UNIVERSITY OF
CAMBRIDGE
Bioinformatics Training

Introduction to

for Biologists

Alison Horst Artwork!

Gita Yadav

Introductions

Tutors

Course participants introductions

This course

- Aimed for **beginners** who never used R before.

Day 1	Day 2
Getting started	Data manipulation and visualisation with tidyverse
Introduction to R	
Starting with data	

Course Materials and Links

Online RStudio access link

TO COVER THIS MORNING

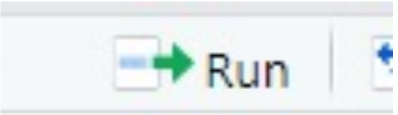
- RSTUDIO IDE
- CREATING PROJECTS & SCRIPTS
- VARIABLES
- FUNCTIONS
- DATA TYPES / DATA STRUCTURES
- VECTORS
- INDEXING

Why learn R?

- Run an R script with **one click/command.**

This Photo by Unknown Author is licensed under CC BY-NC

vs



Rscript <name of file>

Why learn R?

- R promotes **Reproducibility**

Reproducibility is when someone else (including your future self) can obtain the same results from the same dataset when using the same analysis.

  - Automating your analysis.
  - Generate reports.

Why learn R?

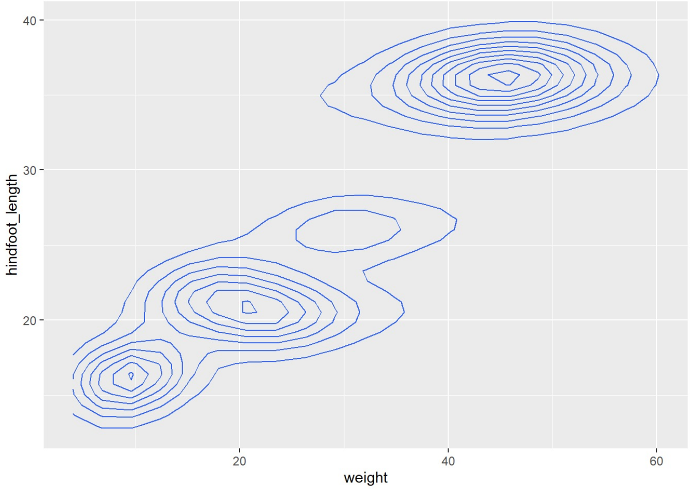
- R is **interdisciplinary**
  - Over 10,000 packages that cover different fields
  - Bioconductor repository for bioinformatics packages

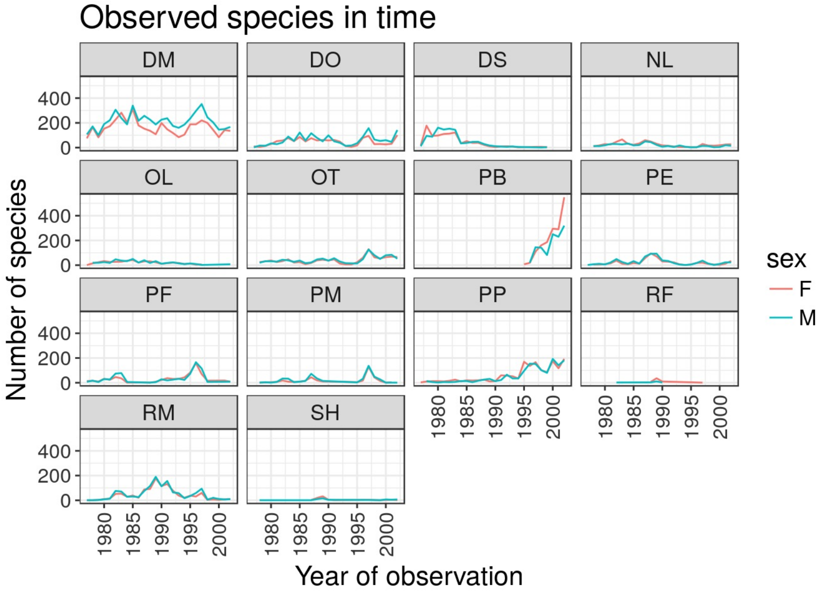
Why learn R?

