DATA VISUALIZATION IN R

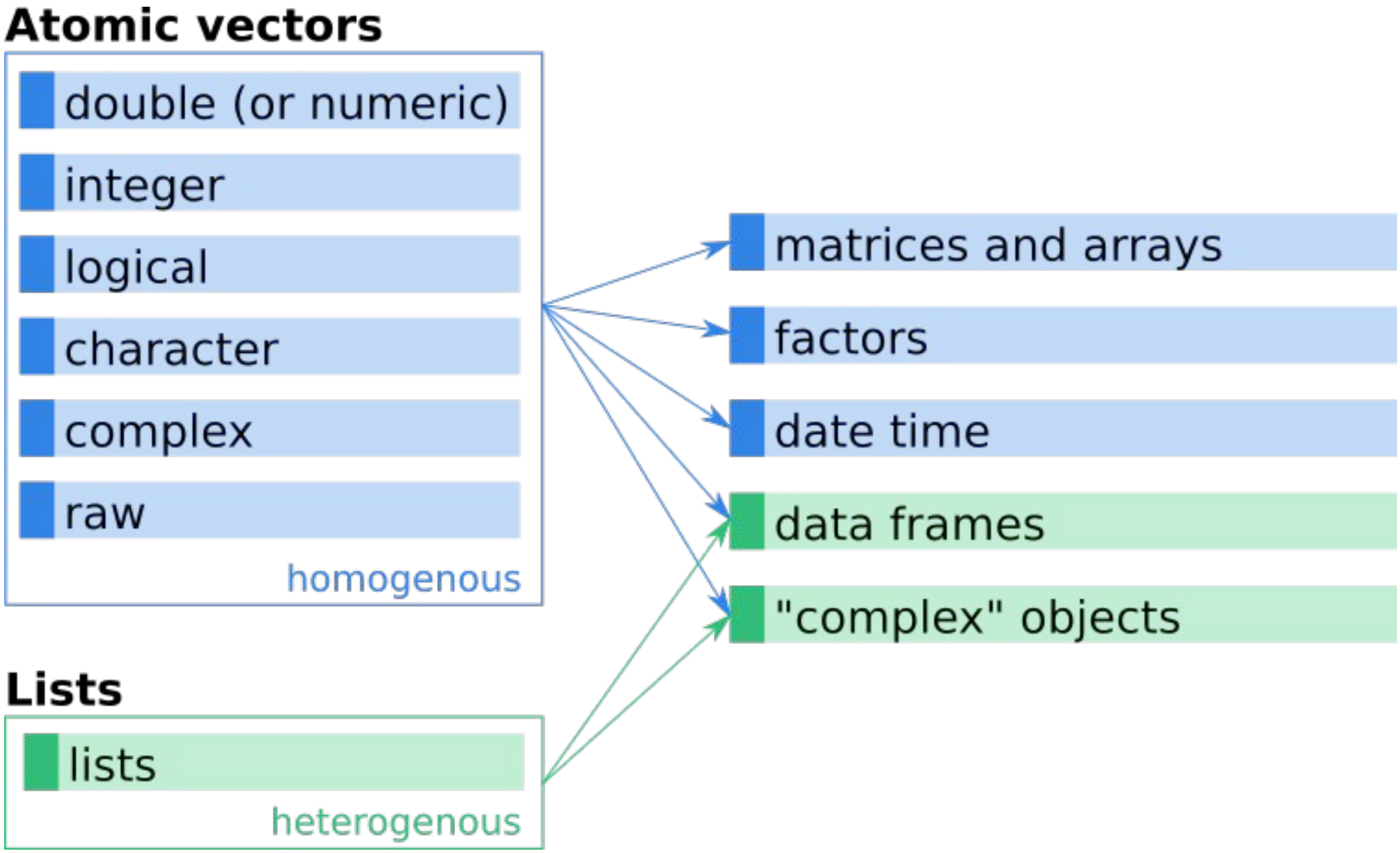
Presenter: Nguyen Le Duc Minh, MD

Sep-8th-2024

Contents

1. Data types
2. Data structure
3. Import and export data
4. Data visualization
5. Homework

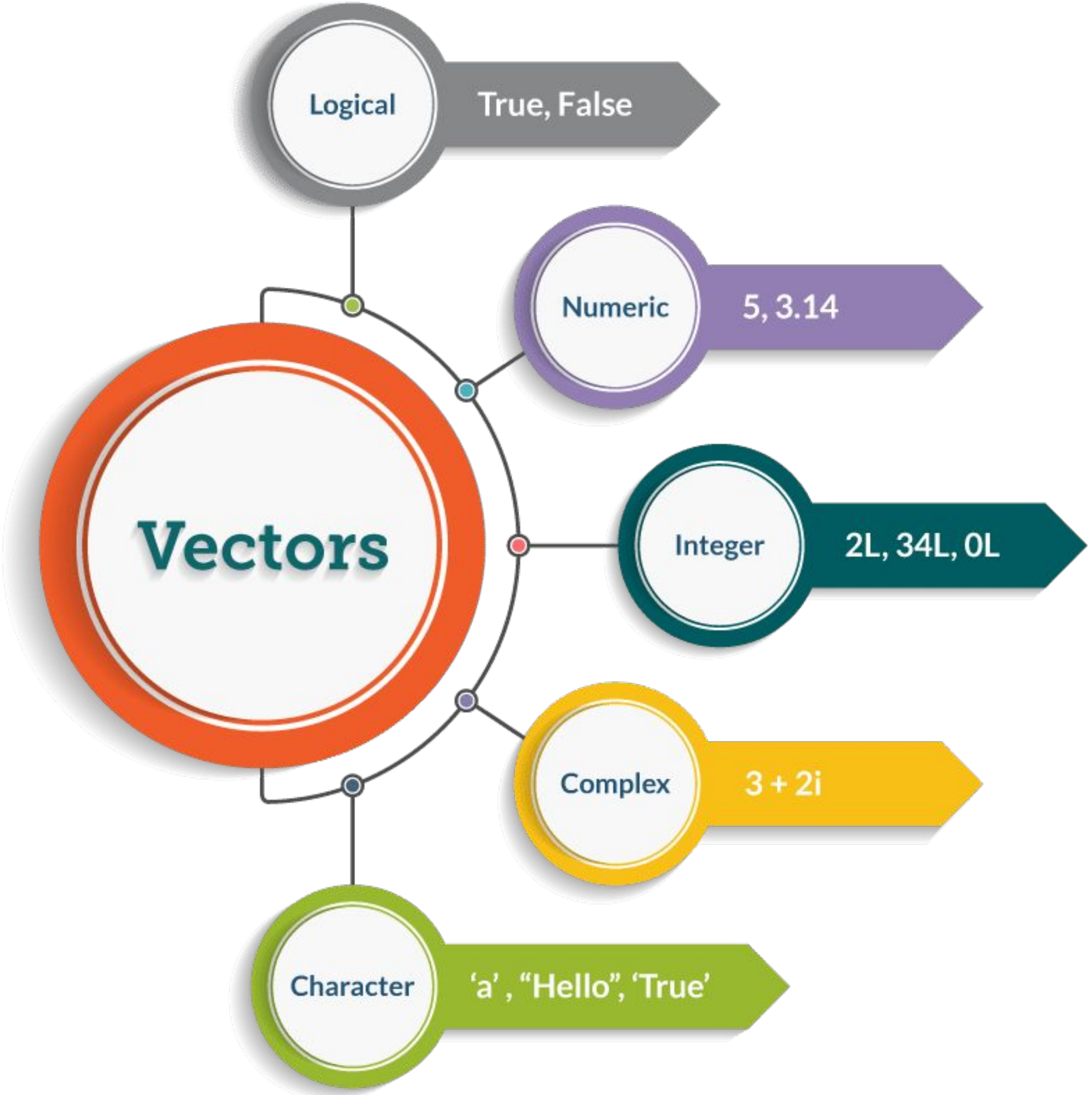
DIFFERENT R OBJECTS



https://discdown.org/rprogramming/vectors.html

DATA TYPES IN R

DATA TYPES IN R



Vectors

Logical: True, False

Numeric: 5, 3.14

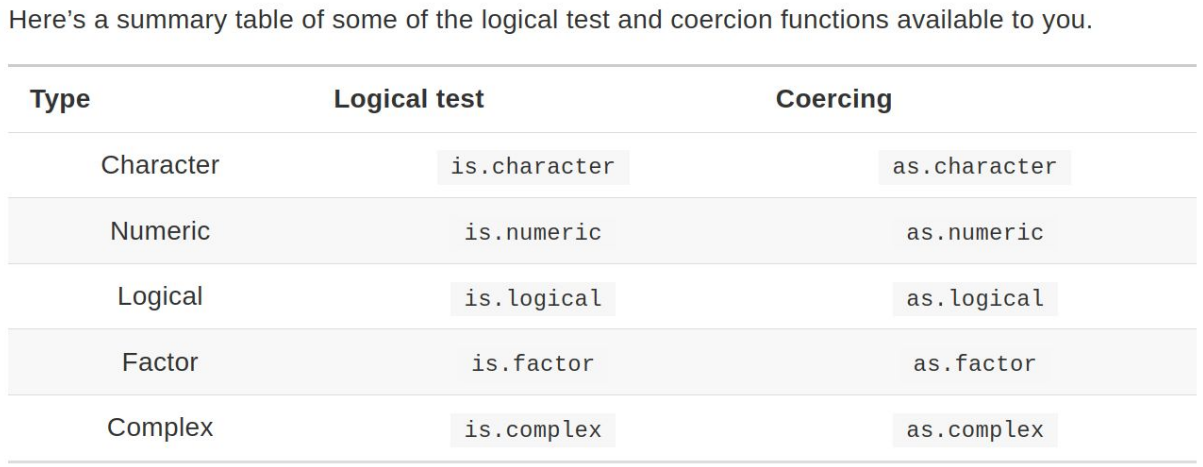
Integer: 2L, 34L, 0L

Complex: 3 + 2i

Character: ‘a’ , “Hello”, ‘True’

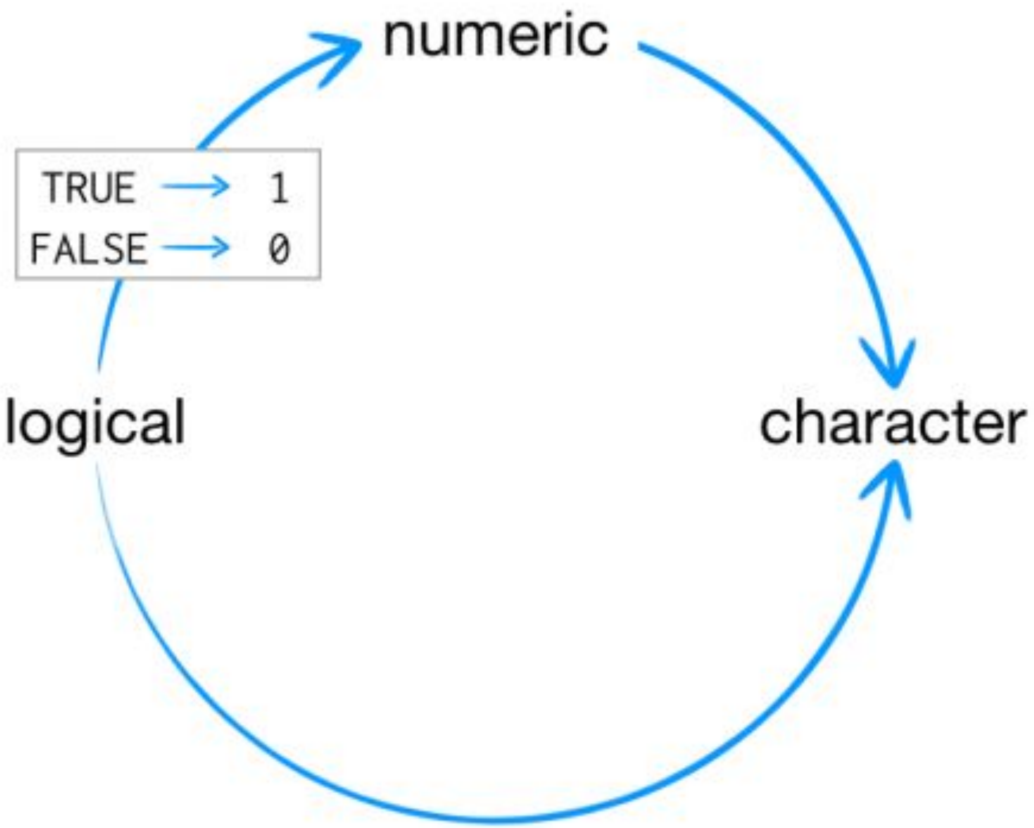
https://www.edureka.co/blog/r-programming-language

What is Coercion in R?



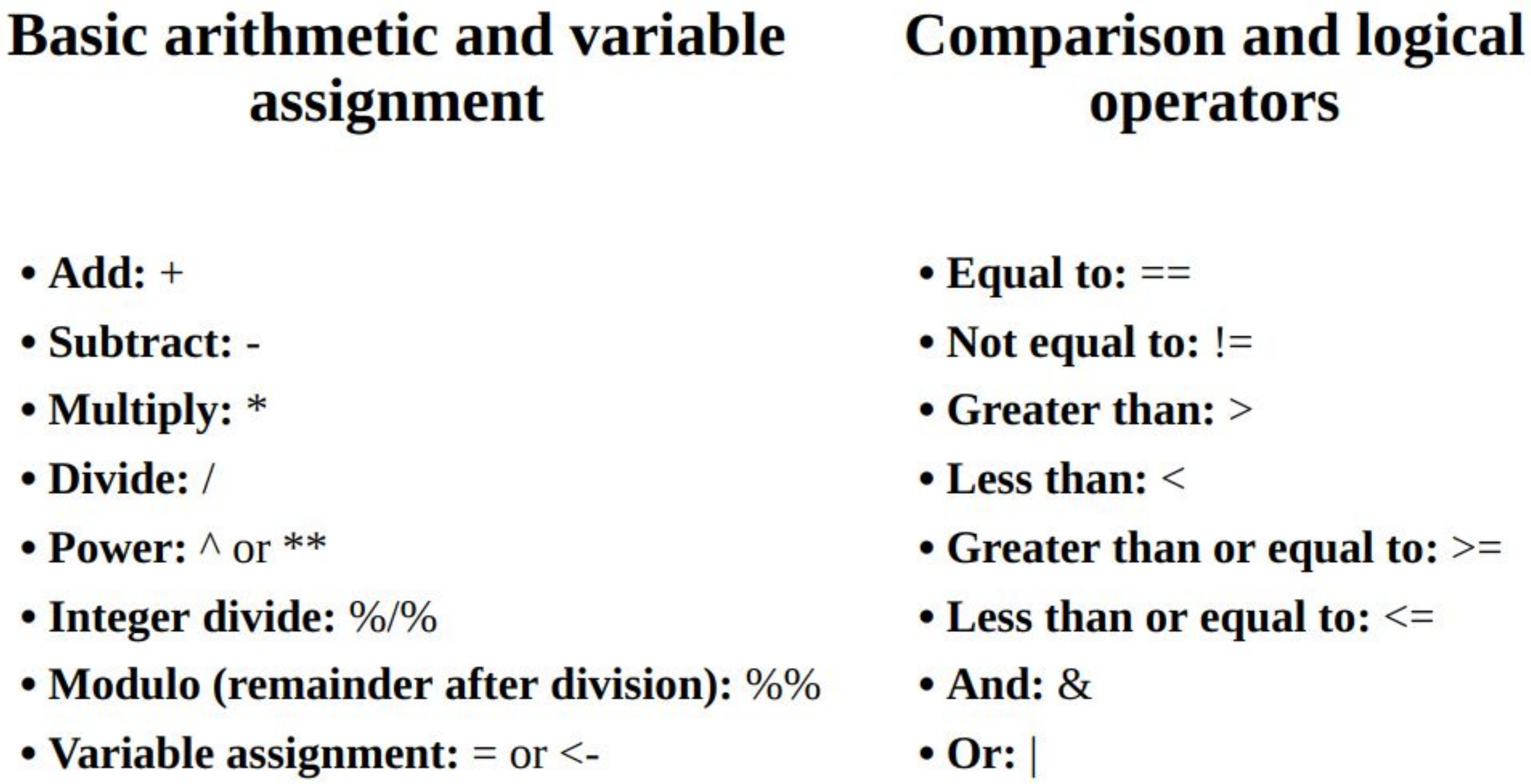
Here’s a summary table of some of the logical test and coercion functions available to you.

