



United Nations  
Educational, Scientific and  
Cultural Organization



AN INTERNATIONAL AWARD-WINNING INSTITUTION FOR SUSTAINABILITY



- UNESCO Chair on Future Studies
  - Anticipation for Sustainability and Well-being

# Basic Biostatistics with R

## – The Practical –

---

Prof. Jamalludin Ab Rahman, MD MPH  
Department of Community Medicine,  
Kulliyyah Of Medicine

# What is R

- Open-source programming language environment
- Used especially for **statistical** computing and **graphics**
- Free
- *Steep learning curve*
- Progressive

# Install R & Rstudio

---

1. For R, download from <https://cran.r-project.org>
2. Then also download RStudio from <https://rstudio.com>

# Repository for this workshop

- <https://github.com/profjama/l/biostatistics>

The screenshot shows a GitHub repository page for the user 'profjamal' with the repository name 'biostatistics'. The repository is public and has 1 branch and 0 tags. The 'Code' tab is selected. A recent commit by 'profjamal' titled 'Add files via upload' is shown, along with several other files: 'Basic Biostatistics 2022.pdf', 'Biostatistics-with-R.pdf', 'README.md', 'healthstatus.csv', and 'multivariate.csv'. The README file contains the text: 'This is the repository for notes, scripts and data used for Biostatistics workshop.'

profjamal / biostatistics Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags Go to file Add file Code

profjamal Add files via upload e397c67 3 hours ago 24 commits

Basic Biostatistics 2022.pdf Add files via upload 8 hours ago

Biostatistics-with-R.pdf Add files via upload 3 hours ago

README.md Create README.md 15 months ago

healthstatus.csv Update healthstatus.csv 15 months ago

multivariate.csv Add files via upload 14 months ago

README.md

This is the repository for notes, scripts and data used for Biostatistics workshop.

The screenshot shows the RStudio interface with the title bar "Introduction to R - RStudio".