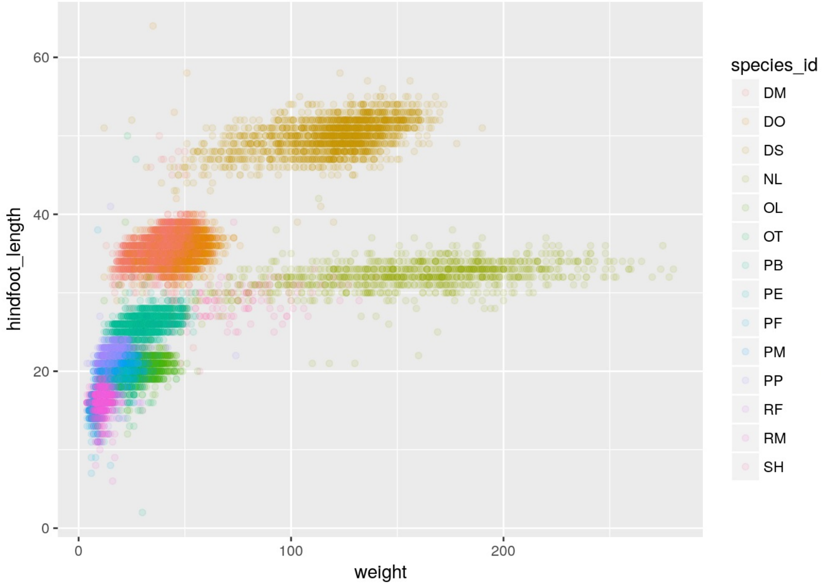
- R works on data of different sizes

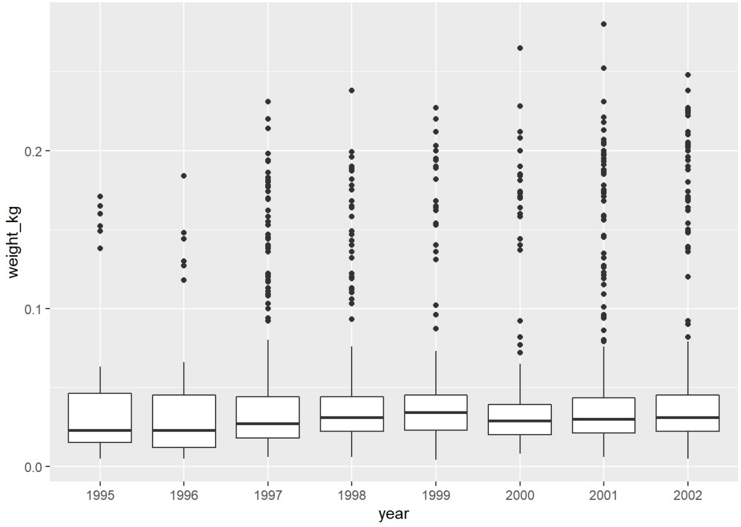
Why learn R?

- R produces high-quality graphics.