Type	Logical test	Coercing
Character	`is.character`	`as.character`
Numeric	`is.numeric`	`as.numeric`
Logical	`is.logical`	`as.logical`
Factor	`is.factor`	`as.factor`
Complex	`is.complex`	`as.complex`



https://www.zigya.com/blog/what-is-coercion-in-r/

Caculation and Comparison in R



Basic arithmetic and variable assignment

- **Add:** +
- **Subtract:** -
- **Multiply:** *
- **Divide:** /
- **Power:** ^ or **
- **Integer divide:** %/%
- **Modulo (remainder after division):** %%
- **Variable assignment:** = or <-

Comparison and logical operators

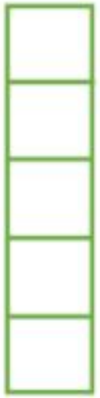
- **Equal to:** ==
- **Not equal to:** !=
- **Greater than:** >
- **Less than:** <
- **Greater than or equal to:** >=
- **Less than or equal to:** <=
- **And:** &
- **Or:** |

DATA STRUCTURES IN R

DATA STRUCTURES IN R

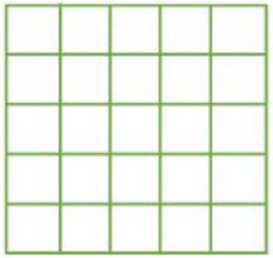
Scalars and vectors

scalar

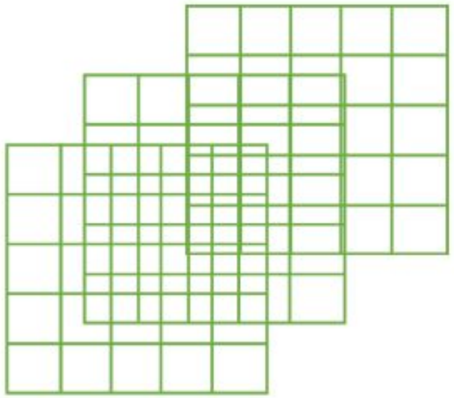


vector

Matrices and arrays

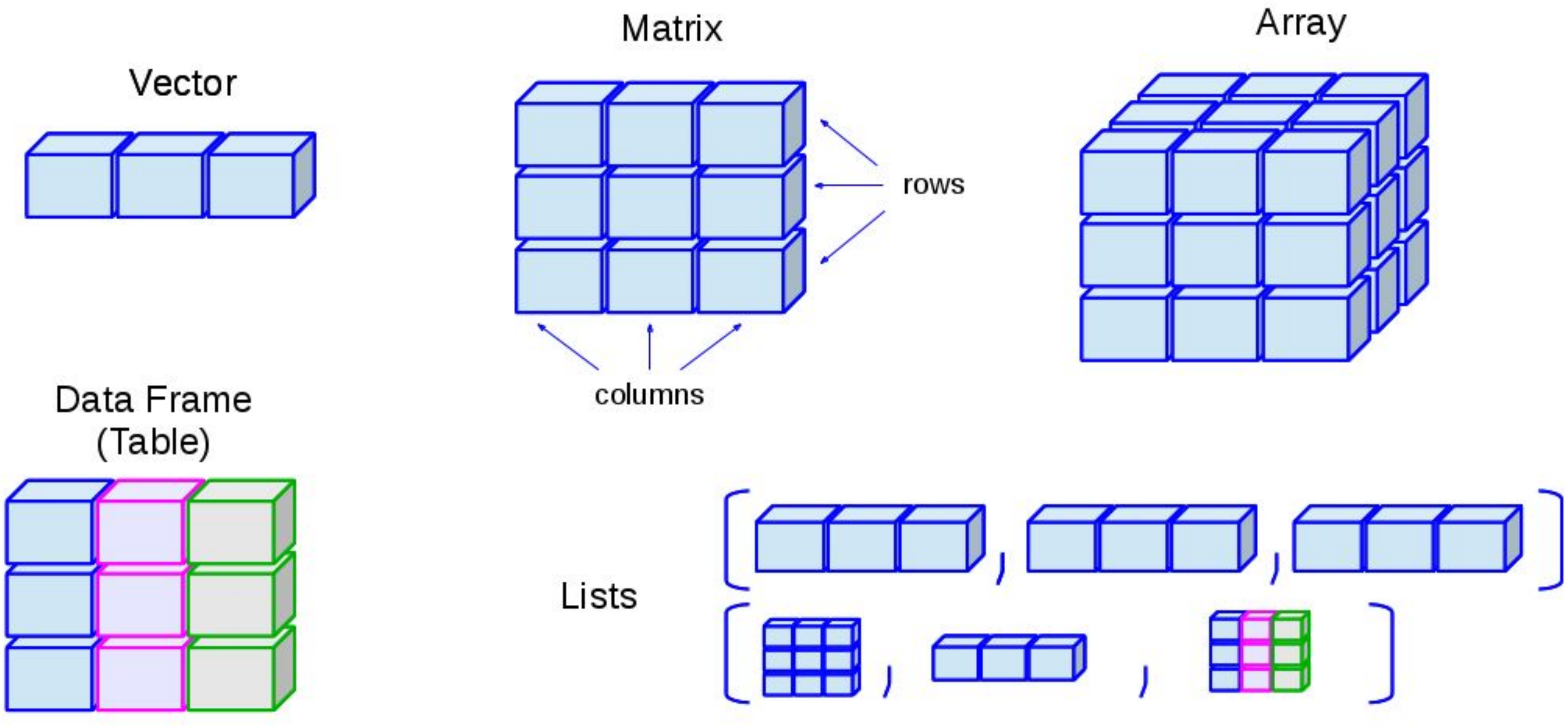


matrix



array

DATA STRUCTURES IN R



http://venus.ifca.unican.es/Rintro/dataStruct.html

Importing data

FILE TYPES

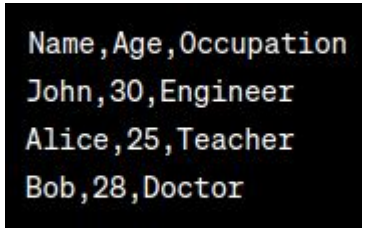
CSV (Comma-Separated Values): CSV files are plain text files that store tabular data, with each line representing a row of data, and values separated by commas (or other delimiters). CSV is one of the most widely used formats for exchanging data between different applications because of its simplicity and compatibility.

TSV (Tab-Separated Values): TSV files are similar to CSV files but use tabs as the delimiter between values instead of commas. TSV files are also plain text files and are often used when data includes commas within the values.

TXT (Text File): TXT files are plain text files that do not have a standardized format for storing tabular data. They can be used to store any text-based information but do not have specific rules for structuring tabular data like CSV or TSV.

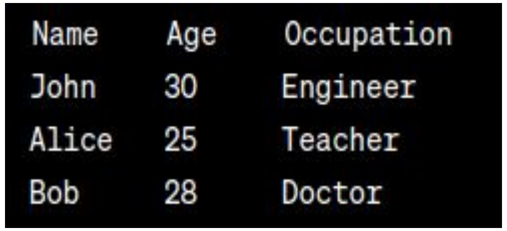
Excel Files (XLS, XLSX): Microsoft Excel files that can store tabular data in multiple sheets along with formatting, formulas, and charts. XLS is the older format used by Excel versions before 2007, while XLSX is the newer XML-based format used by Excel 2007 and later.

CSV



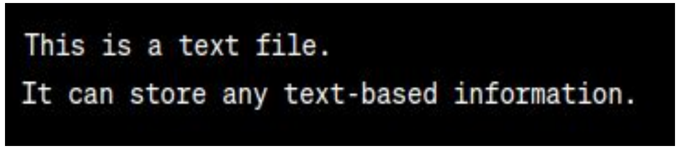
```
Name,Age,Occupation
John,30,Engineer
Alice,25,Teacher
Bob,28,Doctor
```

TSV

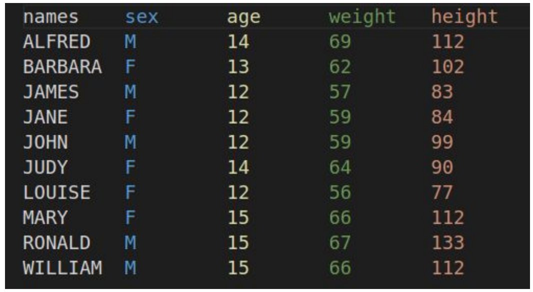


```
Name     Age     Occupation
John     30      Engineer
Alice    25      Teacher
Bob      28      Doctor
```

TXT



```
This is a text file.
It can store any text-based information.
```



```
names    sex     age     weight  height
ALFRED   M       14      69      112
BARBARA  F       13      62      102
JAMES    M       12      57      83
JANE     F       12      59      84
JOHN     M       12      59      99
JUDY     F       14      64      90
LOUISE   F       12      56      77
MARY     F       15      66      112
RONALD   M       15      67      133
WILLIAM  M       15      66      112
```

READ SINGLE FILE IN R

Importing a CSV file in R

```
data_1 <- read_csv('path_to_file.csv', header=T)

data_2 <- read_table('path_to_file.csv', header=T)
```

Importing a TXT, TSV file in R

```
data_3 <- read.delim('path_to_file.txt',header = T)
```

Importing data from Excel into R

```
library(readxl)
data_4 <- read_excel("path_to_file.xlsx", sheet = 1)
```

DEAL WITH MULTI LARGE DATASET

```
install.packages(“data.table”)

library(data.table)
```

Why use data.table packages?

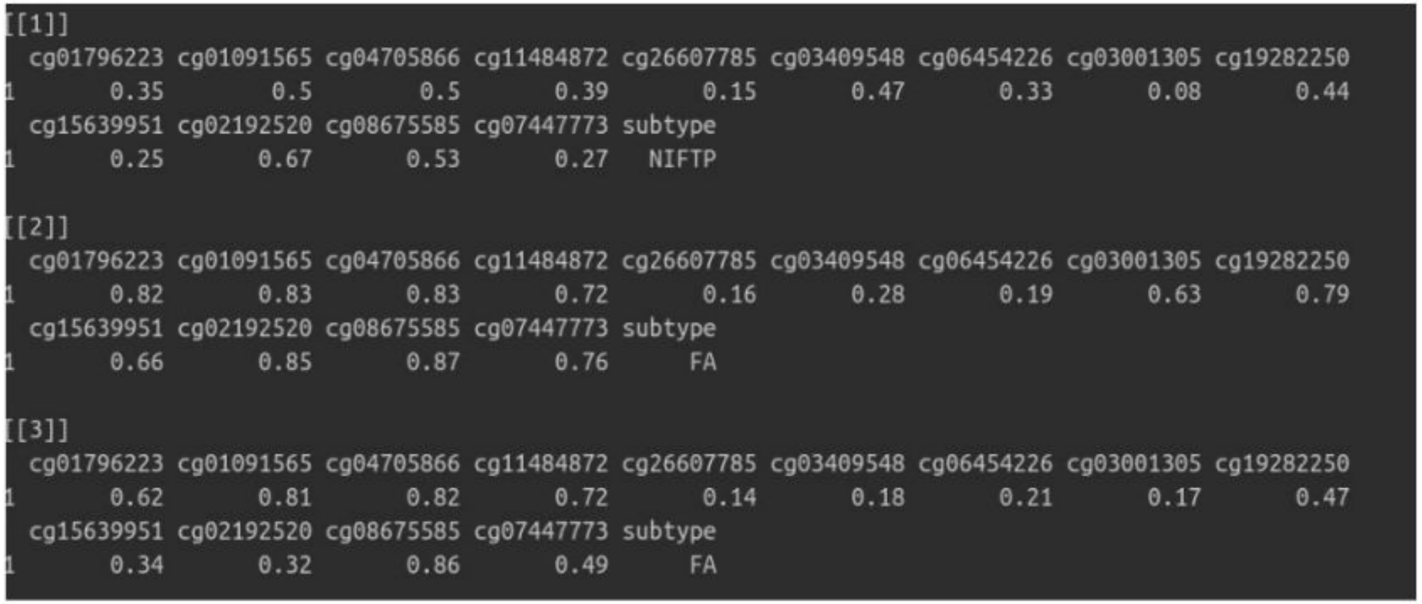
- concise syntax: fast to type, fast to read
- fast speed
- memory efficient
- careful API lifecycle management
- community
- feature rich

```
data <- fread('/home/acer/Downloads/dowload/data.tsv', header=T)
```

https://www.youtube.com/watch?v=opai4Wmp6Zo

do.call vs. Reduce

‘Reduce’ uses a binary function to successively combine the elements of a given vector and a possibly given initial value.
‘do.call’ constructs and executes a function call from a name or a function and a list of arguments to be passed to it.