**Console pane:**

```
R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

**Global Environment pane:**

Environment is empty

**Files pane:**

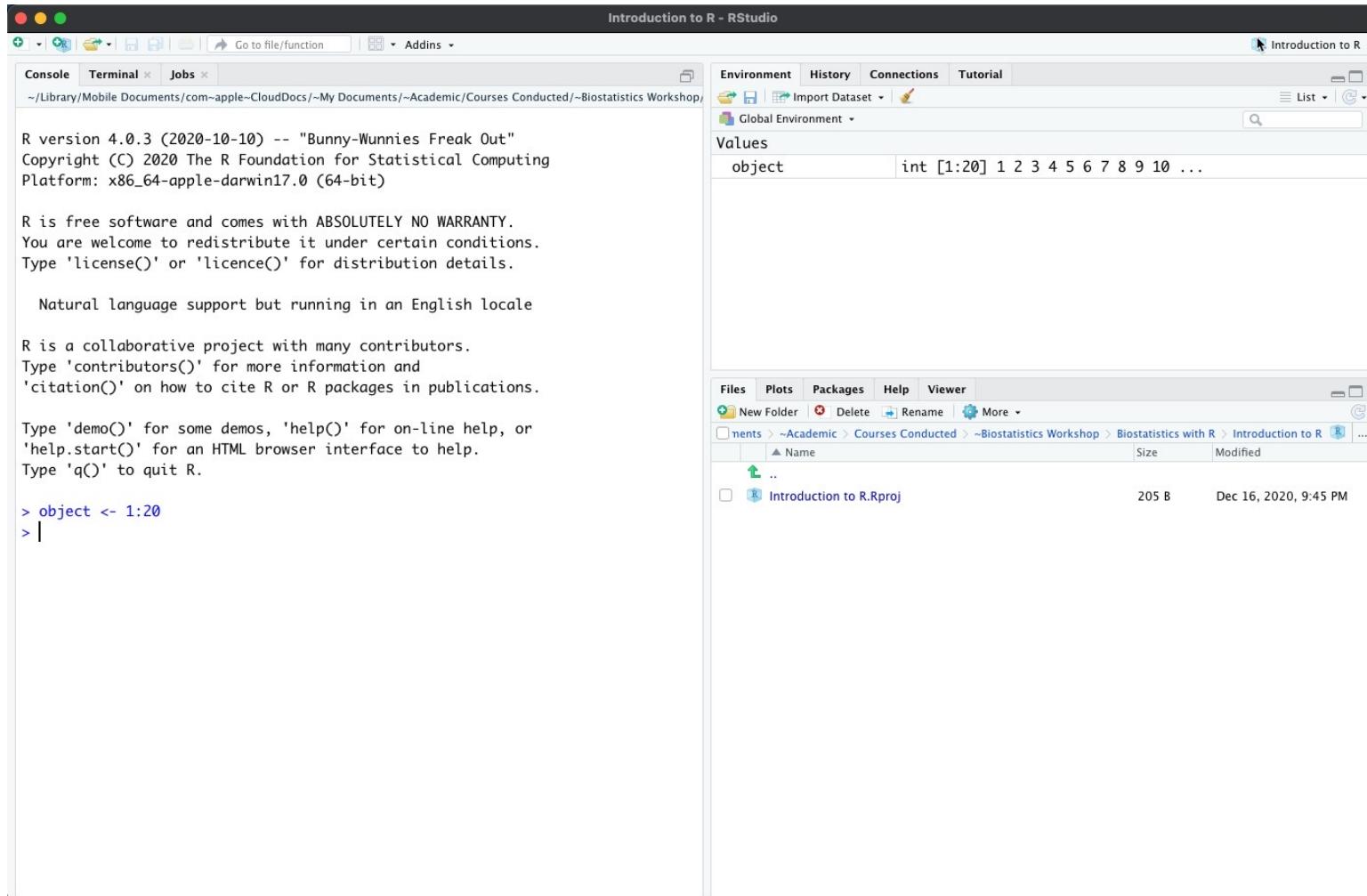
The objects will appear here

Name	Size	Modified
..		
Introduction to R.Rproj	205 B	Dec 16, 2020, 9:45 PM

Type your command here

The graphics, packages, file directory

# R command structure



# Data types (for each variable)

1. Character - similar like text in Excel
2. Numeric - any numbers including decimals
3. Integer - only accept whole number
4. Logical - TRUE, FALSE or NA
5. Complex - combination of data type
6. Date

# Data structure (collection of variables)

1. Atomic vector
2. List
3. Matrix
4. Data frame
5. Factors

A close-up photograph of a stack of papers. A silver paperclip is visible on the left side. A green tab or piece of tape is attached to the stack. The background is a plain, light color.

# Starting R programming

- Google is your friend
- Set the working directory
- Packages can make or break you
- Practice, practice, practice

# Recommended packages

- tidyverse, which contain
  1. dplyr
  2. ggplot2
  3. readr
  4. forcats
  5. stringr
  6. purrr
  7. tidyr
  8. tibble
- ggpubr
- gtsummary
- psych
- car



# Constant need to search for guide

- Read the package documentation
- <https://www.rdocumentation.org>
- Can download the cheat sheet
- <https://www.rstudio.com/resources/cheatsheets/>

**Data visualization with ggplot2 :: CHEAT SHEET**



The cheat sheet is organized into several sections:

- Basics**: Explains the grammar of graphics: data + geom = plot.
- Geoms**: A large section listing various geometric primitives with their R code examples and aesthetic mappings.
- Graphical Primitives**: A subset of Geoms.
- Two Variables**: Both continuous, both discrete, one continuous, one discrete.
- Continuous Bivariate Distribution**: Continuous function, bivariate normal distribution.
- Maps**: Shows how to map data to geographic coordinates.
- Three Variables**: Examples for plots with three variables.

**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>+
  <FACTOR_FUNCTION>+
  <SCALE_FUNCTION>+
  <THEME_FUNCTION>
```

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

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.

**Aes** Common aesthetic values.

color and fill - string ("red", "#RRGGBB")  
 linetype - integer or string (0 = "blank", 1 = "solid", 2 = "dashed", 3 = "dotted", 4 = "dotteddash", 5 = "longdash", 6 = "twodash")  
 lineend - string ("round", "butt", or "square")  
 linejoin - string ("round", "mitre", or "bevel")  
 size - integer (line width in mm)  
 shape - integer/shape name or a single character ("a")

d <- ggplot(mpg, aes(f))  
 d + geom\_bar()

**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**

- a + geom\_blank() and a + expand\_limits()  
 Ensure limits include values across all plots.
- b + geom\_curve(aes(yend = lat + 1, xend = long + 1), curvature = 1) - x, y, end, yend, alpha, angle, color, curvature, linetype, size
- a + geom\_path(linewidth = "butt", linejoin = "round", linemiter = 1)
- x, y, alpha, color, group, linetype, size
- a + geom\_polygon(aes(alpha = 0.5)) - x, y, alpha, color, fill, group, subgroup, linetype, size
- b + geom\_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size
- a + geom\_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)) - x, ymax, ymin, alpha, color, fill, group, linetype, size
- e + geom\_text(label = "cty", nudge\_x = 1, nudge\_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust
- e + geom\_rug(sides = "bl") - x, y, alpha, color, linetype, size
- e + geom\_smooth(method = lm)
- x, y, alpha, color, fill, group, linetype, size, weight
- e + geom\_text(label = "cty", nudge\_x = 1, nudge\_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust
- i + geom\_area() - x, y, alpha, color, fill, linetype, size
- i + geom\_line() - x, y, alpha, color, group, linetype, size
- i + geom\_step(direction = "hv") - x, y, alpha, color, group, linetype, size