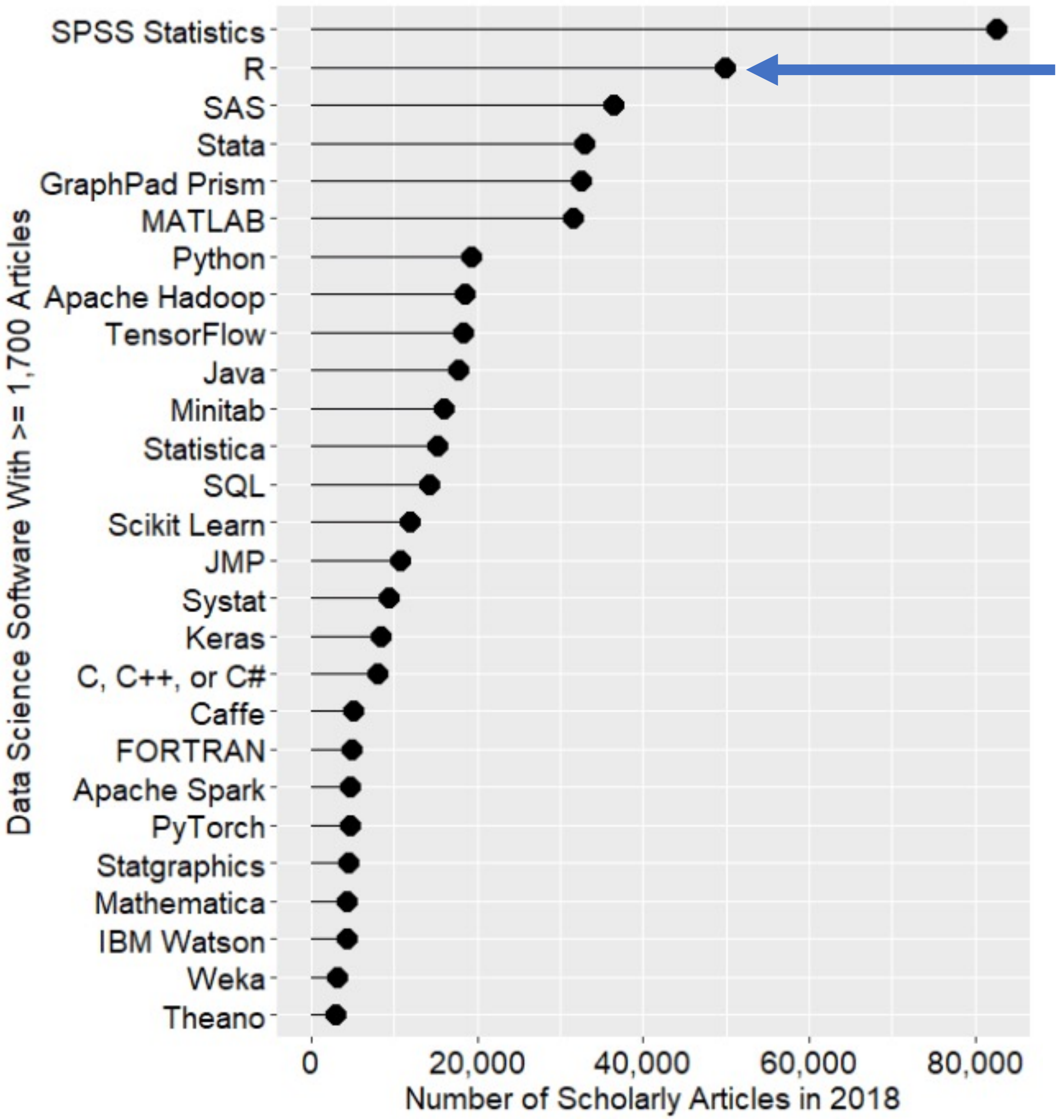




Why learn R?

- R has a large and welcoming community
  - RStudio community
  - Local R User Groups (meetup)
  - R-Ladies
  - R-bloggers
  - The Carpentries
  - Stack Overflow
- Popular in Data Science