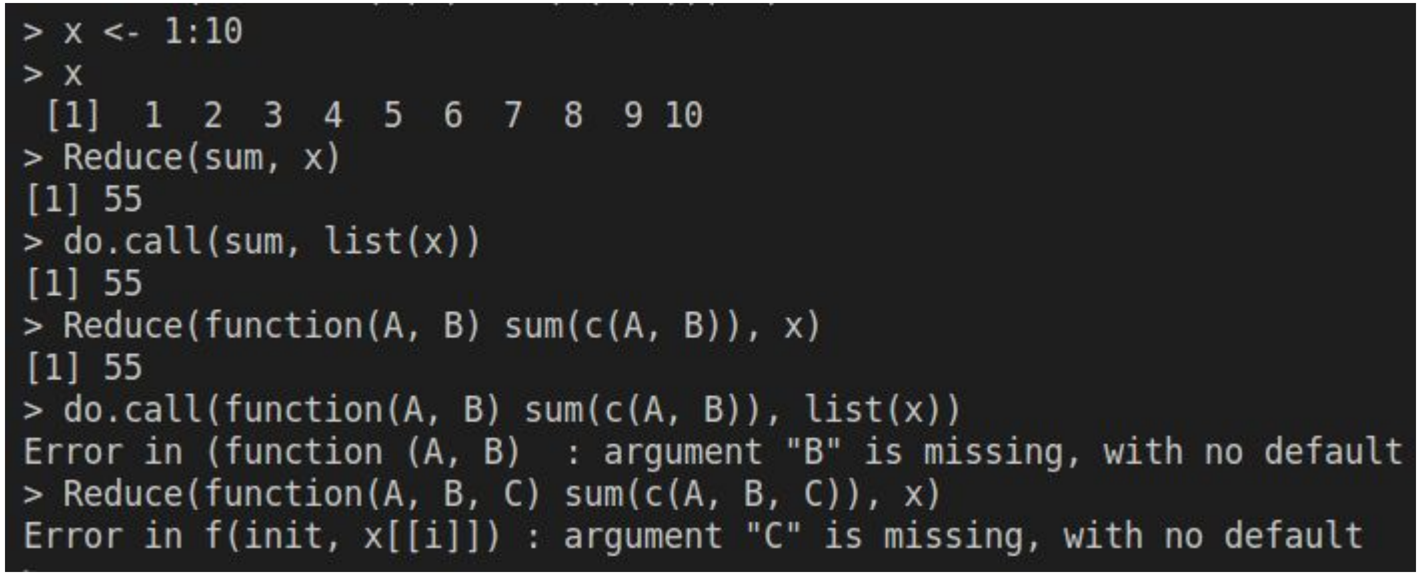


```
[[1]]
  cg01796223 cg01091565 cg04705866 cg11484872 cg26607785 cg03409548 cg06454226 cg03001305 cg19282250
1       0.35        0.5        0.5       0.39       0.15       0.47       0.33       0.08       0.44
  cg15639951 cg02192520 cg08675585 cg07447773 subtype
1       0.25       0.67       0.53       0.27   NIFTP

[[2]]
  cg01796223 cg01091565 cg04705866 cg11484872 cg26607785 cg03409548 cg06454226 cg03001305 cg19282250
1       0.82       0.83       0.83       0.72       0.16       0.28       0.19       0.63       0.79
  cg15639951 cg02192520 cg08675585 cg07447773 subtype
1       0.66       0.85       0.87       0.76      FA

[[3]]
  cg01796223 cg01091565 cg04705866 cg11484872 cg26607785 cg03409548 cg06454226 cg03001305 cg19282250
1       0.62       0.81       0.82       0.72       0.14       0.18       0.21       0.17       0.47
  cg15639951 cg02192520 cg08675585 cg07447773 subtype
1       0.34       0.32       0.86       0.49      FA
```

```
# do.call
data_1 <- do.call("rbind", data_f )

# reduce
data_2<- Reduce(function(x, y) rbind(x,y), data_f)
```

Compare do.call vs. Reduce

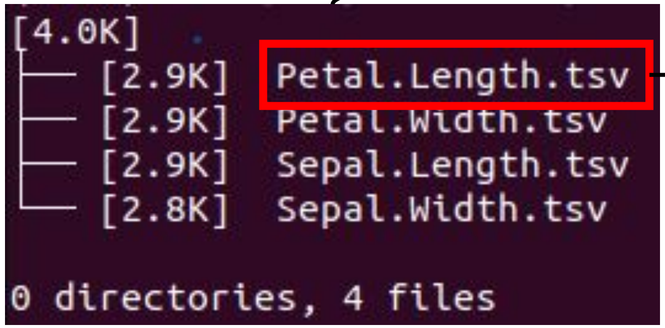
do.call () vs. Reduce() allow you to call any R function, but instead of writing out the arguments one by one, you can use a list to hold the arguments of the function.



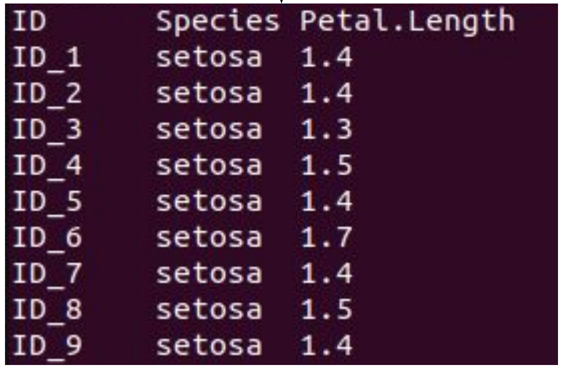
```
> x <- 1:10
> x
 [1]  1  2  3  4  5  6  7  8  9 10
> Reduce(sum, x)
[1] 55
> do.call(sum, list(x))
[1] 55
> Reduce(function(A, B) sum(c(A, B)), x)
[1] 55
> do.call(function(A, B) sum(c(A, B)), list(x))
Error in (function (A, B)  : argument "B" is missing, with no default
> Reduce(function(A, B, C) sum(c(A, B, C)), x)
Error in f(init, x[[i]]) : argument "C" is missing, with no default
```

Do.call	Reduce
apply function for all elements in list	Take 2 elements to apply function for each (only 2 element)

READ MULTI FILES IN R

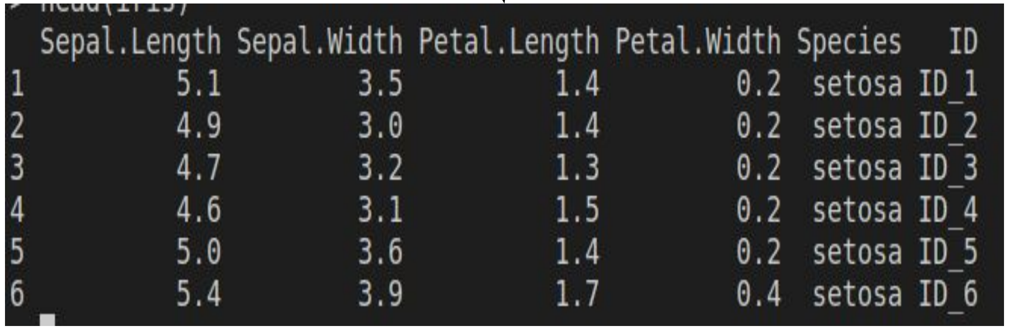


```
[4.0K]  .
├── [2.9K]  Petal.Length.tsv
├── [2.9K]  Petal.Width.tsv
├── [2.9K]  Sepal.Length.tsv
└── [2.8K]  Sepal.Width.tsv

0 directories, 4 files
```

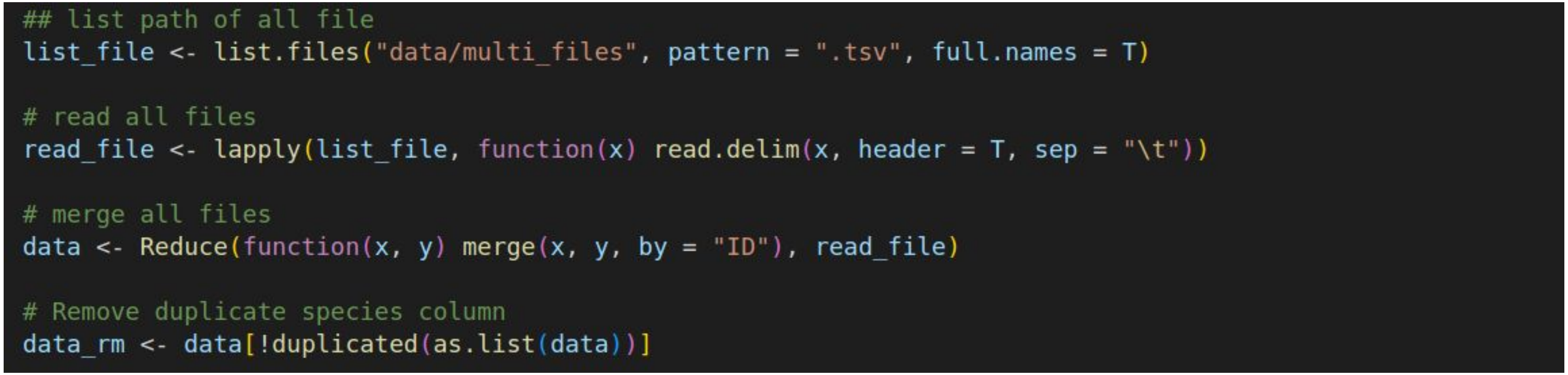


```
ID      Species Petal.Length
ID_1    setosa  1.4
ID_2    setosa  1.4
ID_3    setosa  1.3
ID_4    setosa  1.5
ID_5    setosa  1.4
ID_6    setosa  1.7
ID_7    setosa  1.4
ID_8    setosa  1.5
ID_9    setosa  1.4
```

Target



```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species   ID
1          5.1         3.5          1.4         0.2  setosa ID_1
2          4.9         3.0          1.4         0.2  setosa ID_2
3          4.7         3.2          1.3         0.2  setosa ID_3
4          4.6         3.1          1.5         0.2  setosa ID_4
5          5.0         3.6          1.4         0.2  setosa ID_5
6          5.4         3.9          1.7         0.4  setosa ID_6
```



```r
## list path of all file
list_file <- list.files("data/multi_files", pattern = ".tsv", full.names = T)

# read all files
read_file <- lapply(list_file, function(x) read.delim(x, header = T, sep = "\t"))

# merge all files
data <- Reduce(function(x, y) merge(x, y, by = "ID"), read_file)

# Remove duplicate species column
data_rm <- data[!duplicated(as.list(data))]
```

EXPORT DATA IN R

Write data to txt file:

```
write.table(object, file = "name_file.txt", sep = "\t",row.names = TRUE, col.names = NA)
```

Write data to csv files

```
write.csv(object, file = "name_file.csv")
```

Writing data from R to Excel files

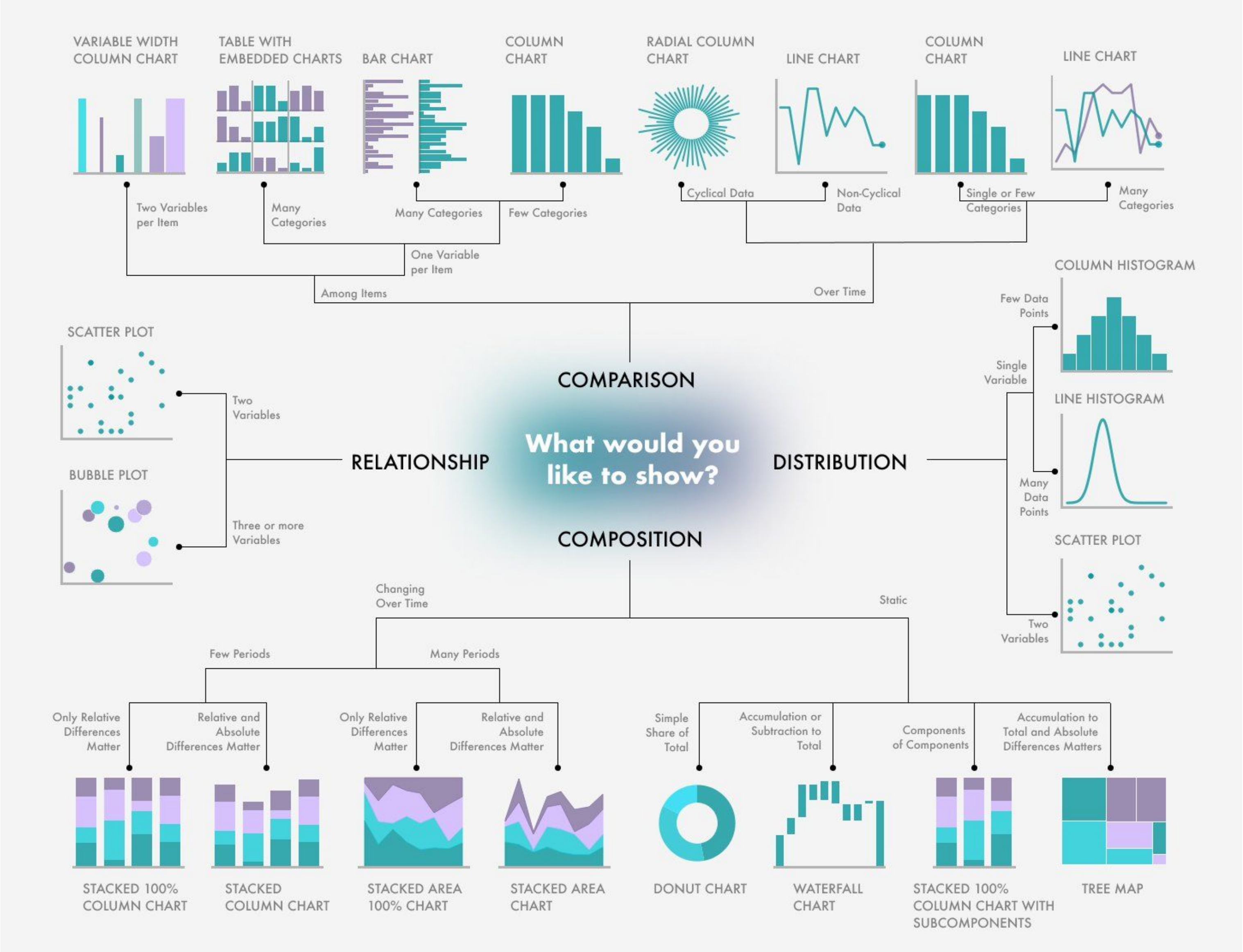
```
library("xlsx")
# Write the first data set in a new workbook
write.xlsx(USArrests, file = "myworkbook.xlsx", sheetName = "USA-ARRESTS", append = FALSE)
# Add a second data set in a new worksheet
write.xlsx(mtcars, file = "myworkbook.xlsx", sheetName="MTCARS", append=TRUE)
```

DATA VISUALIZATION

INTRODUCTION

THERE ARE FOUR BASIC PRESENTATION TYPES THAT YOU CAN USE PRESENT YOUR DATA:

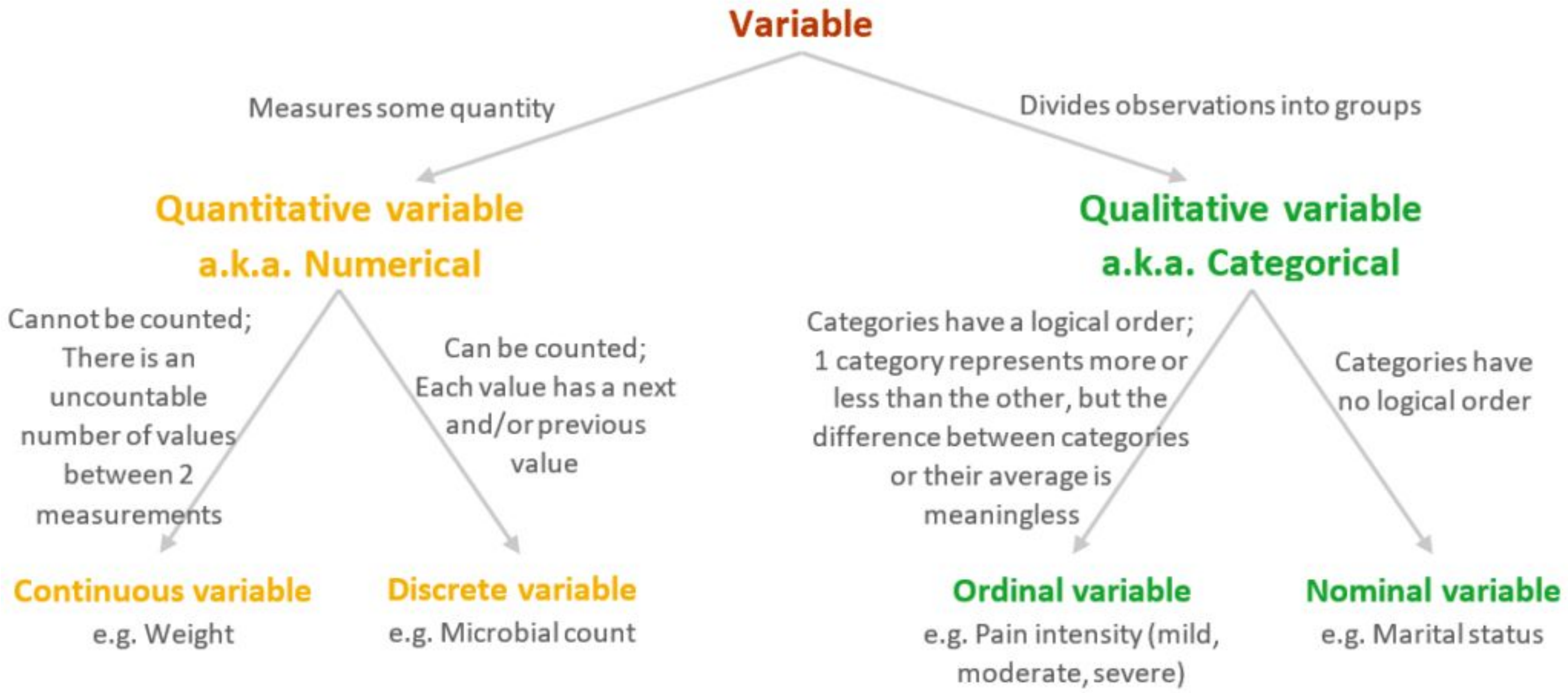
1. Comparison
2. Relationship
3. Distribution
4. Composition



Inspiration: 2009 A. Abela – a.v.abela@gmail.com. Chart Suggestions—A Thought-Starter.

DESIGNED BY ADEDAMOLA | LADIPO

Variable Types



Variable

Measures some quantity

Divides observations into groups

Quantitative variable
a.k.a. Numerical

Qualitative variable
a.k.a. Categorical

Cannot be counted;
There is an uncountable number of values between 2 measurements

Can be counted;
Each value has a next and/or previous value

Categories have a logical order;
1 category represents more or less than the other, but the difference between categories or their average is meaningless

Categories have no logical order

Continuous variable
e.g. Weight

Discrete variable
e.g. Microbial count

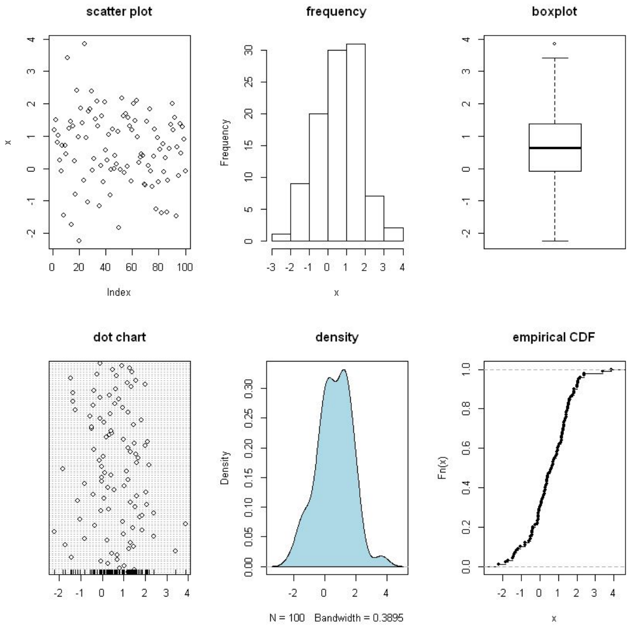
Ordinal variable
e.g. Pain intensity (mild, moderate, severe)

Nominal variable
e.g. Marital status

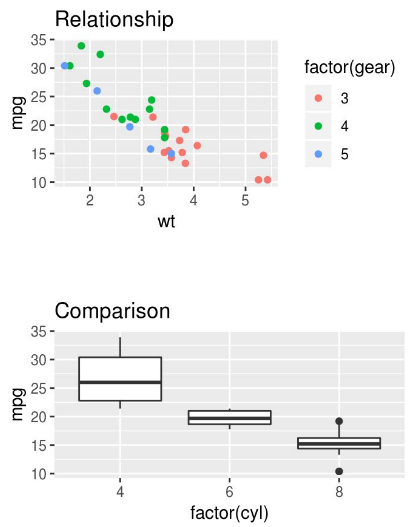
https://quantifyinghealth.com/variable-types/

Univariate and Bivariate Graphs

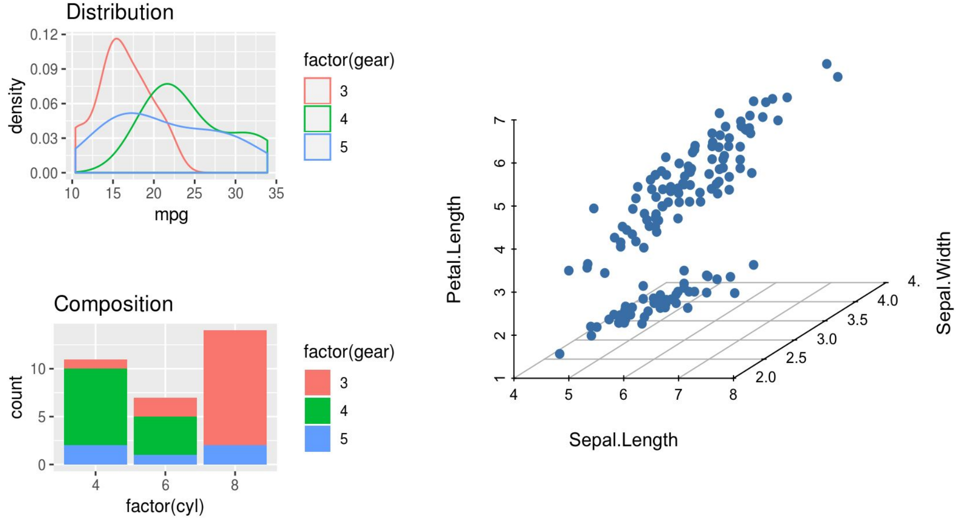
- **Univariate** analysis is when only **one variable** is analyzed.
- **Bivariate** data analysis is when exactly **two variables** are analyzed.
- **Multivariate** analysis is when **more than two variables** get analyzed.



Univariate Graph



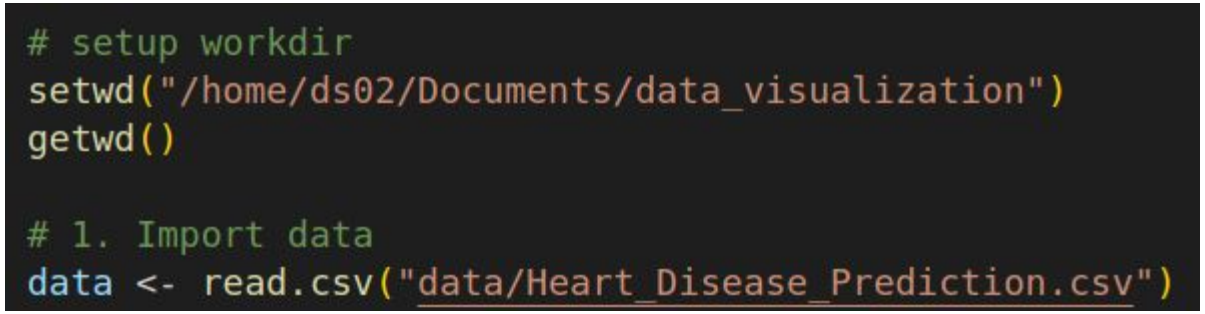
Bivariate Graph



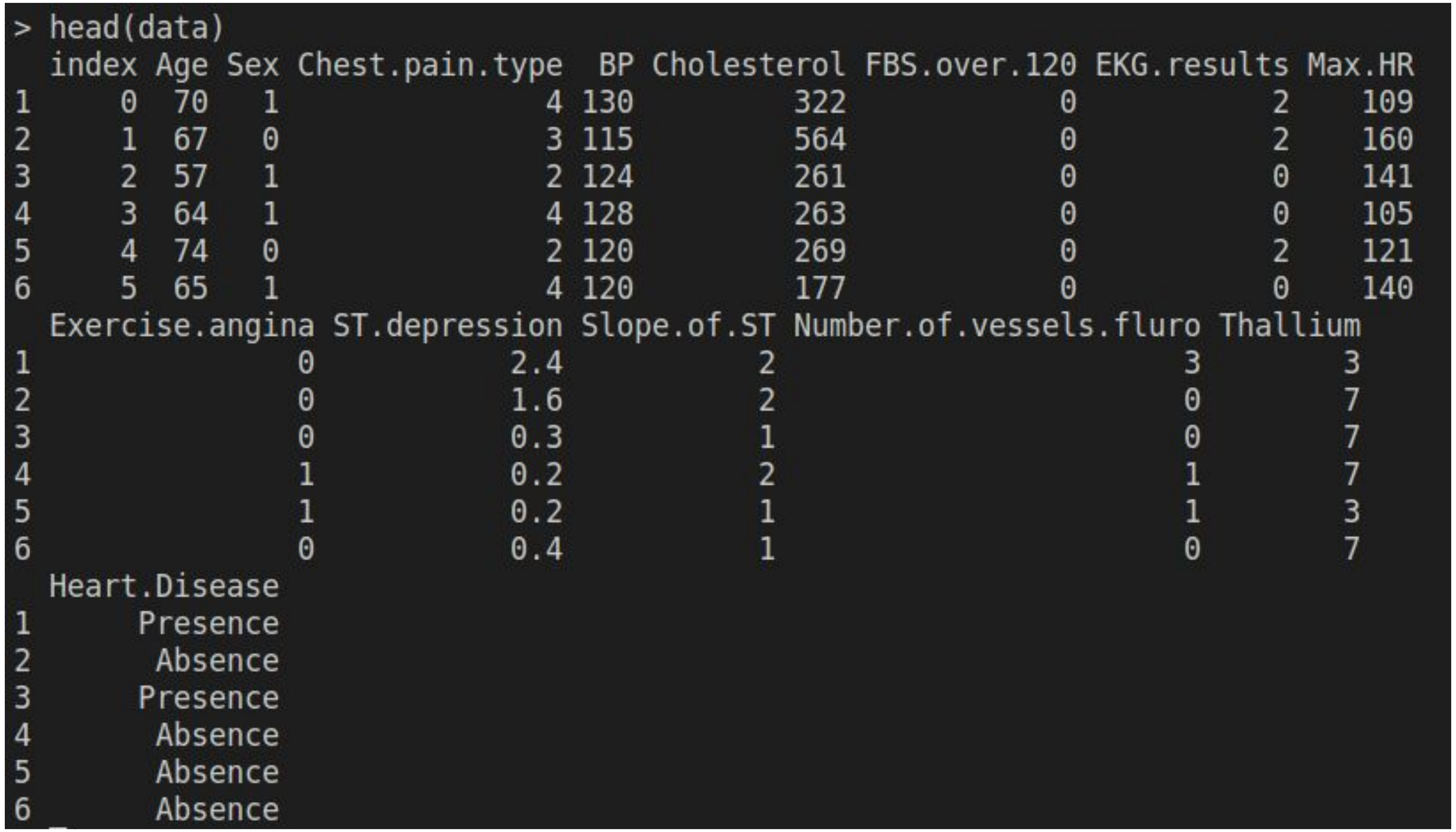
Multivariate graph

BASE PLOT AND GGPLOT2

IMPORT DATA INTO R



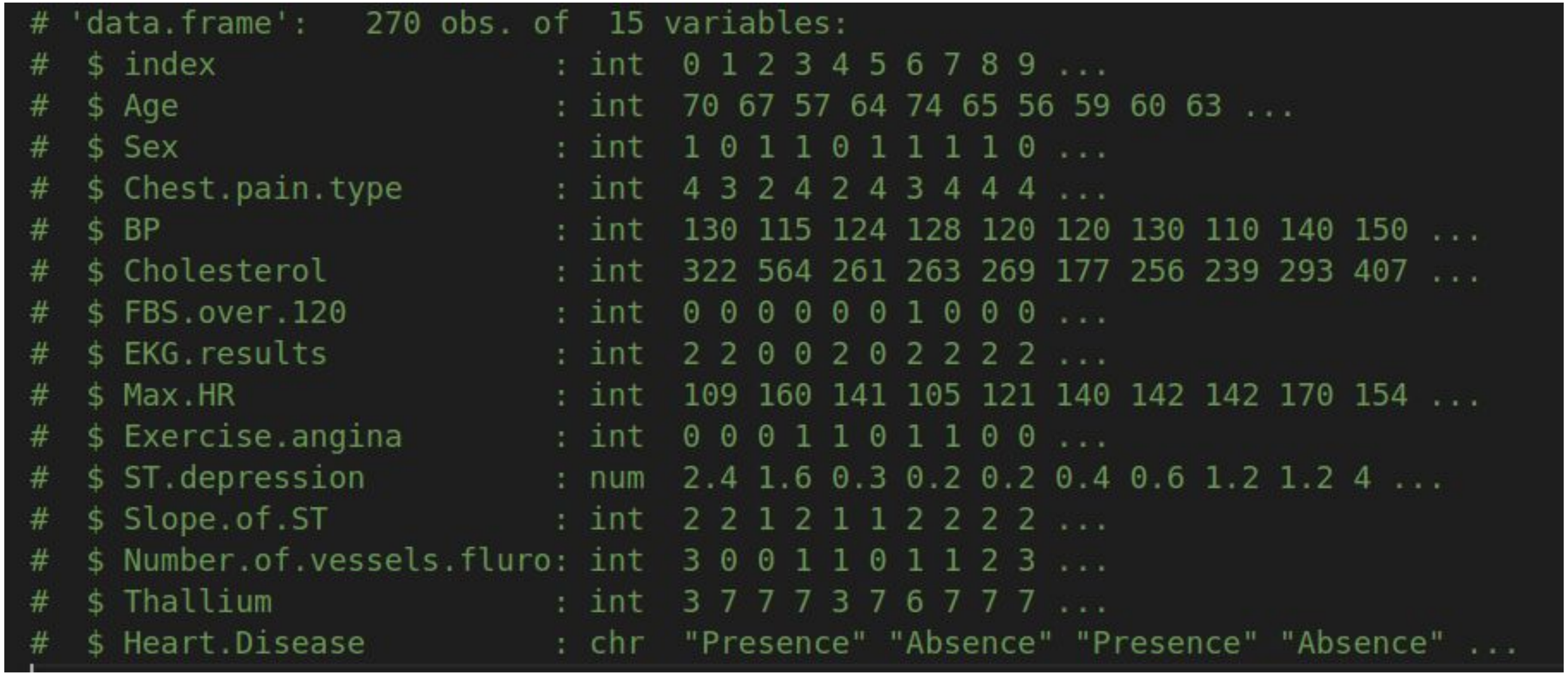
```
# setup workdir
setwd("/home/ds02/Documents/data_visualization")
getwd()

# 1. Import data
data <- read.csv("data/Heart_Disease_Prediction.csv")
```



```
> head(data)
  index Age Sex Chest.pain.type  BP Cholesterol FBS.over.120 EKG.results Max.HR
1     0  70   1               4 130         322            0           2    109
2     1  67   0               3 115         564            0           2    160
3     2  57   1               2 124         261            0           0    141
4     3  64   1               4 128         263            0           0    105
5     4  74   0               2 120         269            0           2    121
6     5  65   1               4 120         177            0           0    140
  Exercise.angina ST.depression Slope.of.ST Number.of.vessels.fluro Thallium
1               0           2.4           2                       3        3
2               0           1.6           2                       0        7
3               0           0.3           1                       0        7
4               1           0.2           2                       1        7
5               1           0.2           1                       1        3
6               0           0.4           1                       0        7
  Heart.Disease
1      Presence
2       Absence
3      Presence
4       Absence
5       Absence
6       Absence
```