**TWO VARIABLES**

- both continuous
- e + geom\_label(ae(label = cty), nudge\_x = 1, nudge\_y = 1) - x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust
- e + geom\_point() - x, y, alpha, color, fill, shape, size, stroke
- e + geom\_quantile() - x, y, alpha, color, group, linetype, size, weight
- e + geom\_hex() - x, y, alpha, color, fill, size

**continuous function**

- i <- ggplot(economics, aes(date, unemploy))
- i + geom\_area() - x, y, alpha, color, fill, linetype, size
- i + geom\_line() - x, y, alpha, color, group, linetype, size
- i + geom\_step(direction = "hv") - x, y, alpha, color, group, linetype, size

**one discrete, one continuous**

- f + geom\_col() - x, y, alpha, color, fill, group, linetype, size
- f + geom\_boxplot() - x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight
- f + geom\_dotplot(binaxis = "y", stackdir = "center") - x, y, alpha, color, fill, group
- f + geom\_violin(scale = "area") - x, y, alpha, color, fill, group, linetype, size, weight

**one discrete**

- f + geom\_crossbar(fatten = 2) - x, y, ymax, ymin, alpha, color, group, linetype, size, width
- j + geom\_errorbar() - x, y, ymax, ymin, alpha, color, group, linetype, size, width
- j + geom\_linerange() - x, ymin, ymax, alpha, color, group, linetype, size
- j + geom\_pointrange() - x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

**maps**

- data <- data.frame(murder = USAreets\$Murder, state = tolower(rownames(USAreets)))
- map <- map\_data("state")
- k <- ggplot(data, aes(fill = murder))
- k + geom\_map(aes(map\_id = state), map = map)
- k + expand\_limits(x = map\$long, y = map\$lat)
- map\_id, alpha, color, fill, linetype, size