R tops other programming languages in academia



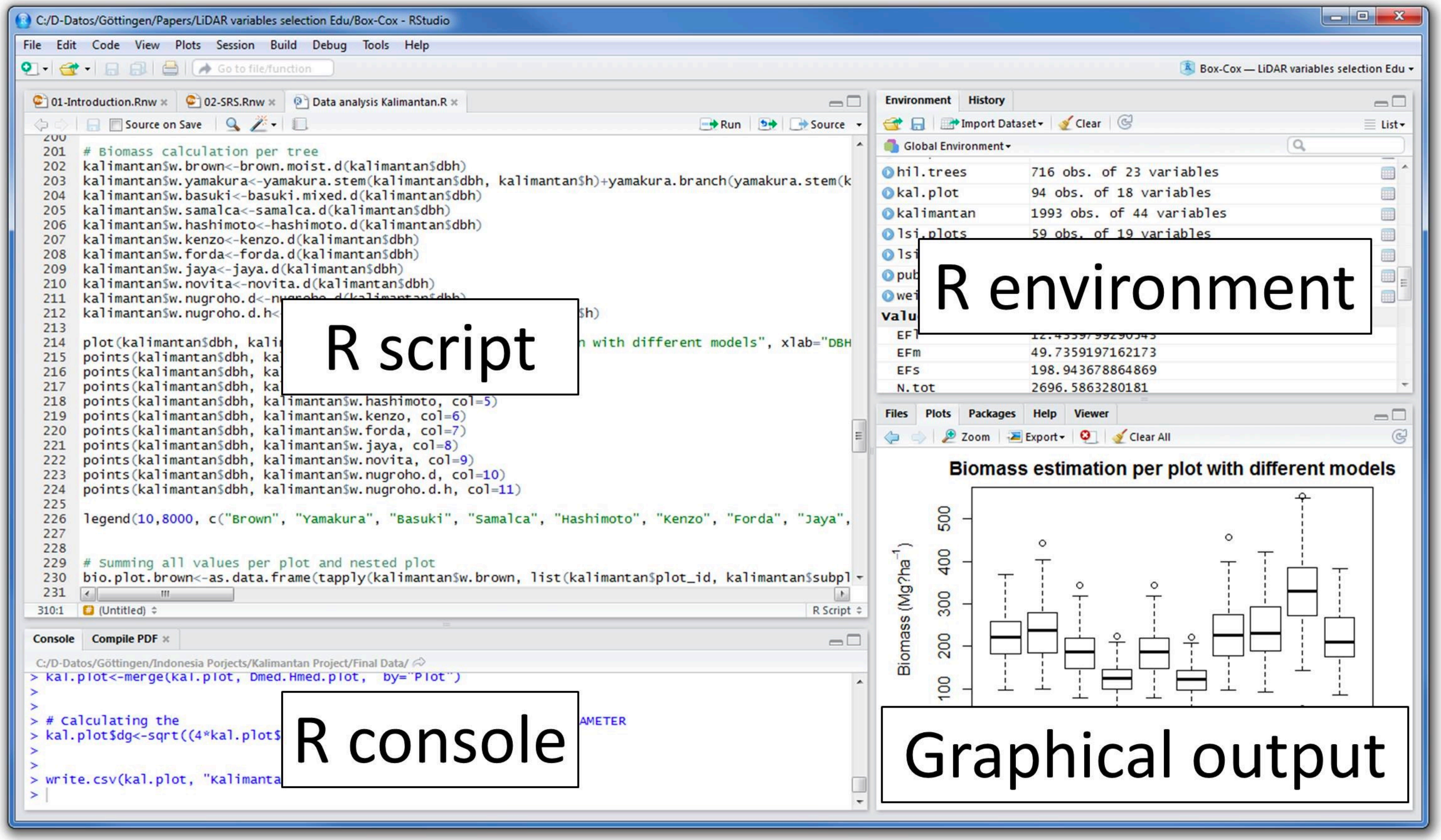
Why learn R?

- Free
- Open-source
- Cross-platform

RStudio

Let us start the action!