Introduce data in R

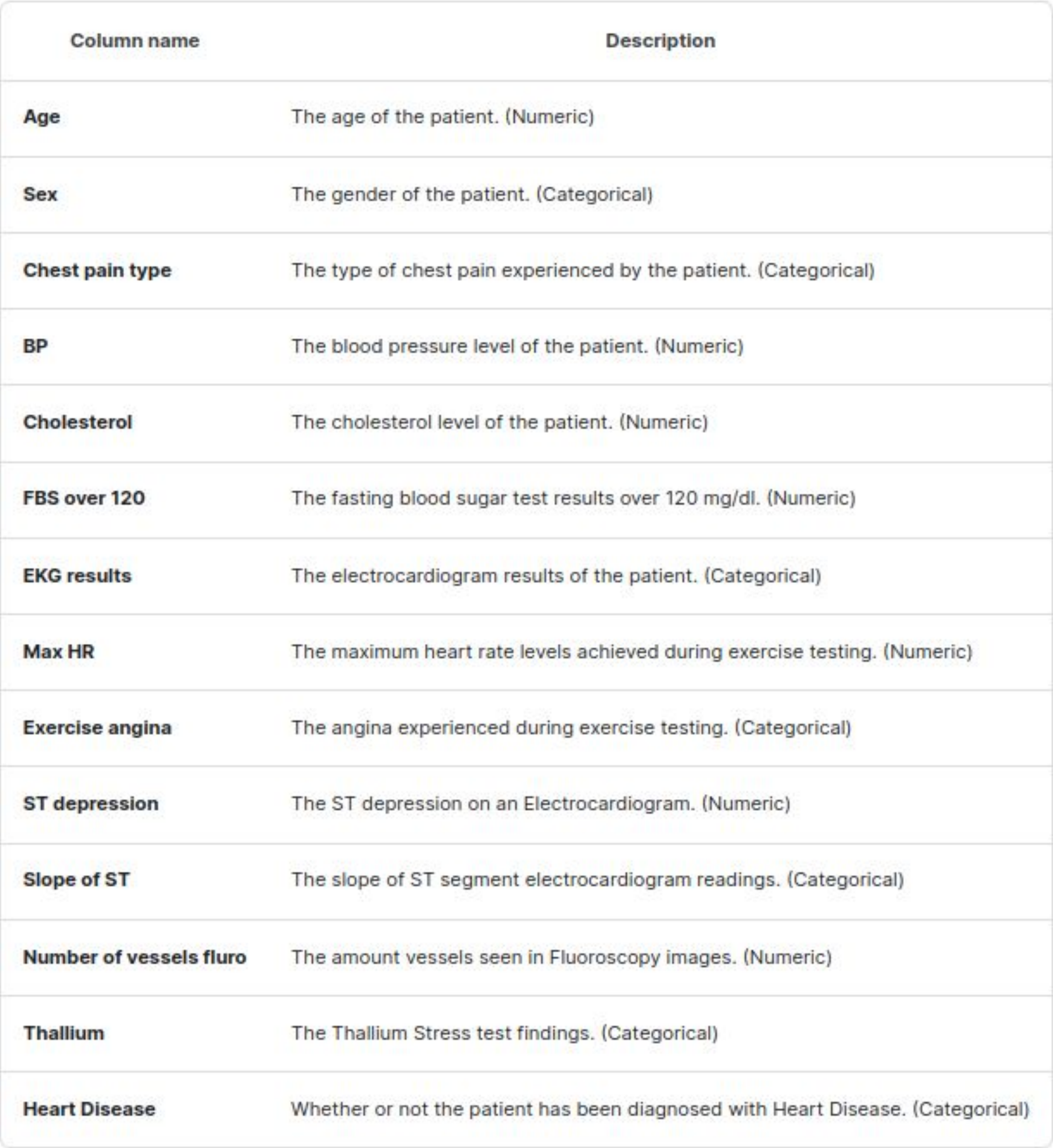
This dataset contains **270 case studies of individuals** classified as either having or not having **heart disease** based on results from cardiac catheterizations - the gold standard in heart health assessment.



```
# 'data.frame':   270 obs. of  15 variables:
#  $ index                  : int  0 1 2 3 4 5 6 7 8 9 ...
#  $ Age                    : int  70 67 57 64 74 65 56 59 60 63 ...
#  $ Sex                    : int  1 0 1 1 0 1 1 1 1 0 ...
#  $ Chest.pain.type        : int  4 3 2 4 2 4 3 4 4 4 ...
#  $ BP                     : int  130 115 124 128 120 120 130 110 140 150 ...
#  $ Cholesterol            : int  322 564 261 263 269 177 256 239 293 407 ...
#  $ FBS.over.120           : int  0 0 0 0 0 0 1 0 0 0 ...
#  $ EKG.results            : int  2 2 0 0 2 0 2 2 2 2 ...
#  $ Max.HR                 : int  109 160 141 105 121 140 142 142 170 154 ...
#  $ Exercise.angina        : int  0 0 0 1 1 0 1 1 0 0 ...
#  $ ST.depression          : num  2.4 1.6 0.3 0.2 0.2 0.4 0.6 1.2 1.2 4 ...
#  $ Slope.of.ST            : int  2 2 1 2 1 1 2 2 2 2 ...
#  $ Number.of.vessels.fluro: int  3 0 0 1 1 0 1 1 2 3 ...
#  $ Thallium               : int  3 7 7 7 3 7 6 7 7 7 ...
#  $ Heart.Disease          : chr  "Presence" "Absence" "Presence" "Absence" ...
```

https://www.kaggle.com/datasets/thedevastator/predicting-heart-disease-risk-using-clinical-var

DESCRIBE DATASET

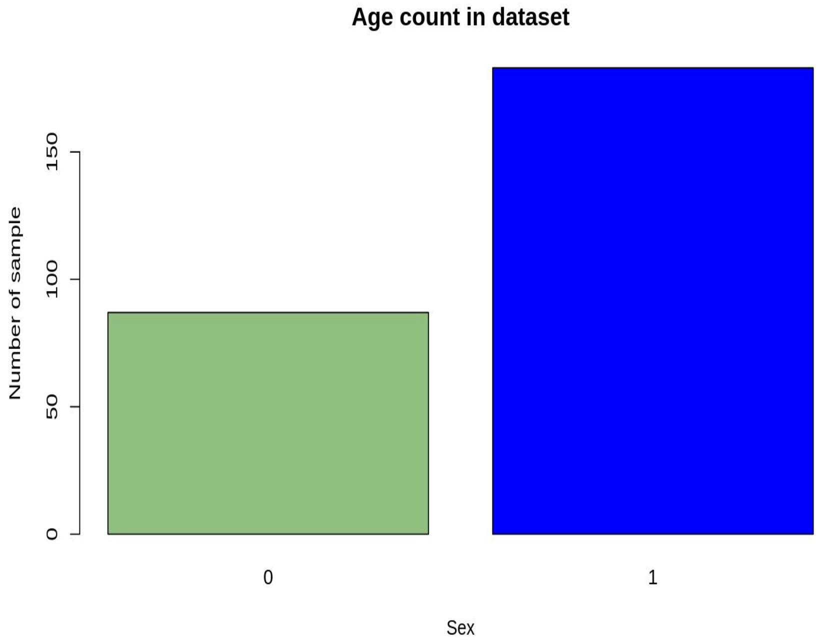


Column name	Description
Age	The age of the patient. (Numeric)
Sex	The gender of the patient. (Categorical)
Chest pain type	The type of chest pain experienced by the patient. (Categorical)
BP	The blood pressure level of the patient. (Numeric)
Cholesterol	The cholesterol level of the patient. (Numeric)
FBS over 120	The fasting blood sugar test results over 120 mg/dl. (Numeric)
EKG results	The electrocardiogram results of the patient. (Categorical)
Max HR	The maximum heart rate levels achieved during exercise testing. (Numeric)
Exercise angina	The angina experienced during exercise testing. (Categorical)
ST depression	The ST depression on an Electrocardiogram. (Numeric)
Slope of ST	The slope of ST segment electrocardiogram readings. (Categorical)
Number of vessels fluro	The amount vessels seen in Fluoroscopy images. (Numeric)
Thallium	The Thallium Stress test findings. (Categorical)
Heart Disease	Whether or not the patient has been diagnosed with Heart Disease. (Categorical)

Barplot

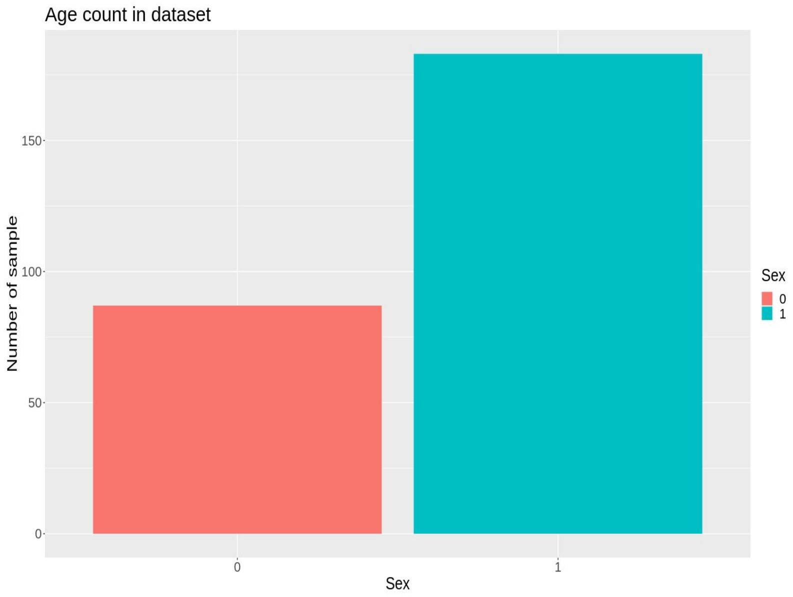
Basic Plot

```
barplot(table(data$Sex),
   ylab = "Number of sample", # add y label
   xlab = "Sex", # add x label
   main = "Age count in dataset", # add title for graph
   col = c("#90bf80", "blue"), # add color
   fill = c("pink")
)
```

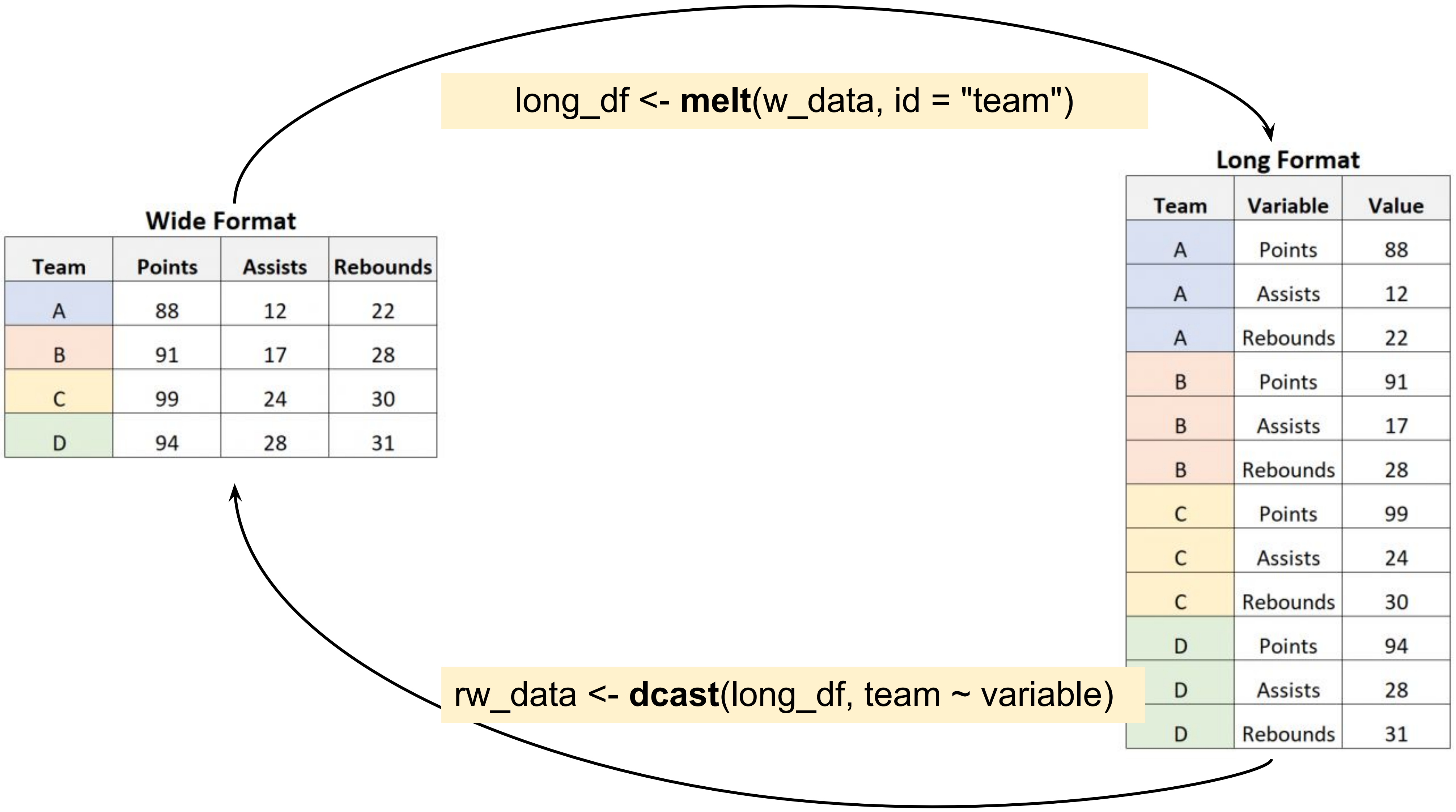


ggplot2 Plot

```
ggplot(data, aes(as.factor(Sex), fill = as.factor(Sex))) +
   geom_bar() + # select a type of plot
   labs(x = "Sex", y = "Number of sample", title = "Age count in dataset") + # add title, x axis's name, y axis'name
   guides(fill = guide_legend(title = "Sex")) +
   theme(text = element_text(size = 20))
```