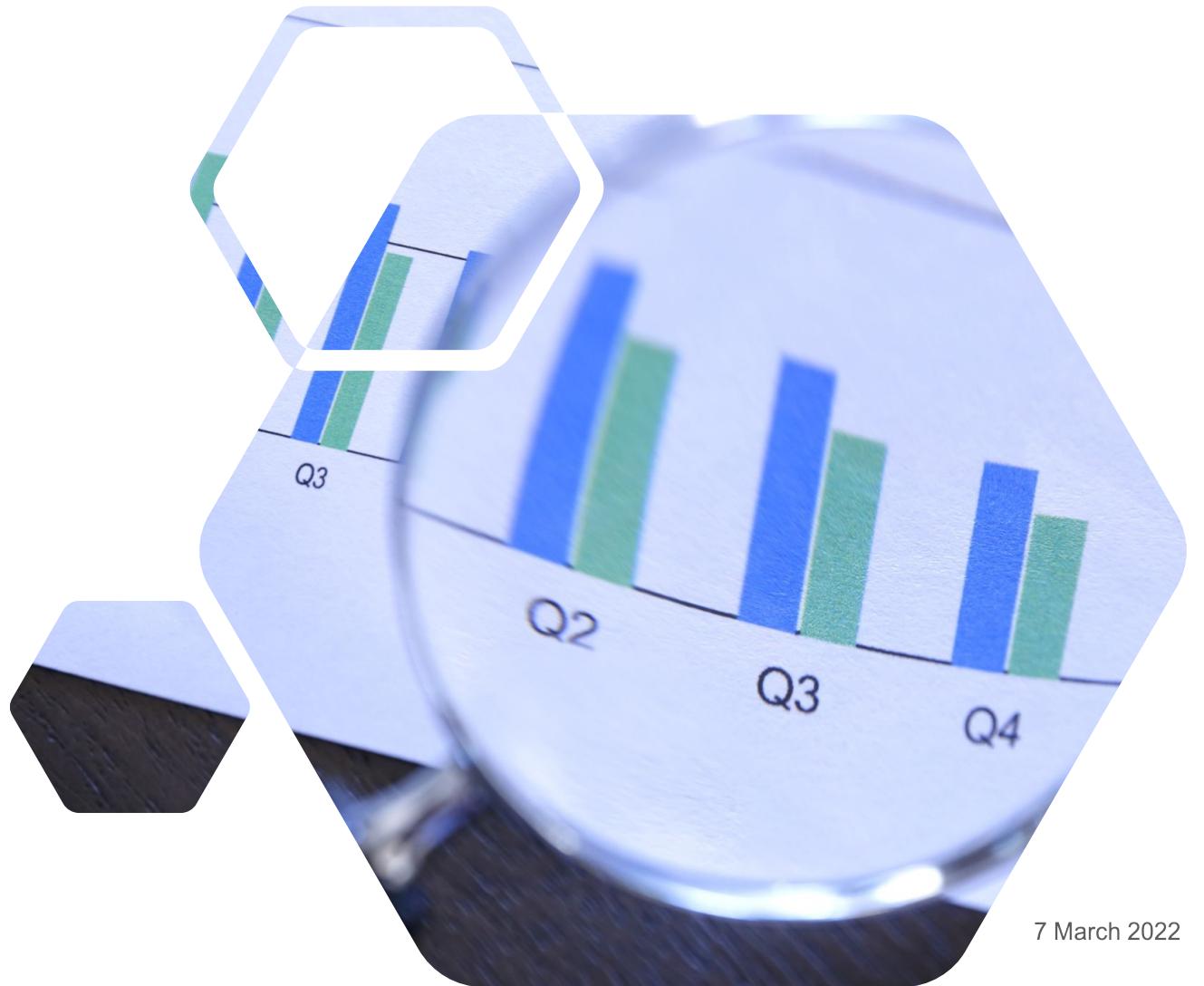
**three variables**

- l + geom\_contour(aes(z = z)) - x, y, z, alpha, color, group, linetype, size, weight
- vjust = 0.5, interpolate = FALSE)
- x, y, alpha, fill
- l + geom\_contour\_filled(aes(fill = z)) - x, y, alpha, color, fill, group, linetype, size, subgroup
- x, y, alpha, color, fill, linetype, size, width

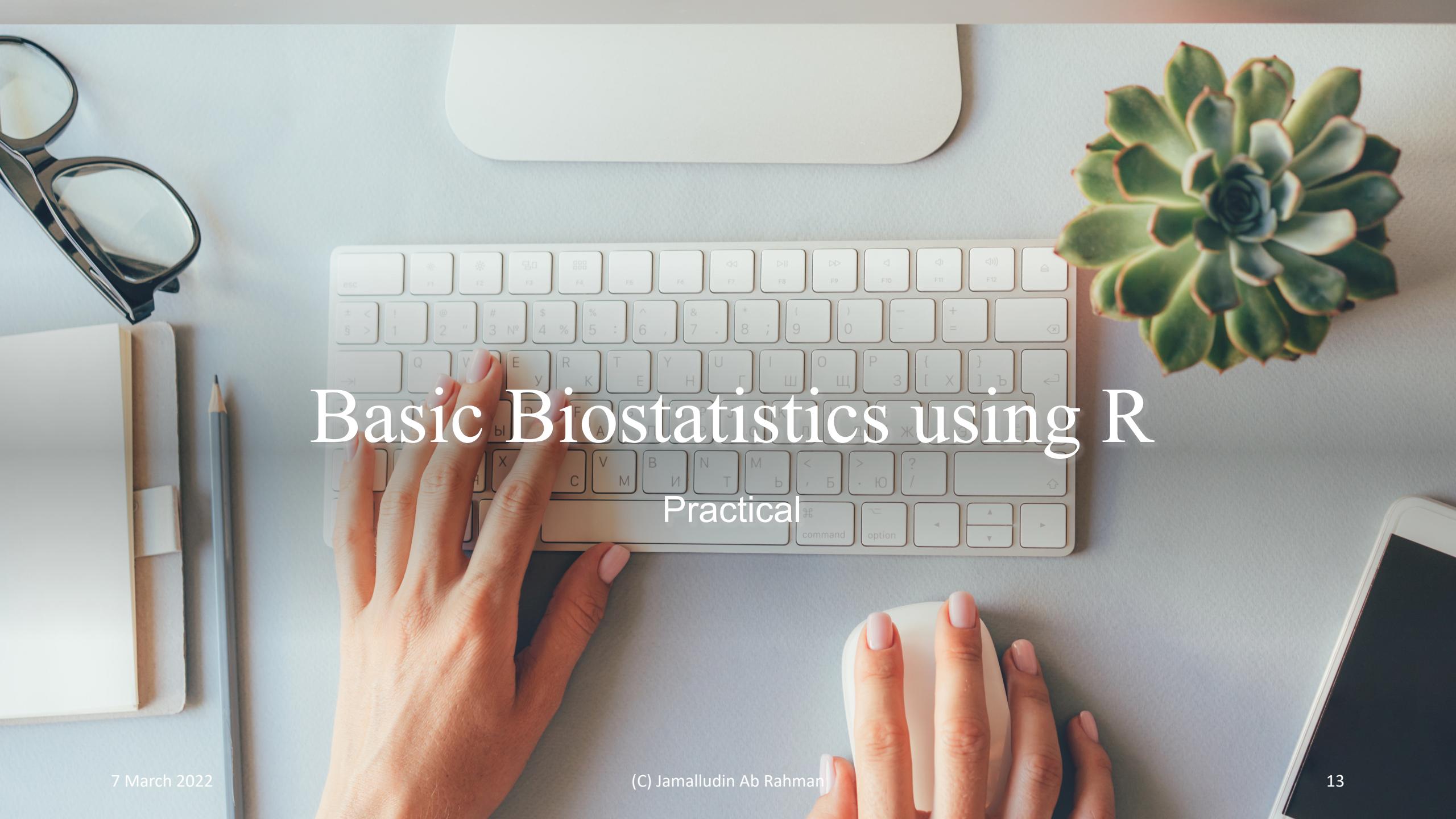
RStudio® is a trademark of RStudio, PBC • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at [ggplot2.tidyverse.org](https://ggplot2.tidyverse.org) • ggplot2 3.3.5 • Updated: 2021-08