Rstudio IDE



```
# Biomass calculation per tree
kalimantan$w.brown<-brown.moist.d(kalimantan$dbh)
kalimantan$w.yamakura<-yamakura.stem(kalimantan$dbh, kalimantan$h)+yamakura.branch(yamakura.stem(k
kalimantan$w.basuki<-basuki.mixed.d(kalimantan$dbh)
kalimantan$w.samalca<-samalca.d(kalimantan$dbh)
kalimantan$w.hashimoto<-hashimoto.d(kalimantan$dbh)
kalimantan$w.kenzo<-kenzo.d(kalimantan$dbh)
kalimantan$w.forda<-forda.d(kalimantan$dbh)
kalimantan$w.jaya<-jaya.d(kalimantan$dbh)
kalimantan$w.novita<-novita.d(kalimantan$dbh)
kalimantan$w.nugroho.d<-nugroho.d(kalimantan$dbh)
kalimantan$w.nugroho.d.h<                                    $h)

plot(kalimantan$dbh, kali                                    n with different models", xlab="DBH
points(kalimantan$dbh, ka
points(kalimantan$dbh, ka
points(kalimantan$dbh, ka
points(kalimantan$dbh, kalimantan$w.hashimoto, col=5)
points(kalimantan$dbh, kalimantan$w.kenzo, col=6)
points(kalimantan$dbh, kalimantan$w.forda, col=7)
points(kalimantan$dbh, kalimantan$w.jaya, col=8)
points(kalimantan$dbh, kalimantan$w.novita, col=9)
points(kalimantan$dbh, kalimantan$w.nugroho.d, col=10)
points(kalimantan$dbh, kalimantan$w.nugroho.d.h, col=11)

legend(10,8000, c("Brown", "Yamakura", "Basuki", "Samalca", "Hashimoto", "Kenzo", "Forda", "Jaya",


# Summing all values per plot and nested plot
bio.plot.brown<-as.data.frame(tapply(kalimantan$w.brown, list(kalimantan$plot_id, kalimantan$subpl
```

```
hil.trees       716 obs. of 23 variables
kal.plot        94 obs. of 18 variables
kalimantan      1993 obs. of 44 variables
lsi.plots       59 obs. of 19 variables
lsi
pub
wei
Valu
  EF1
  EFm           49.7359197162173
  EFs           198.943678864869
  N.tot         2696.5863280181
```

```
> kal.plot<-merge(kal.plot, Dmed.Hmed.plot, by="Plot")
>
>
> # Calculating the                              AMETER
> kal.plot$dg<-sqrt((4*kal.plot$
>
>
> write.csv(kal.plot, "Kalimanta
>
```

