione, come Python) nei corsi di statistica di livello superiore. L'utilizzo di R nei corsi introduttivi di statistica non è così unanimamente accettato. Riconosciamo che questa aggiunta può essere onerosa, tuttavia riteniamo che apprendere uno strumento applicabile oltre il corso introduttivo di statistica e che migliori le capacità di problem solving degli studenti sia un onere che vale la pena sopportare.

# R packages

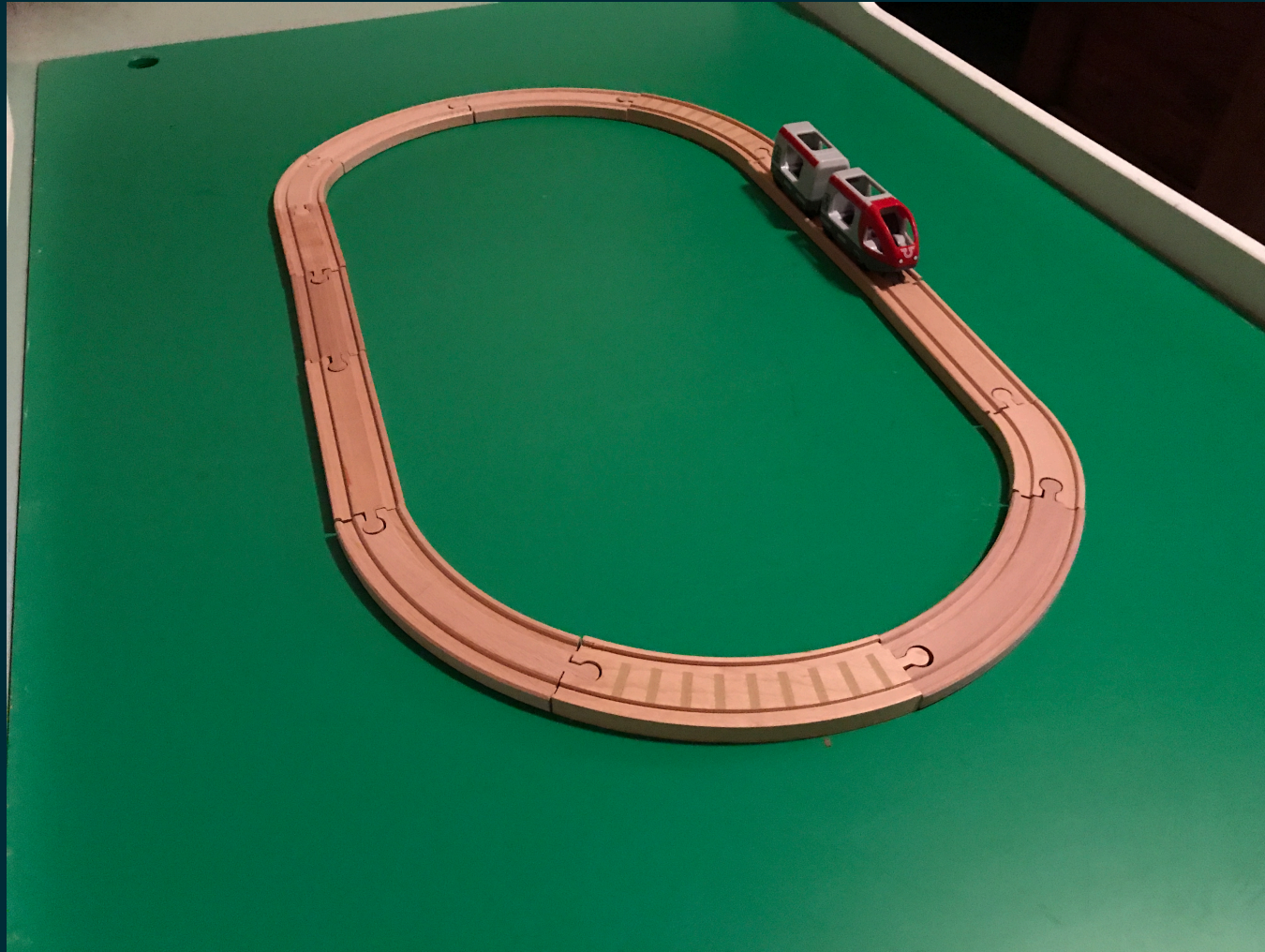
- **Packages** sono le unità fondamentali quando si produce un codice R (riproducibile). Essi contengono funzioni R riutilizzabili, la documentazione che descrive come usarle, e dati di esempio<sup>1</sup>
- A Febbraio 2026, ci sono più di 23,295 R packages disponibili in **CRAN** (the Comprehensive R Archive Network)<sup>2</sup>

<sup>1</sup> Wickham and Bryan, R Packages.

<sup>2</sup> CRAN contributed packages.

# Il sistema *package*

Il download di R è solo un punto di partenza.