# Steps in statistics

1. Getting of data – import, or create
2. Examine the data – for errors, distribution
3. Analyse – descriptive & analytical
4. Interpret & conclude
5. Report



7 March 2022



# Basic Biostatistics using R

Practical

# Content

1. Preparing the data – *create, read (import) and export data*
2. Checking the data
3. Descriptive statistics
4. Comparing proportions
5. Comparing two means
6. Comparing more than two means
7. Correlation analysis
8. Non-parametric tests
9. Comparing paired means



# Before we begin

1. Download the practical script
2. Set your working directory
3. `set.seed(1234)` – for any random number generation

```
setwd("Your Directory")
set.seed(1234)
```

# Part 1

Create, export, import (read), format, check/screen data

# Part 1.1 – Create data

- Practical to create a dummy set of data
- Create random values for ID, Age, Sex, SBP, DBP

```
set.seed(1234)
data <- cbind(1:120,
              rnorm(120,40,5),
              rbinom(120,1,.4),
              rnorm(120,130,10),
              rnorm(120,80,10)
            )
data <- as.data.frame(data)

colnames(data) <- c("ID", "Age", "Sex", "SBP", "DBP")
data$Sex <- factor(data$Sex, c(0,1), labels = c("Female", "Male") )

### Alternative method, and rounding numerics
set.seed(1234)
data2 <- data.frame(
  ID=1:120,
  Age=round(as.numeric(rnorm(120,40,5))),0),
  Sex=factor(rbinom(120,1,.4)),
  SBP=round(as.numeric(rnorm(120,130,10))), 0),
  DBP=round(as.numeric(rnorm(120,80,10))),0)
)

data2$Sex <- factor(data2$Sex, c(0,1), labels = c("Female", "Male") )
```

▲	ID	Age	Sex	SBP	DBP
1	1	33.96467	Female	102.67780	74.20043
2	2	41.38715	Female	129.00209	70.46721
3	3	45.42221	Female	139.76032	78.20571
4	4	28.27151	Male	134.13869	90.09808
5	5	42.14562	Male	139.12322	80.23627
6	6	42.53028	Female	149.83732	73.50972
7	7	37.12630	Male	141.69109	74.95626
8	8	37.26684	Female	124.91263	96.14391
9	9	37.17774	Male	137.04180	75.53040
10	10	35.54981	Male	128.01584	87.63177
11	11	37.61404	Female	124.61929	94.71719
12	12	35.00807	Female	101.44241	84.43665
13	13	36.11873	Female	122.10353	75.78278
14	14	40.32229	Female	134.87815	79.59998
15	15	44.79747	Female	151.68033	75.07720
16	16	39.44857	Female	135.00695	92.27717
17	17	37.44495	Female	136.20210	78.50446
18	18	35.44402	Female	120.34097	95.49983
19	19	35.81414	Female	131.62655	74.38387
20	20	52.07918	Female	109.21762	73.52883

▲	ID	Age	Sex	SBP	DBP
1	1	34	Female	103	74
2	2	41	Female	129	70
3	3	45	Female	140	78
4	4	28	Male	134	90
5	5	42	Male	139	80
6	6	43	Female	150	74
7	7	37	Male	142	75
8	8	37	Female	125	96
9	9	37	Male	137	76
10	10	36	Male	128	88
11	11	38	Female	125	95
12	12	35	Female	101	84
13	13	36	Female	122	76
14	14	40	Female	135	80
15	15	45	Female	152	75
16	16	39	Female	135	92
17	17	37	Female	136	79
18	18	35	Female	120	95
19	19	36	Female	132	74
20	20	52	Female	109	74

## Part 1.2 – Export your data

- Use package `readr` or already available in `tidyverse`
- Export to your working directory unless specified otherwise

```
write_csv(data2, file = "data2.csv")
```