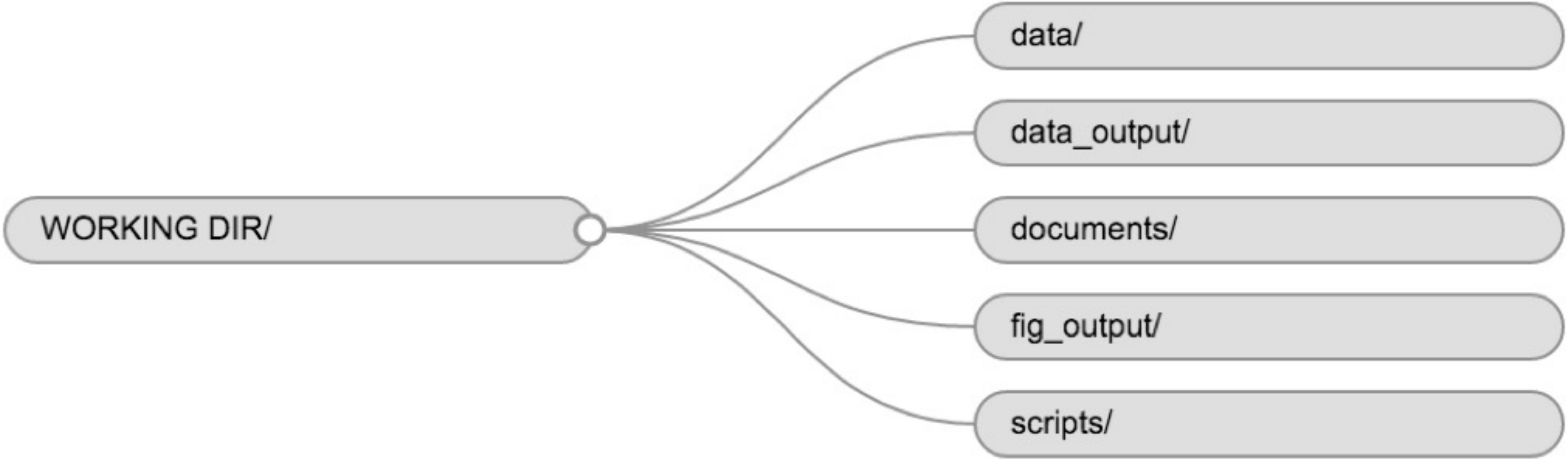
R script

R environment

R console

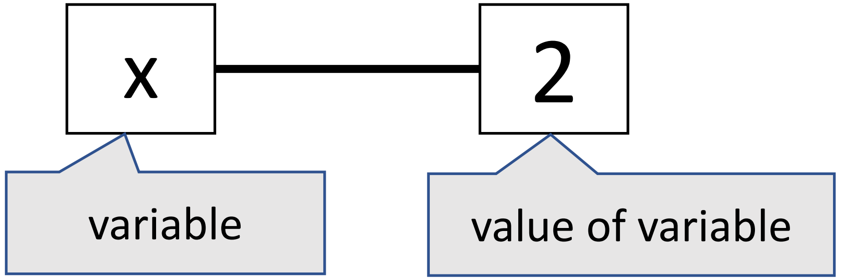
Graphical output

Working Directory



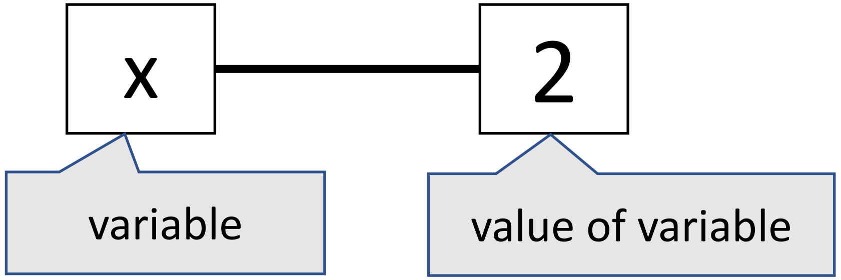
Variables

A variable is a letter or word that stores a value in it.



Variables

A variable is a letter or word that stores a value in it.

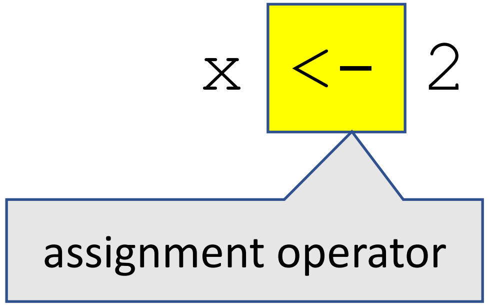


x

2

variable

value of variable



```
x <- 2
```

assignment operator

Variable names

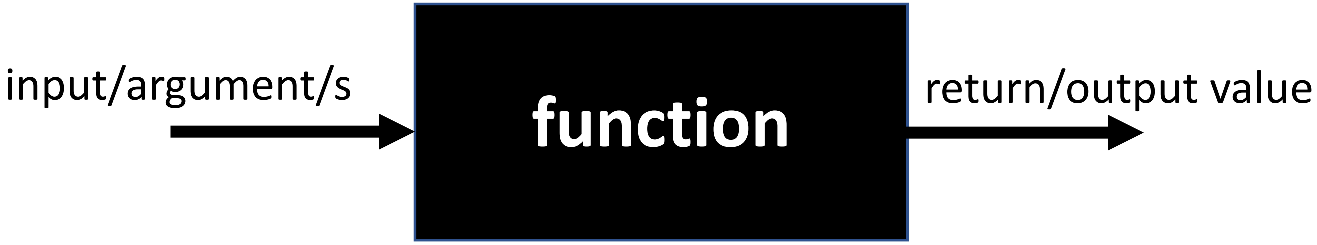
- Case-sensitive
- Do not start with numbers
- Not too long
- Some names cannot be used as they are reserved