Reshape2::melt,dcast



```
long_df <- melt(w_data, id = "team")
```

Long Format

Team	Variable	Value
A	Points	88
A	Assists	12
A	Rebounds	22
B	Points	91
B	Assists	17
B	Rebounds	28
C	Points	99
C	Assists	24
C	Rebounds	30
D	Points	94
D	Assists	28
D	Rebounds	31

Wide Format

Team	Points	Assists	Rebounds
A	88	12	22
B	91	17	28
C	99	24	30
D	94	28	31

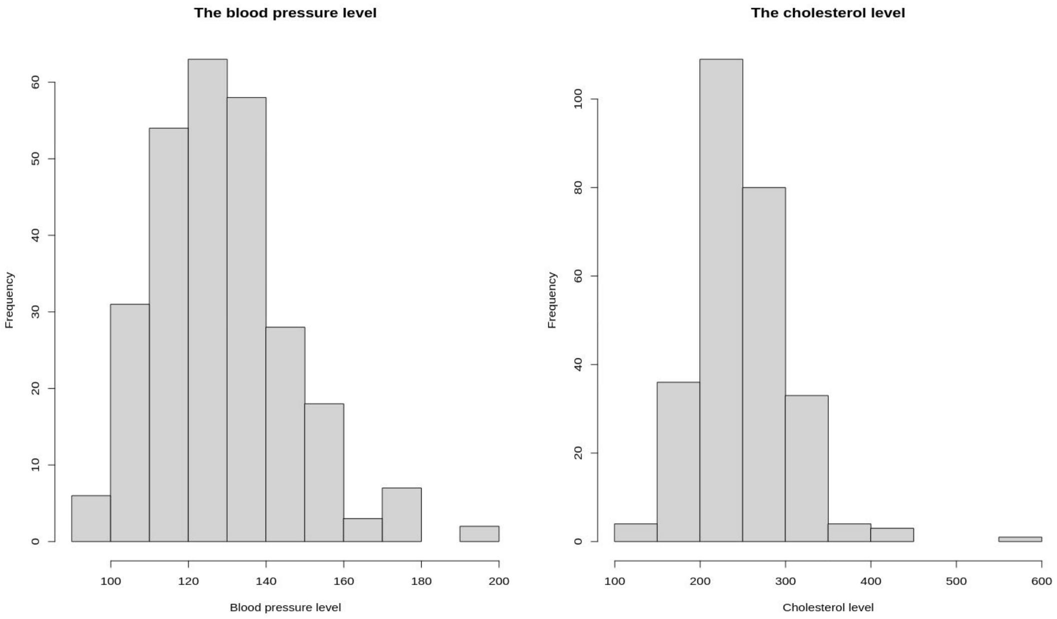
```
rw_data <- dcast(long_df, team ~ variable)
```

https://ademos.people.uic.edu/Chapter8.html

Histograms

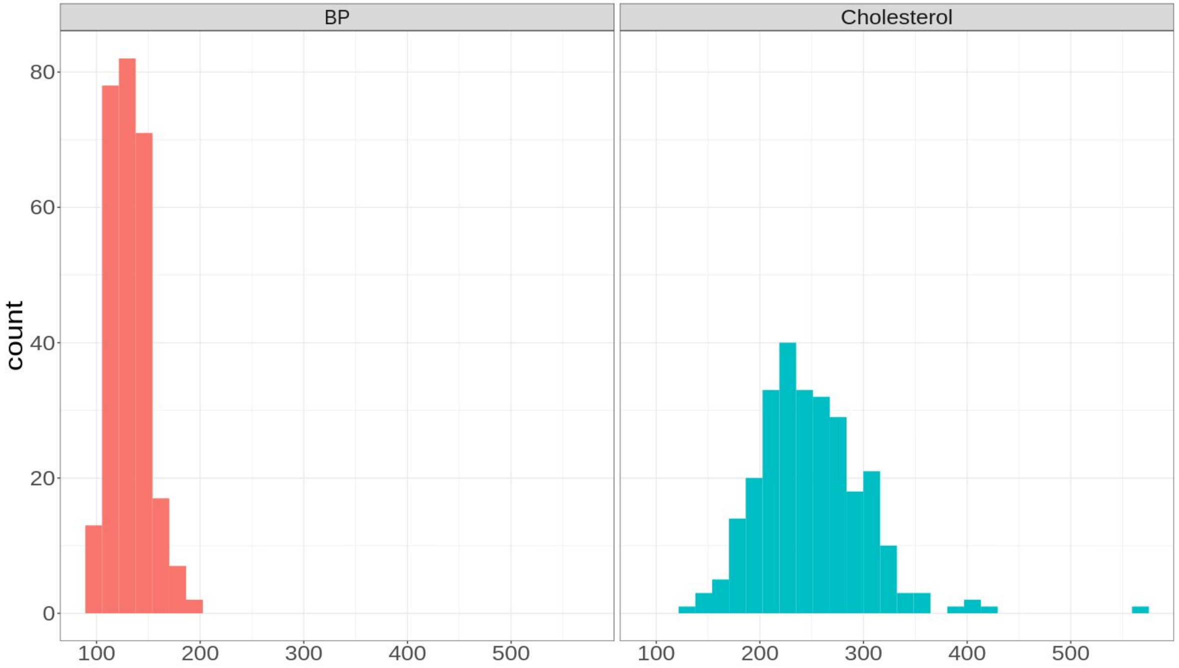
Basic Plot

```
par(mfrow = c(1, 2))
hist(data$BP,
   main = "The blood pressure level",
   xlab = "Blood pressure level"
)
hist(data$Cholesterol,
   main = "The cholesterol level",
   xlab = "Cholesterol level"
)
```



ggplot2 Plot

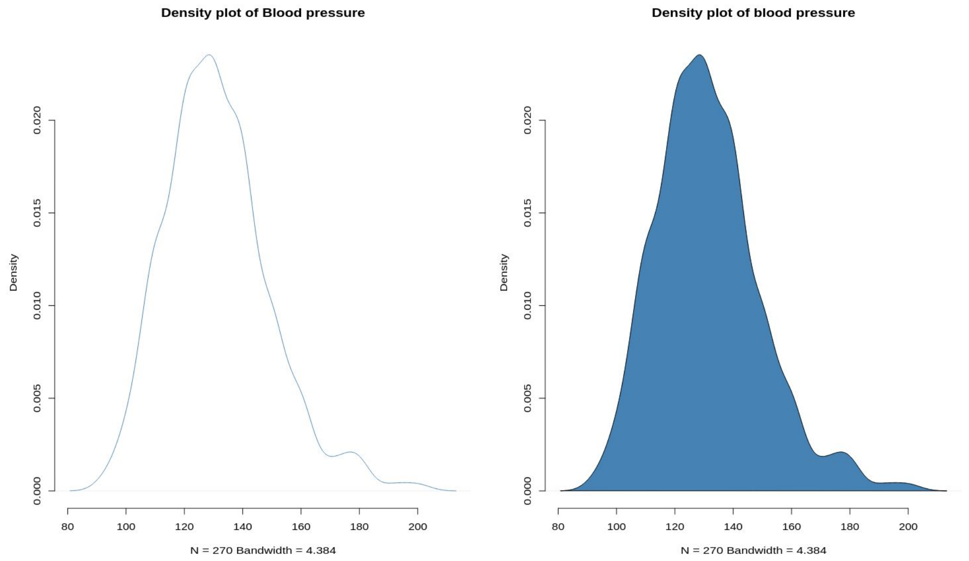
```
ggplot(mdata_sub, aes(value, fill = variable)) +
   theme_bw() +
   geom_histogram() +
   facet_grid(~variable) +
   theme(legend.position = "NULL") +
   labs(x = "") +
   theme(text = element_text(size = 25))
```



Density plot

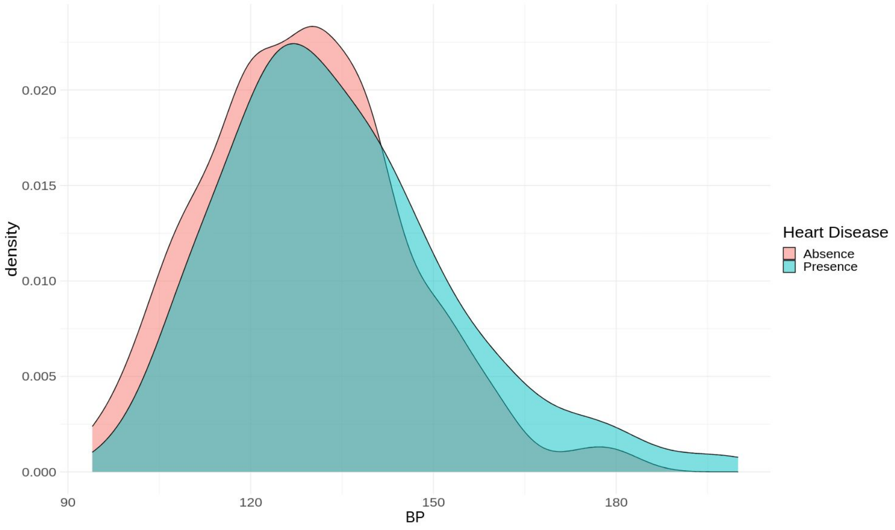
Basic Plot

```
### Compute the density data
dens <- density(data$BP)
par(mfrow = c(1, 2))
### plot density
plot(dens, frame = FALSE, col = "steelblue",
   main = "Density plot of Blood pressure"
)
### Fill the density plot using polygon()
plot(dens, frame = FALSE, col = "steelblue",
   main = "Density plot of blood pressure"
)
polygon(dens, col = "steelblue")
```

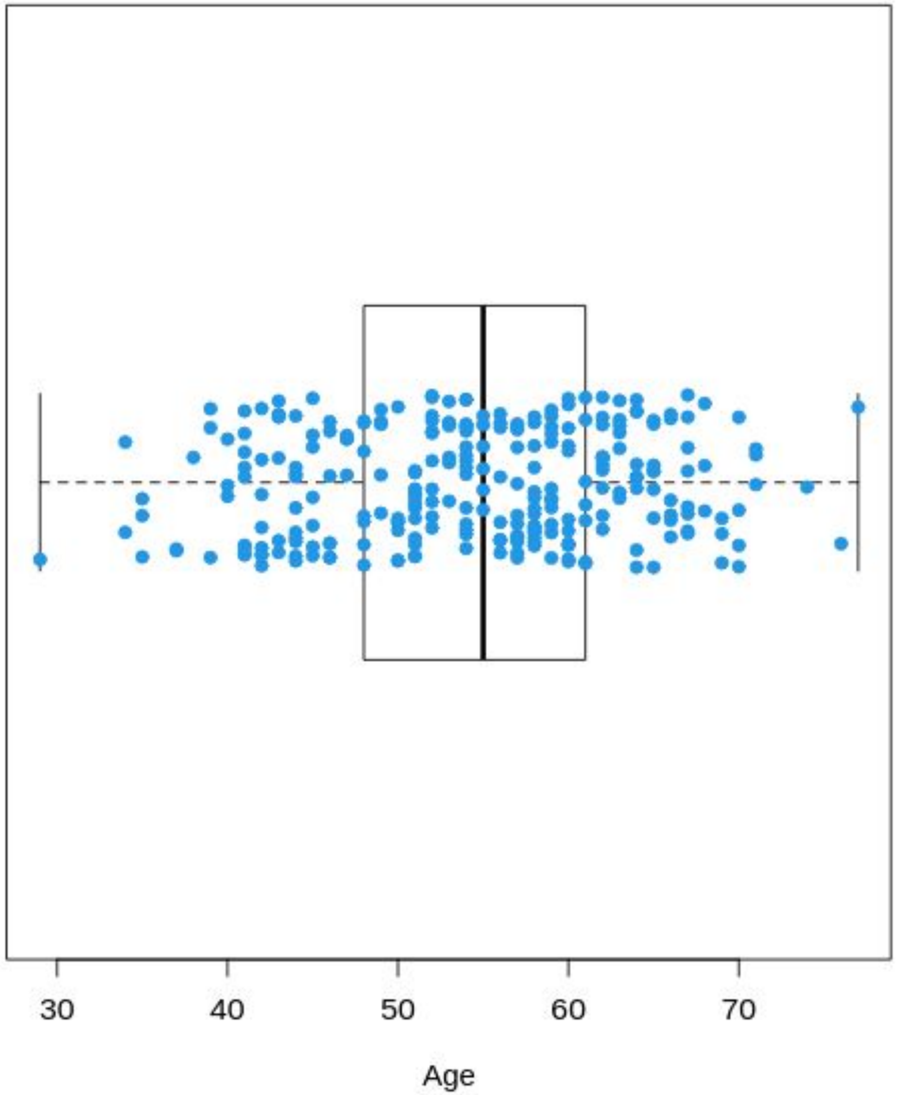


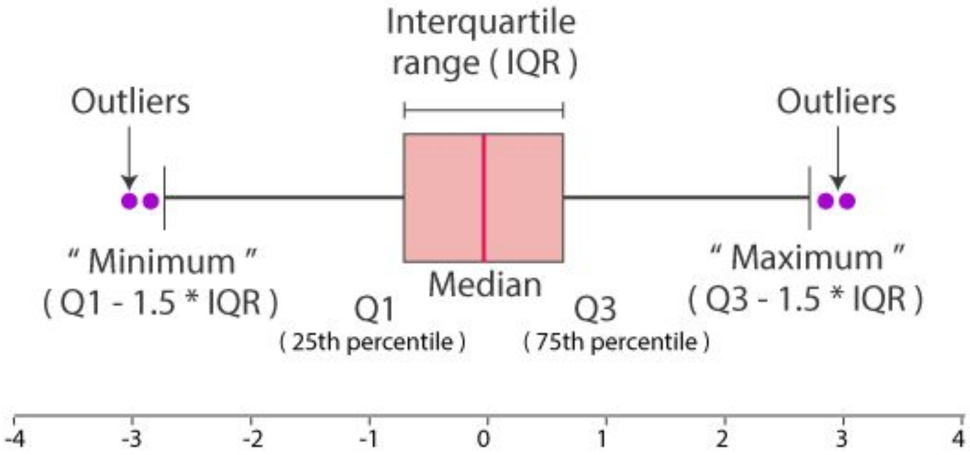
ggplot2 Plot

```
ggplot(data, aes(BP, fill = Heart.Disease)) +
   geom_density(alpha = 0.5) +
   theme_minimal() +
   theme(text = element_text(size = 20)) +
   guides(fill = guide_legend(title = "Heart Disease"))
```