# Part 1.3 – Import/Read external data

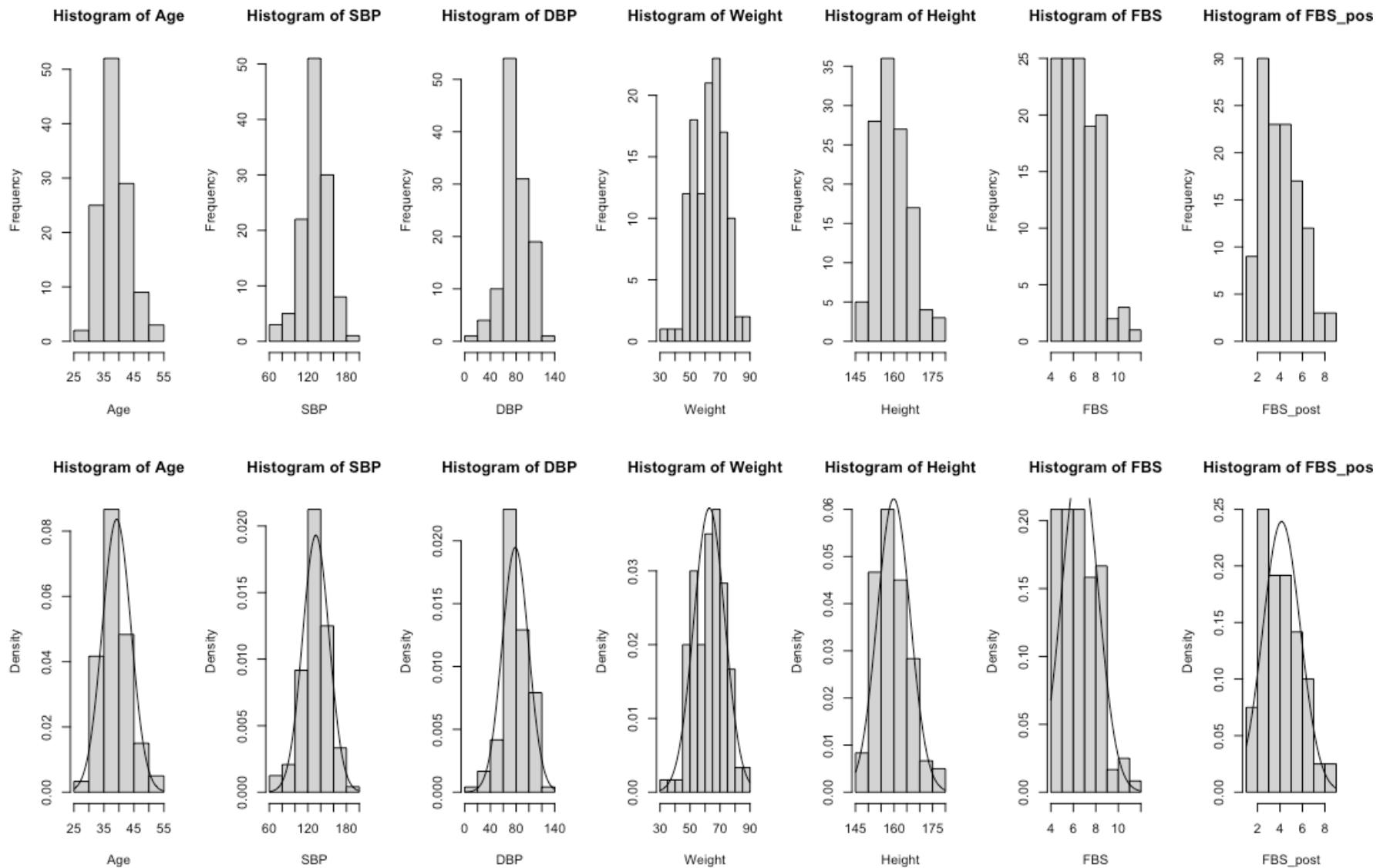
```
training <- read_csv("training.csv")
```

ID	Age	Sex	Smoking	SBP	DBP	Weight	Height	FBS	Intervention	FBS_post	Smoking2
1	34	Female	0	75	68	50.9	155	4.1	Diet & Exercise	2.3	No
2	41	Female	0	128	61	59.6	166	4.7	Diet & Exercise	2.3	No
3	45	Female	1	150	76	60.3	153	6.7	Diet	4.8	Yes
4	28	Male	0	138	100	67.2	168	4.7	Diet & Exercise	2.1	No
5	42	Male	1	148	80	52.4	166	6.2	Diet & Exercise	3.1	Yes
6	43	Female	0	170	67	59.9	149	8.1	Diet	6.3	No
7	37	Male	0	153	70	69.3	170	6.6	Diet & Exercise	3.8	No
8	37	Female	1	120	112	62.8	151	8.2	Diet & Exercise	5.4	Yes
9	37	Male	1	144	71	67.7	163	8.2	Diet & Exercise	5.2	Yes
10	36	Male	0	126	95	70.3	158	8.1	Diet & Exercise	5.1	No
11	38	Female	1	119	109	38.9	160	5.8	Diet & Exercise	2.9	Yes
12	35	Female	0	73	89	48.4	163	6.5	Diet & Exercise	3.8	No
13	36	Female	1	114	72	74.3	155	6.6	Diet	5.0	Yes
14	40	Female	1	140	79	61.4	151	8.1	Diet & Exercise	5.2	Yes
15	45	Female	0	173	70	67.4	158	9.0	Diet	7.1	No
16	39	Female	1	140	105	66.2	153	8.1	Diet	6.0	Yes
17	37	Female	0	142	77	73.7	148	8.3	Diet	6.3	No
18	35	Female	0	111	111	49.4	157	5.2	Diet & Exercise	2.0	No