  e.g. if <- 66 ✗
- Do not use function names
- Use nouns if using a word to represent a variable

  e.g. gene_id <- 12345 ✓
- Be consistent - style guide

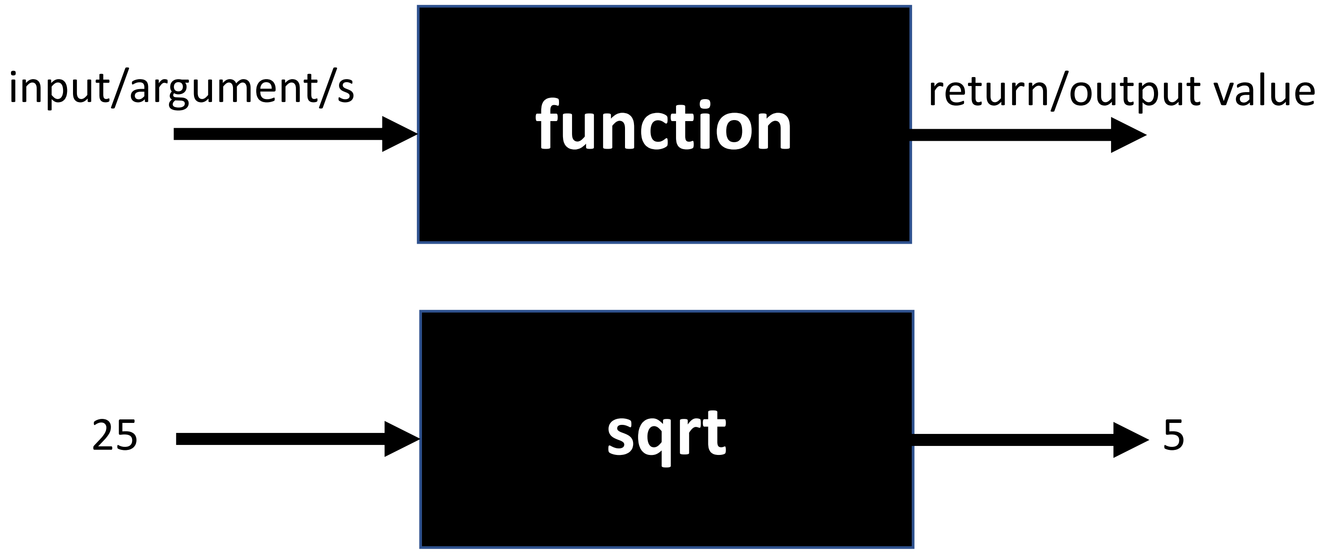
Calling functions

Functions execute a defined set of commands – automate a process



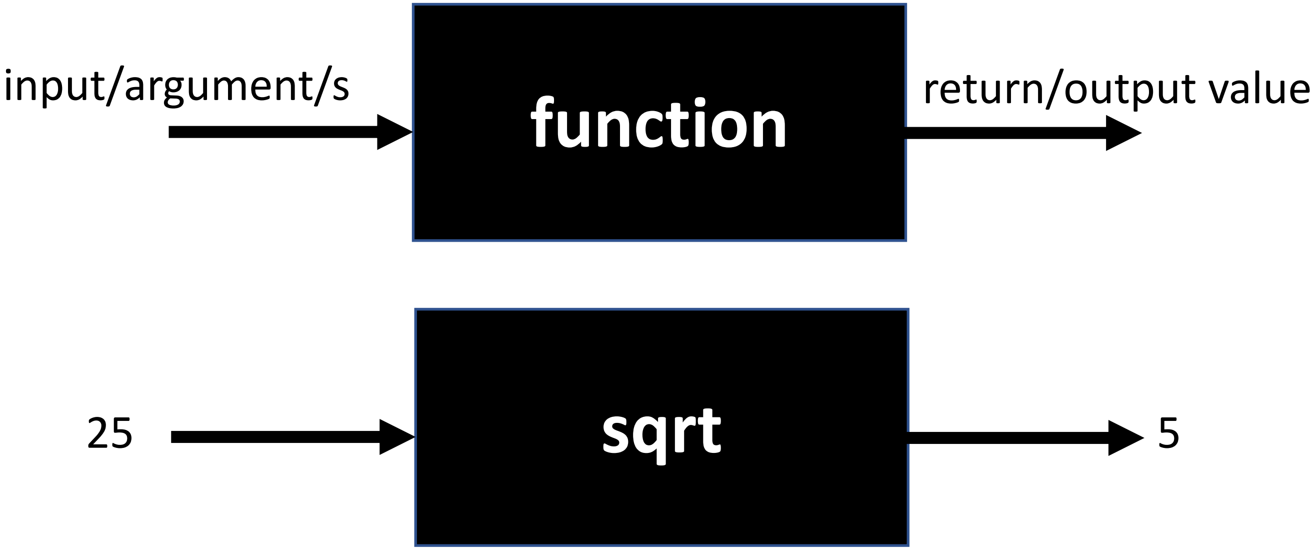
Calling functions

Functions execute a defined set of commands – automate a process



Calling functions

Functions execute a defined set of commands – automate a process



```
sqrt(25)
```

FUN QUIZ!

- What is temperature?

```
temperature <- 26.789
round (x = temperature, digits = 1)
```

A. **Variable**
B. **Function**
C. **Place holder**

(Poll Question – Choose the right option!)

FUN QUIZ !

- What is the value of temperature after executing both lines of code?

```
temperature <- 26.789
round (x = temperature, digits = 1)
```

A. 27

B. 26.7

C. 26.789

D. 26.8

(Poll Question – Choose the right option!)

Data types

- **logical**: `TRUE FALSE`
- **integer**: whole numbers. *e.g.,* `40`
- **double**: numbers with decimal points. *e.g.,* `2.666`
- **character**: words or **strings**. *e.g.,* `"Hello"`

Data structures

- **vector**: list of items of the same data type. *e.g.*, 4, 6, 9, 12