Puoi espandere le funzionalità di R con i **pacchetti**



Ogni pacchetto è fondamentalmente un insieme di funzioni R extra.

Puoi ottenere questi pacchetti "aggiuntivi" in diversi modi.

- CRAN (Comprehensive R Archive Network), la fonte principale
- Bioconductor, in particolare per pacchetti relativi alla bioinformatica
- GitHub (servizio di hosting per progetti software, della società GitHub Inc.) sta diventando sempre più popolare, soprattutto per i pacchetti ancora in fase di sviluppo.
- Puoi anche creare e condividere pacchetti con i tuoi collaboratori o colleghi.

# Tour: R and RStudio

data viewer

arithmetic



load  
package

view  
data

get  
help

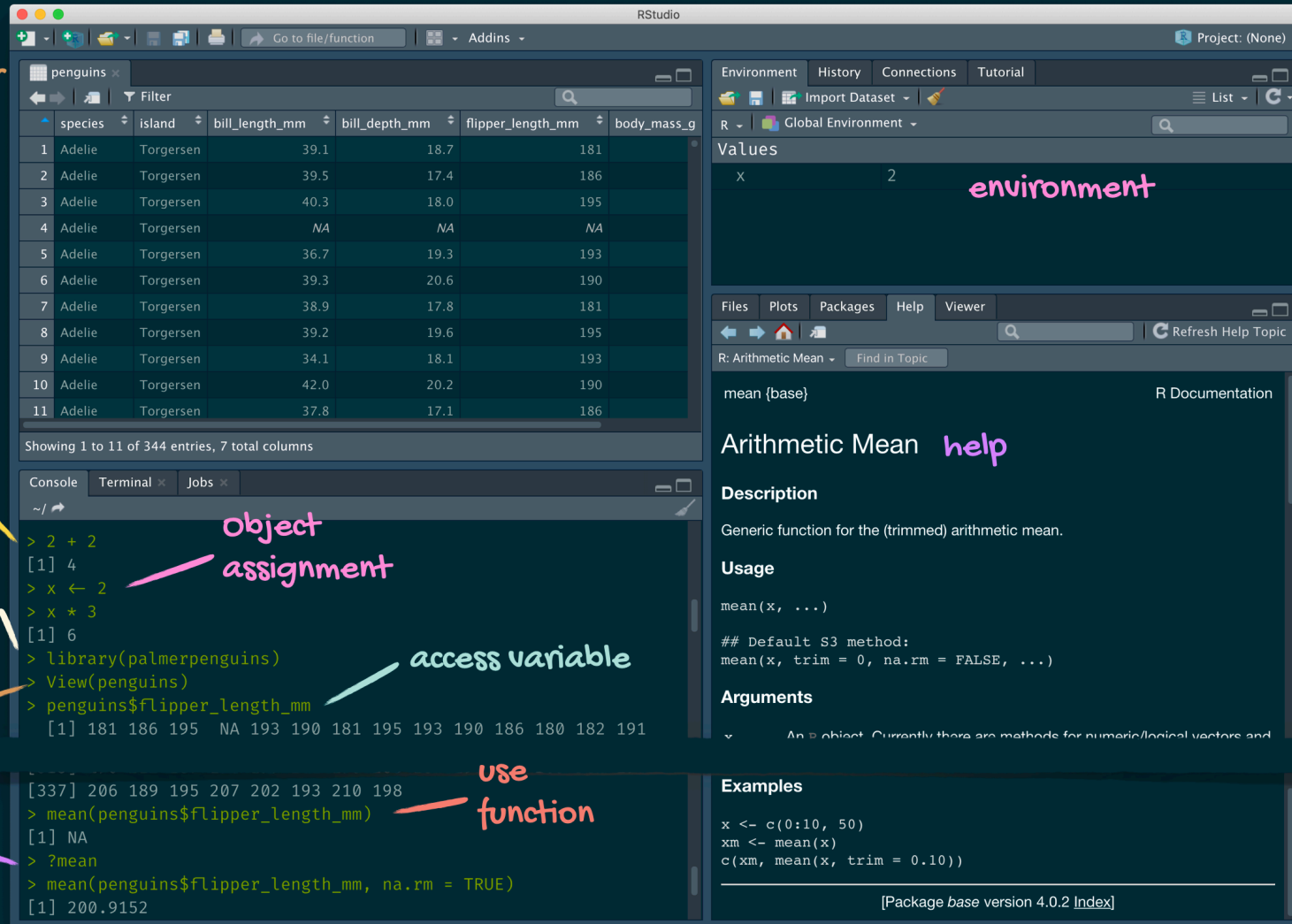
object  
assignment

access variable

use  
function

environment

help



The screenshot shows the RStudio interface with several panels and annotations:

- Environment Panel:** Shows the Global Environment with a variable `x` assigned the value `2`. An annotation "environment" points to this panel.
- Viewer Panel:** Displays the R documentation for the `mean` function. An annotation "help" points to this panel.
- Console Panel:** Contains the following R code and output:

```
> 2 + 2
[1] 4
> x <- 2
> x * 3
[1] 6
> library(palmerpenguins)
> View(penguins)
> penguins$flipper_length_mm
[1] 181 186 195 NA 193 190 181 195 193 190 186 180 182 191
[337] 206 189 195 207 202 193 210 198
> mean(penguins$flipper_length_mm)
[1] NA
> ?mean
> mean(penguins$flipper_length_mm, na.rm = TRUE)
[1] 200.9152
```

Annotations point to specific lines of code: "object assignment" points to `x <- 2`, "access variable" points to `penguins$flipper_length_mm`, and "use function" points to `mean(penguins$flipper_length_mm)`.
- Data Viewer Panel:** Displays a table of penguin data with columns: species, island, bill\_length\_mm, bill\_depth\_mm, flipper\_length\_mm, and body\_mass\_g. An annotation "data viewer" points to this panel.
- Files Panel:** Shows the project files.
- Plots Panel:** Empty.
- Packages Panel:** Shows installed packages.
- Help Panel:** Shows the R documentation for the `mean` function.

# Un brevissimo elenco (per ora) di R essentials

- Le **funzioni** sono (molto spesso) verbi, seguiti da ciò su cui saranno applicati (e come) in parentesi:

```
do_this(to_this)  
do_that(to_this, to_that, with_those)
```

- I **packages** sono installati con la funzione `install.packages` e caricati con la funzione `library`, una volta per sessione:

```
install.packages("package_name")  
library(package_name)
```



# R essentials (continued)

- La documentazione sugli oggetti o l'*Help* sono accessibili con ?

```
?mean
```

- Le colonne (variabili) in un data frame sono accessibili con \$:

```
dataframe$var_name
```

# tidyverse



[tidyverse.org](https://tidyverse.org)

- **tidyverse** è una collezione di R packages designati per l'analisi dei dati (*data science*)
- tutti i packages hanno una filosofia sottostante e una grammatica comuni

# rmarkdown

rmarkdown.rstudio.com

- il pacchetto **rmarkdown** e i vari packages correlati consentono agli utilizzatori di R di scrivere il loro codice e testo in documenti (computazionali) riproducibili
- Mostreremo un documento R Markdown (con estensione .Rmd)



# R Markdown

- Reports pienamente riproducibili -- ogni volta che tu *knit* l'analisi viene eseguita dall'inizio
- La sintassi markdown per il testo è (relativamente) semplice
- Il codice sta nei *chunks*, definito da tre *backticks*, il testo sta fuori dei chunks

# Tour: R Markdown

The screenshot shows the RStudio interface with an R Markdown document titled "Bechdel.Rmd" open in the editor. The document is divided into three main sections: a YAML header, a text block, and a code chunk. Handwritten annotations highlight key features:

- knit**: A yellow arrow points to the Knit button in the top toolbar.
- link**: A green arrow points to a URL in the text block.
- code chunk**: A pink bracket highlights a code block.
- yaml**: A red bracket highlights the YAML header.

The R Markdown document content is as follows:

```
1 ---
2 title: "Bechdel"
3 author: "Mine Çetinkaya-Rundel"
4 output:
5   html_document:
6     fig_height: 4
7     fig_width: 9
8 ---
9
10 In this mini analysis we work with the data used
    in the FiveThirtyEight story titled ["The
    Dollar-And-Cents Case Against Hollywood's
    Exclusion of Women"](https://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-exclusion-of-women/). Your task is to fill in
    the blanks denoted by `___`.
11
12 ## Data and packages
13
14 We start with loading the packages we'll use.
15
16 ```{r load-packages, message=FALSE}
17 library(fivethirtyeight)
18 library(tidyverse)
19 ```
```

The rendered HTML output on the right shows the document's appearance:

## Bechdel

Mine Çetinkaya-Rundel

In this mini analysis we work with the data used in the FiveThirtyEight story titled ["The Dollar-And-Cents Case Against Hollywood's Exclusion of Women"](https://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-exclusion-of-women/). Your task is to fill in the blanks denoted by `___`.

## Data and packages

We start with loading the packages we'll use.

```
library(fivethirtyeight)
library(tidyverse)
```

The dataset contains information on 1794 movies released between 1970 and 2013. However we'll focus our analysis on movies released between 1990 and 2013.

```
bechdel190_13 <- bechdel %>%
  filter(between(year, 1990, 2013))
```

There are `___` such movies.

The financial variables we'll focus on are the following:

- `budget_2013` : Budget in 2013 inflation adjusted dollars
- `domgross_2013` : Domestic gross (US) in 2013 inflation adjusted dollars
- `intgross_2013` : Total International (i.e., worldwide) gross in 2013 inflation adjusted dollars