# Part 1.4 – Screen & prepare the data

- **# Assign proper data type for each variable**
- `training$Sex <- factor(training$Sex)`
- `training$Intervention <- factor(training$Intervention, levels = c(0,1), labels = c("Diet", "Diet & Exercise"))`
- `training$Smoking2 <- factor(training$Smoking, levels = c(0,1), labels = c("No", "Yes"))`
- **# Showing Normal curve**
- `hist(Age, probability = TRUE)`
- `curve(dnorm(x, mean=mean(Age), sd=sd(Age)), add=TRUE)`
- **# Checking distribution for numerical variables**
- `## Visually`
- `par(mfrow= c(2,7))`
- `hist(Age)`
- `hist(SBP)`
- `hist(DBP)`
- **# Showing Normal curve**
- `hist(SBP, probability = TRUE)`
- `curve(dnorm(x, mean=mean(SBP), sd=sd(SBP)), add=TRUE)`
- **# Showing Normal curve**
- `hist(DBP, probability = TRUE)`
- `curve(dnorm(x, mean=mean(DBP), sd=sd(DBP)), add=TRUE)`

Only for 3 variables



# Checking Normality

```
## Showing the moment values - determine
library(psych)
describe(Age)
describe(SBP)
describe(DBP)

# Normality test
shapiro.test(Age)
shapiro.test(SBP)
shapiro.test(DBP)
```

```
> ## Showing the moment values - determine
> library(psych)
> describe(Age)
  vars   n   mean    sd median trimmed   mad min max range skew kurtosis    se
X1     1 120 39.24 4.77      39    38.9 4.45    28   53     25 0.55      0.2 0.44
> describe(SBP)
  vars   n   mean    sd median trimmed   mad min max range skew kurtosis    se
X1     1 120 132.45 20.67    135.5 132.96 17.05    65  188    123 -0.43      0.97 1.89
> describe(DBP)
  vars   n   mean    sd median trimmed   mad min max range skew kurtosis    se
X1     1 120 78.81 20.48    78.5  79.23 18.53    12  125   113 -0.25      0.32 1.87
> # Normality test
> shapiro.test(Age)

  Shapiro-Wilk normality test

  data: Age
  W = 0.97127, p-value = 0.01135

> shapiro.test(SBP)

  Shapiro-Wilk normality test

  data: SBP
  W = 0.97457, p-value = 0.02247

> shapiro.test(DBP)

  Shapiro-Wilk normality test

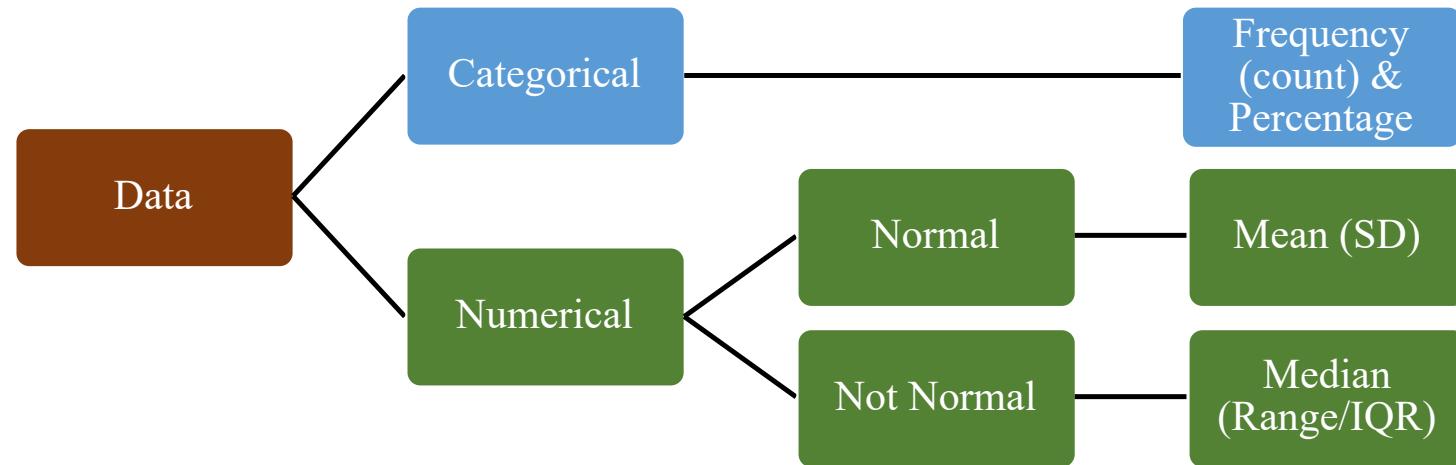
  data: DBP
  W = 0.98402, p-value = 0.1677
```