- **factor**: categorical data (has to be a character vector). *e.g.*, Male, Female
- **data.frame**: contains tabular data – normally data is loaded into data.frame when reading in a file

Vector

- List of data types (must be same type)
- One-dimensional

4	12	7	9

```
c(4,12,7,9)
```

Vector indices

1	2	3	4
4	12	7	9

Relational Operators

Used to compute a condition/comparisons:

==	equal
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
!=	not equal to

Logical operators

&	and
|	or
!	not

AND &

- TRUE & TRUE results in TRUE
- TRUE & FALSE results in FALSE
- FALSE & TRUE results in FALSE
- FALSE & FALSE results in FALSE

OR |

- TRUE | TRUE results in TRUE
- TRUE | FALSE results in TRUE
- FALSE | TRUE results in TRUE
- FALSE | FALSE results in FALSE

NOT !

- !TRUE results in FALSE
- !FALSE results in TRUE

What we have learned so far

- How to create a Project in RStudio
- How to code and execute R code in RStudio
- Create variables *e.g.*, `weight <- 24`
- Data types: integer (`11`), double (`11.01`), character (`“Hello”`), logical (`TRUE/FALSE`)
- Calling functions *e.g.*, `sqrt(25)`
- Create vectors *e.g.*, `weight_mm <- c(22, 24, 10, 34)`
- Index vectors *e.g.*, `weight_mm[3]`
- Relational and Logical operators `(& | ! == < > <= >= !=)`
- Missing data `NA`