Box and violin plots





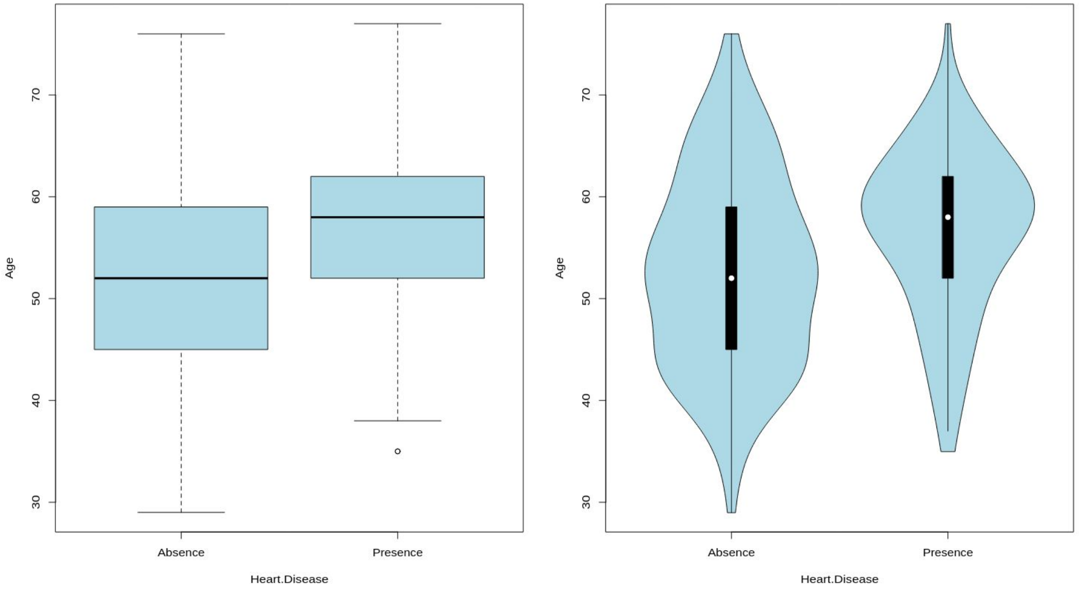
Different parts of boxplot

© Byjus.com

Box and violin plots

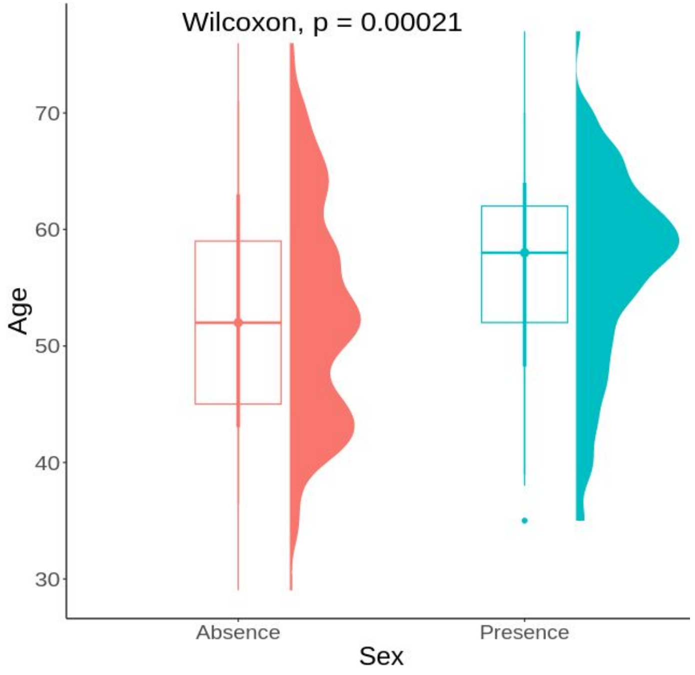
Basic Plot

```
par(mfrow = c(1, 2))
boxplot(Age ~ Heart.Disease, data = data, col = "lightblue")
vioplot(Age ~ Heart.Disease, data = data, col = "lightblue")
dev.off()
```



ggplot2 Plot

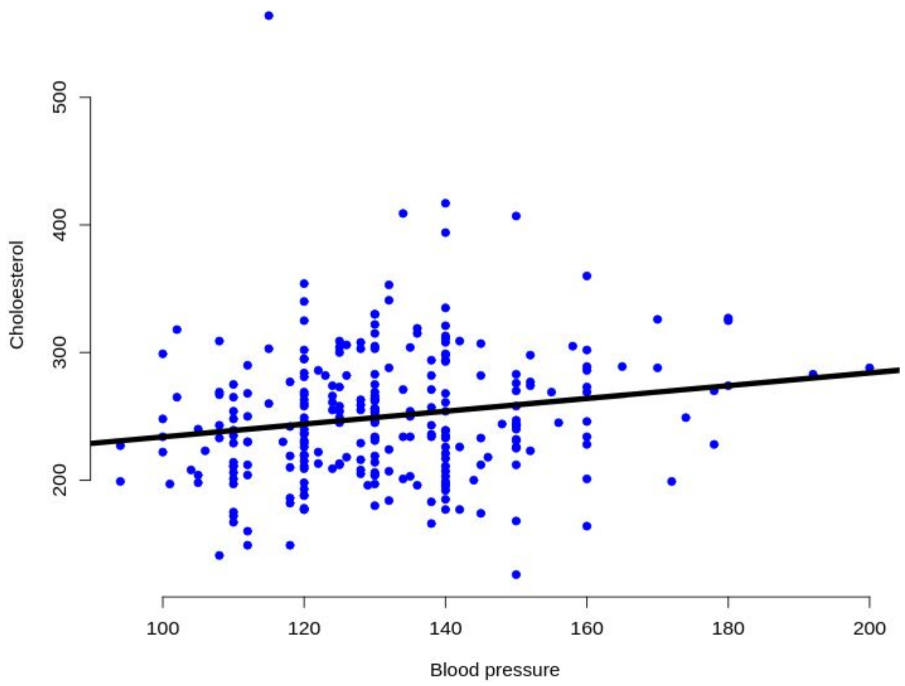
```
ggplot(data, aes(x = Heart.Disease, y = Age, col = Heart.Disease)) +
   geom_boxplot(width = 0.3) +
   stat_compare_means(size = 7) +
   theme_classic() +
   labs(x = "Sex") +
   theme(legend.position = "NULL") +
   theme(text = element_text(size = 20)) +
   ggdist::stat_halfeye(aes(fill = Heart.Disease),
                  adjust = 0.5, justification = -0.5, width = 0.4)
```



Scatter Plots add text

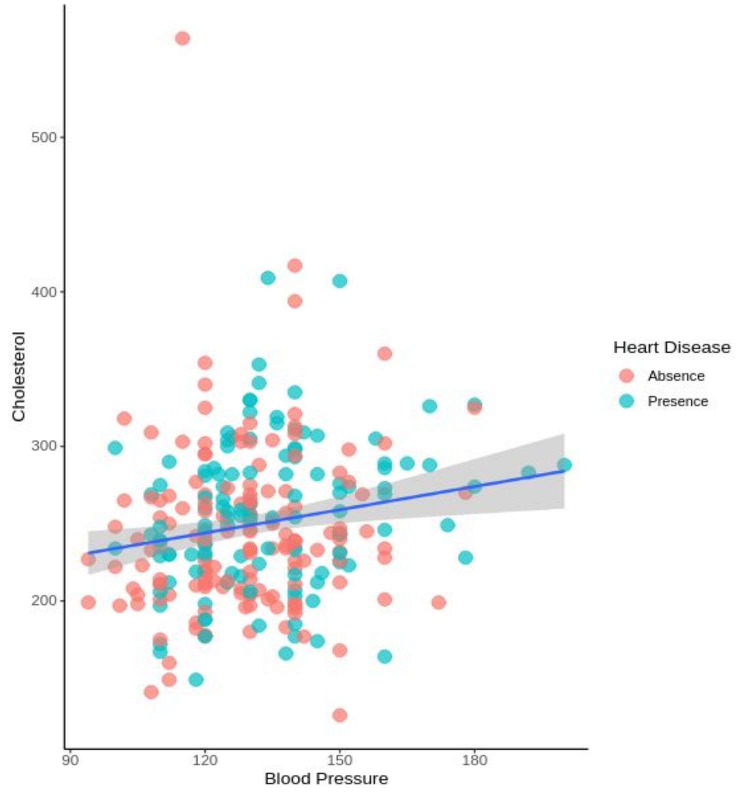
Basic Plot

```
x <- data$BP
y <- data$Cholesterol
plot(x, y,
   xlab = "Blood pressure",
   ylab = "Choloesterol",
   pch = 19, frame = FALSE, col = "blue")
abline(lm(y ~ x, data = data),
   col = "black", lwd = 5, lty = 1)
```



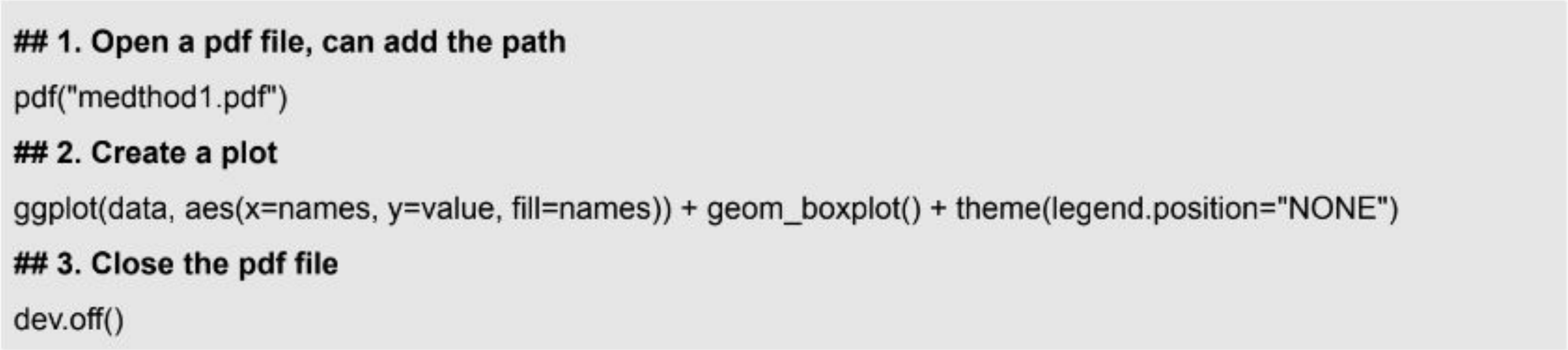
ggplot2 Plot

```
ggplot(data, aes(x = BP, y = Cholesterol)) +
   geom_point(aes(col = Heart.Disease), size = 4, alpha = 0.7) +
   geom_smooth(method = "lm") +
   theme(text = element_text(size = 15)) +
   theme_classic() +
   labs(x = "Blood Pressure", col = "Heart Disease")
```



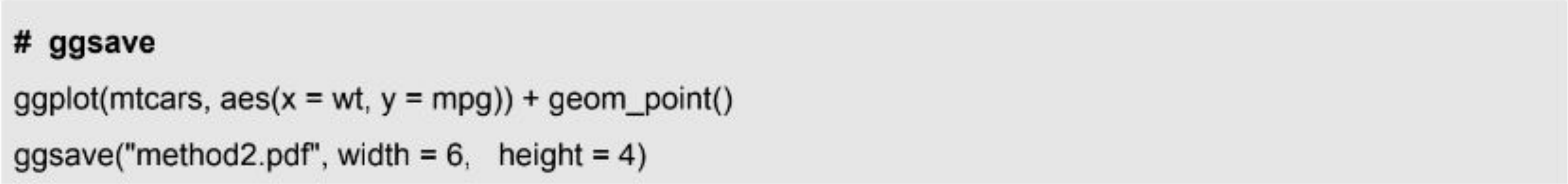
Save the plot

Method 1



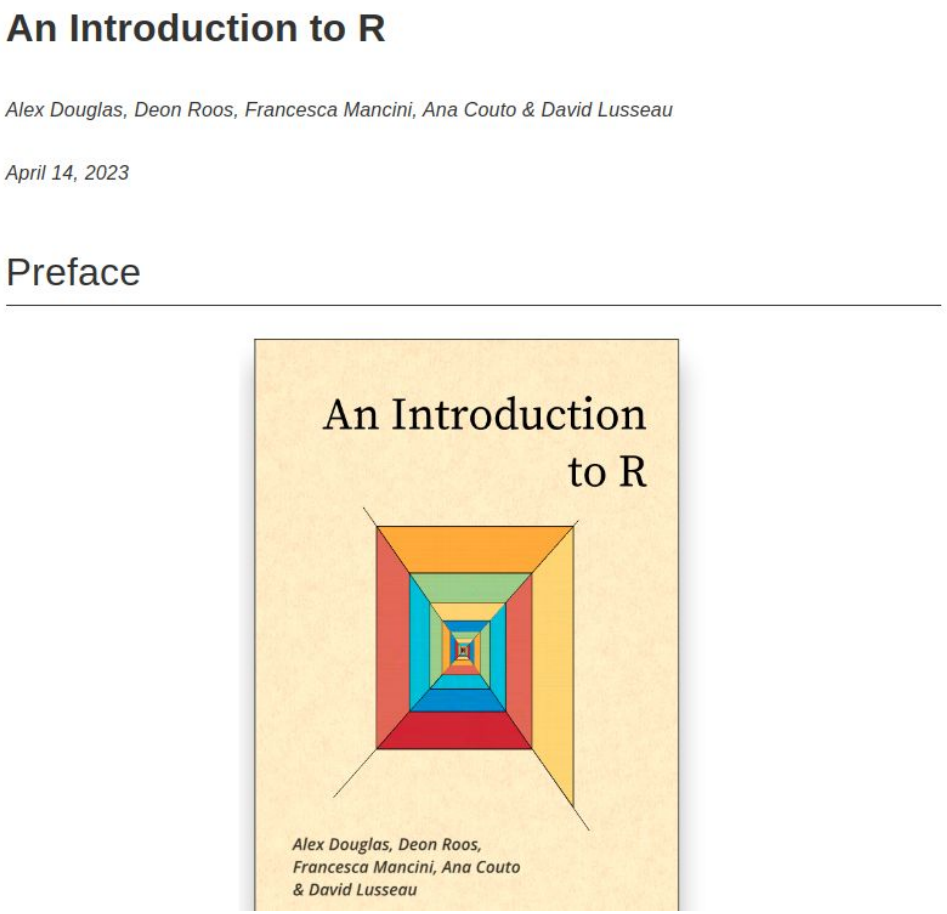
```
## 1. Open a pdf file, can add the path
pdf("medthod1.pdf")
## 2. Create a plot
ggplot(data, aes(x=names, y=value, fill=names)) + geom_boxplot() + theme(legend.position="NONE")
## 3. Close the pdf file
dev.off()
```

Method 2



```
#  ggsave
ggplot(mtcars, aes(x = wt, y = mpg)) + geom_point()
ggsave("method2.pdf", width = 6,   height = 4)
```

eBOOK



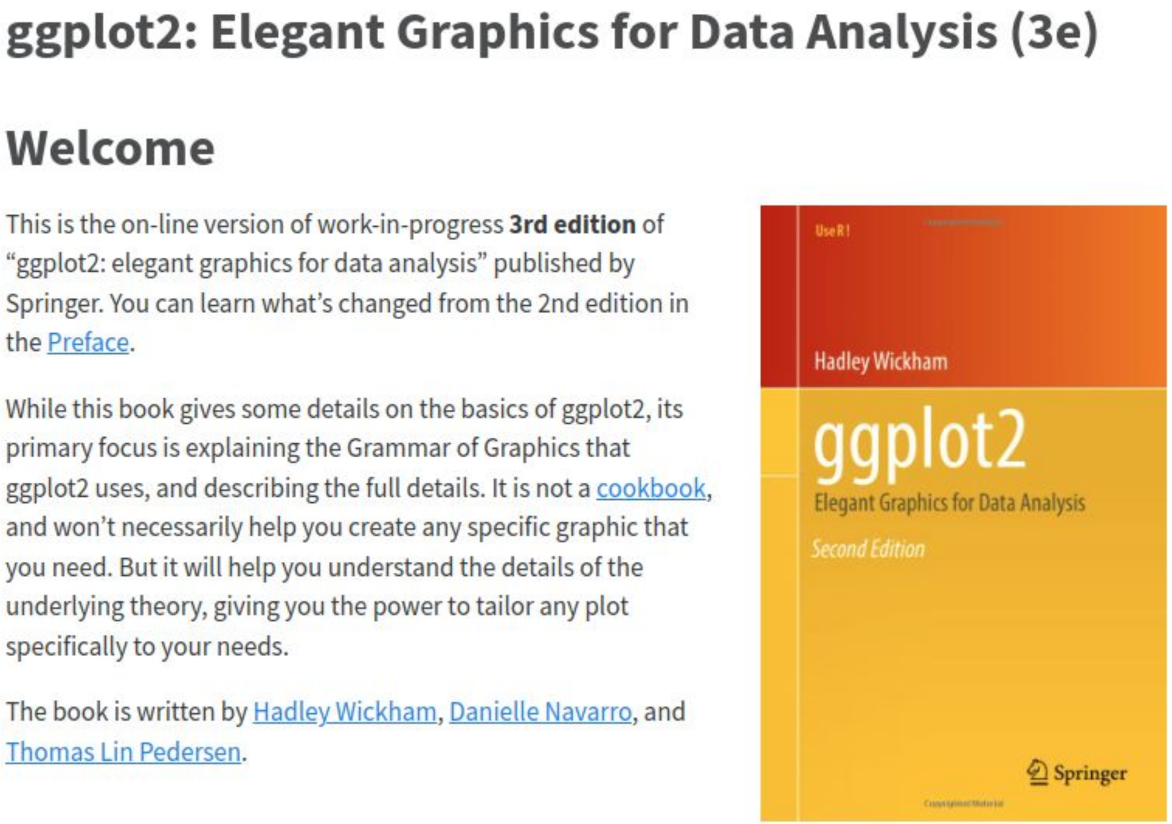
An Introduction to R

Alex Douglas, Deon Roos, Francesca Mancini, Ana Couto & David Lusseau

April 14, 2023

Preface

https://intro2r.com/



ggplot2: Elegant Graphics for Data Analysis (3e)

Welcome

This is the on-line version of work-in-progress **3rd edition** of "ggplot2: elegant graphics for data analysis" published by Springer. You can learn what's changed from the 2nd edition in the Preface.

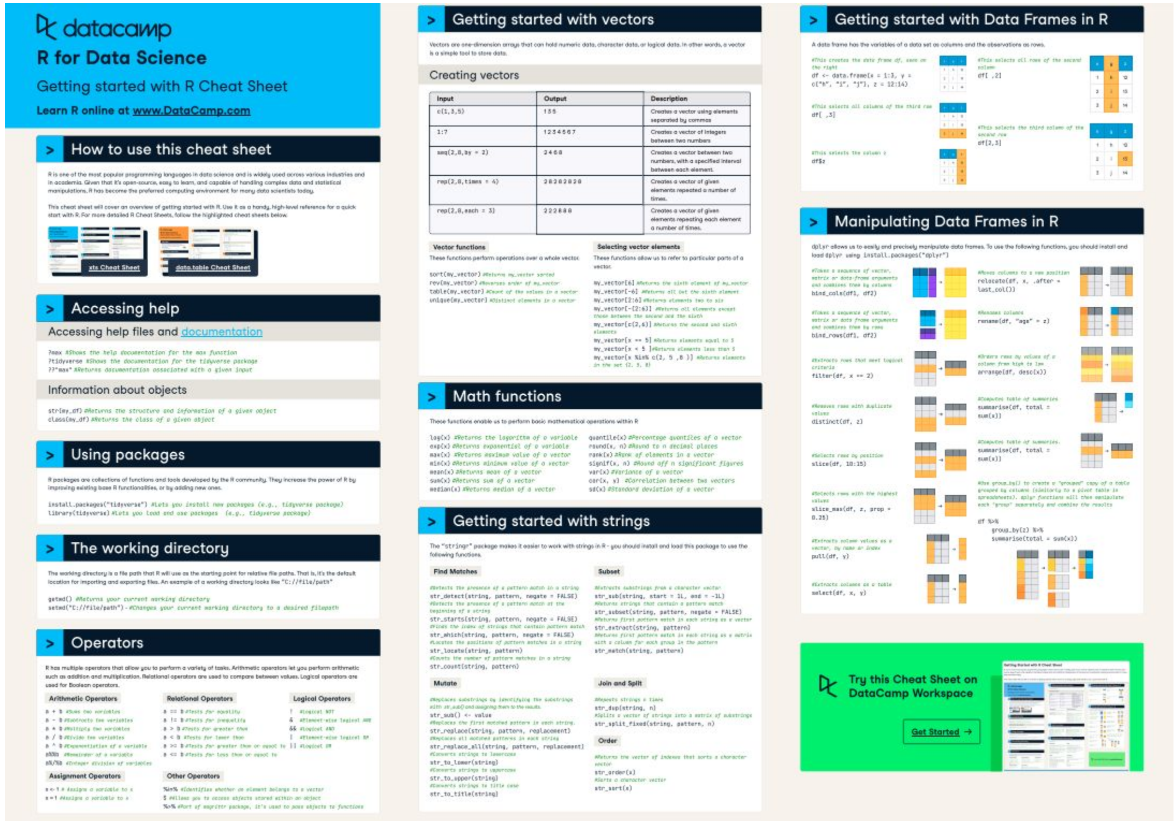
While this book gives some details on the basics of ggplot2, its primary focus is explaining the Grammar of Graphics that ggplot2 uses, and describing the full details. It is not a cookbook, and won't necessarily help you create any specific graphic that you need. But it will help you understand the details of the underlying theory, giving you the power to tailor any plot specifically to your needs.

The book is written by Hadley Wickham, Danielle Navarro, and Thomas Lin Pedersen.

https://ggplot2-book.org/

Cheetsheet

Cheat Sheet for R



datacamp

R for Data Science

Getting started with R Cheat Sheet

Learn R online at www.DataCamp.com

> How to use this cheat sheet

> Accessing help

Accessing help files and documentation

Information about objects

> Using packages

> The working directory

> Operators

> Getting started with vectors

Creating vectors

Input	Output	Description
c(1,3,5)	1 3 5	Creates a vector using elements separated by commas
1:7	1 2 3 4 5 6 7	Creates a vector of integers between two numbers
seq(2,8,by = 2)	2 4 6 8	Creates a vector between two numbers, with a specified interval between each element.
rep(2,8,times = 4)	2 8 2 8 2 8 2 8	Creates a vector of given elements repeated a number of times.
rep(2,8,each = 3)	2 2 2 8 8 8	Creates a vector of given elements repeating each element a number of times.

Vector functions

Selecting vector elements

> Math functions

> Getting started with strings

> Getting started with Data Frames in R

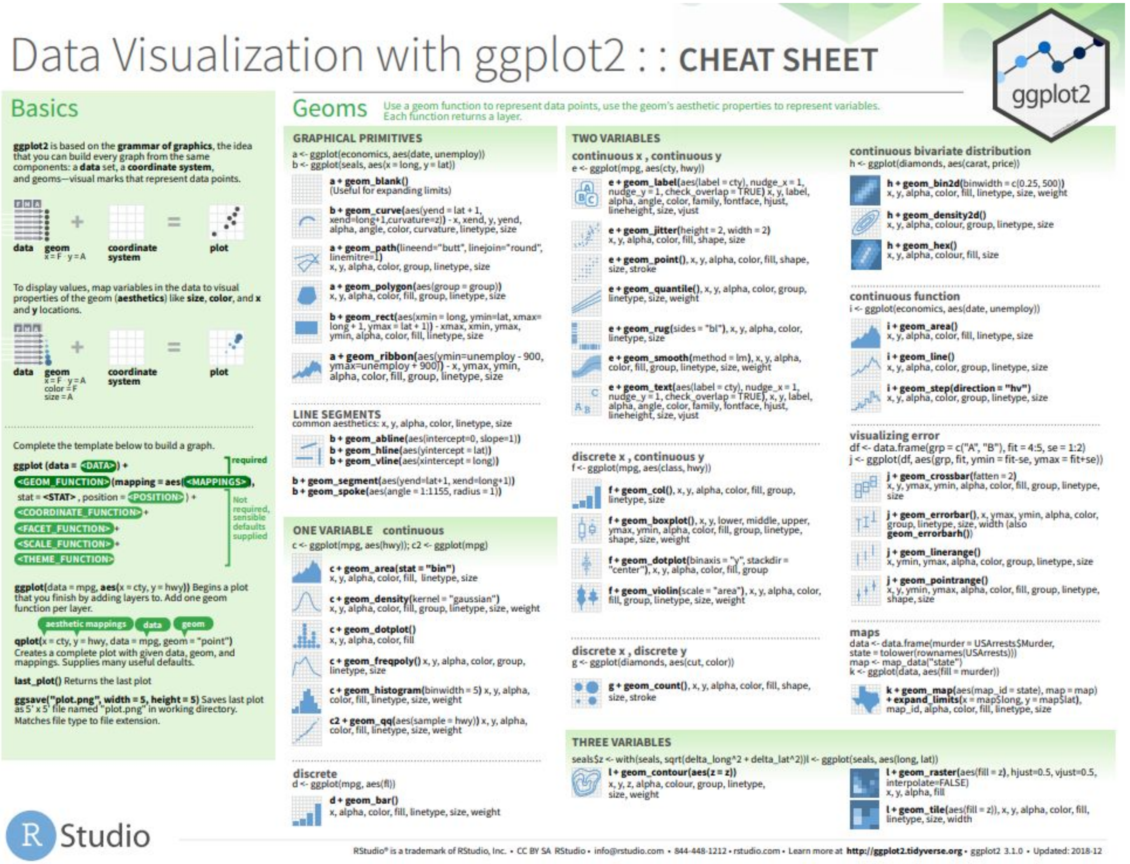
> Manipulating Data Frames in R

Try this Cheat Sheet on DataCamp Workspace

Get Started →

https://images.datacamp.com/image/upload/v1654763044/Marketing/Blog/R_Cheat_Sheet.pdf

Cheat Sheet for ggplot2



Data Visualization with ggplot2 : : CHEAT SHEET

ggplot2

Basics

ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same components: a data set, a coordinate system, and geoms—visual marks that represent data points.

To display values, map variables in the data to visual properties of the geom (aesthetics) like size, color, and x and y locations.

Complete the template below to build a graph.

ggplot (data = <DATA>) +
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
stat = <STAT>, position = <POSITION>) +
<COORDINATE_FUNCTION> +
<FACET_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>

required

Not required, sensible defaults supplied

ggplot(data = mpg, aes(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

qplot(x = cty, y = hwy, data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot() Returns the last plot

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Geoms — Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

LINE SEGMENTS

ONE VARIABLE continuous

discrete

TWO VARIABLES

continuous x , continuous y

discrete x , continuous y

discrete x , discrete y

continuous bivariate distribution

continuous function

visualizing error

maps

THREE VARIABLES

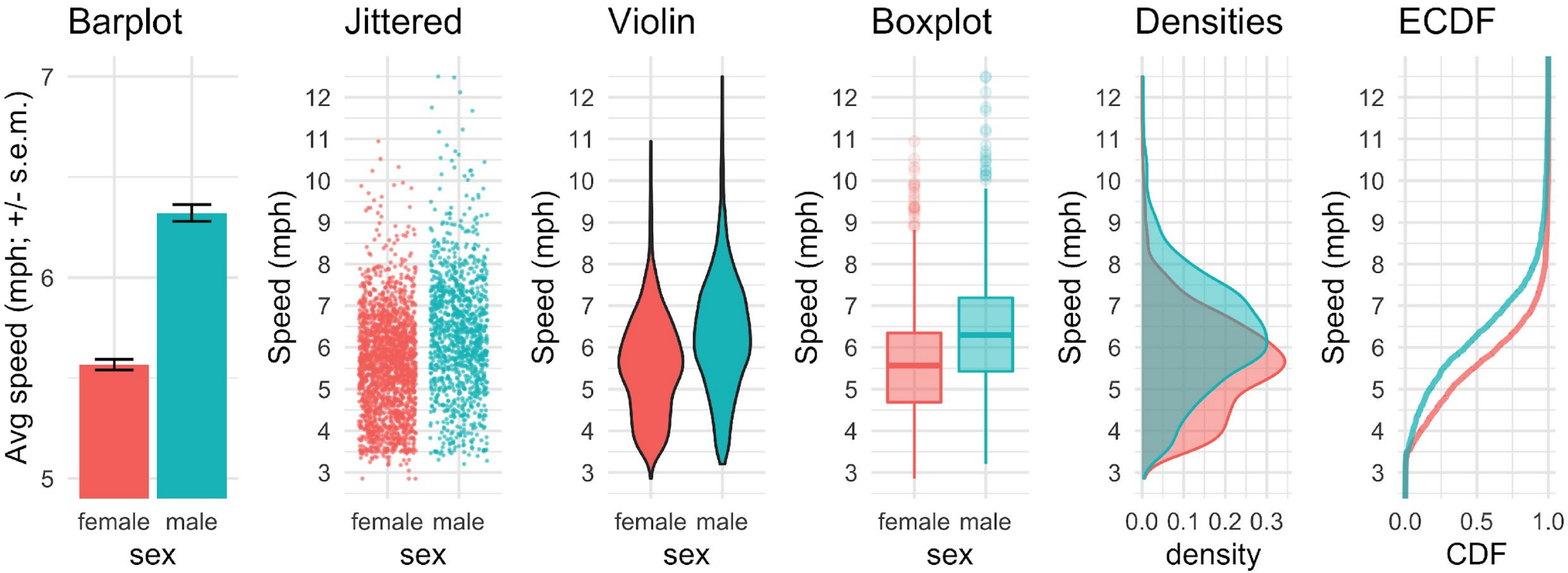
R Studio

RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at http://ggplot2.tidyverse.org • ggplot2 3.1.0 • Updated: 2018-12

https://www.maths.usyd.edu.au/u/UG/SM/STAT3022/r/current/Misc/data-visualization-2.1.pdf

Thank you

Homework


Barplot
Avg speed (mph; +/- s.e.m.)
5
6
7
female
male
sex
Jittered
Speed (mph)
3
4
5
6
7
8
9
10
11
12
female
male
sex
Violin
Speed (mph)
female
male
sex
Boxplot
Speed (mph)
female
male
sex
Densities
Speed (mph)
0.0
0.1
0.2
0.3
density
ECDF
Speed (mph)
0.0
0.5
1.0
CDF