A close-up photograph of a person's hands typing on a laptop keyboard. The hands are positioned over the center of the keyboard, with fingers pressing keys. A silver laptop trackpad is visible below the hands. The background is blurred, showing the warm glow of a computer screen.

# Part 2

## Descriptive statistics

# Descriptive statistics

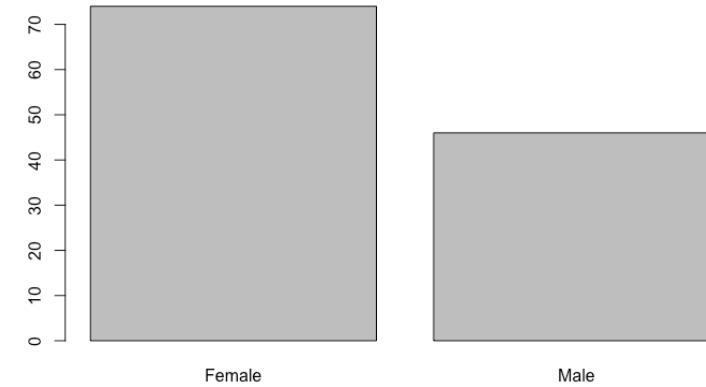


# Part 2.1 – Describe categorical variable

---

- `table(Sex)`
- `round(prop.table(table(Sex))*100, digit=1)`
- `plot(Sex) # Bar chart for the count`

```
> # Describe categorical variable (Sex)
> table(Sex)
Sex
Female   Male
    74     46
> round(prop.table(table(Sex))*100, digit=1)
Sex
Female   Male
  61.7   38.3
> |
```



# Using gtsummary for “Table 1”

```
# Using gtsummary to describe all variables

library(gtsummary)
library(tidyverse)

# The simplest format
training %>%
 tbl_summary()

# A better format
training %>%
 tbl_summary(
    statistic = list(all_continuous() ~ "{mean} ({sd})",
                     all_categorical() ~ "{n}/{N} ({p}%)"),
    digits = all_continuous() ~ 1
  )

# Even a better format
training %>%
  select(Age:Smoking2) %>%
  tbl_summary(
    statistic = list(c(Age, SBP, DBP, Weight, Height) ~ "{mean} ({sd})",
                     c(FBS, FBS_post) ~ "{median} ({p25}, {p75})",
                     all_categorical() ~ "{n}/{N} ({p}%)"),
    digit = all_continuous() ~ 1
  )
```

Characteristic	N = 120 <sup>1</sup>	Characteristic	N = 120 <sup>1</sup>	Characteristic	N = 120 <sup>1</sup>
ID	60 (31, 90)	ID	60.5 (34.8)	Age	39.2 (4.8)
Age	39.0 (36.0, 42.0)	Age	39.2 (4.8)	Sex	
Sex		Sex		Female	74/120 (62%)
Female	74 (62%)	Female	74/120 (62%)	Male	46/120 (38%)
Male	46 (38%)	Male	46/120 (38%)	Smoking	60/120 (50%)
Smoking	60 (50%)	Smoking	60/120 (50%)	SBP	132.4 (20.7)
SBP	136 (122, 144)	SBP	132.4 (20.7)	DBP	78.8 (20.5)
DBP	78 (67, 91)	DBP	78.8 (20.5)	Weight	62.9 (10.4)
Weight	64 (53, 70)	Weight	62.9 (10.4)	Height	159.9 (6.4)
Height	160 (155, 164)	Height	159.9 (6.4)	FBS	6.4 (5.3, 7.9)
FBS	6.40 (5.27, 7.90)	FBS	6.6 (1.6)	Intervention	
Intervention		Intervention		Diet	58/120 (48%)
Diet	58 (48%)	Diet	58/120 (48%)	Diet & Exercise	62/120 (52%)
Diet & Exercise	62 (52%)	Diet & Exercise	62/120 (52%)	FBS_post	4.0 (2.8, 5.2)
FBS_post	4.00 (2.80, 5.20)	FBS_post	4.1 (1.7)	Smoking2	60/120 (50%)
Smoking2	60 (50%)	Smoking2	60/120 (50%)	<sup>1</sup> Mean (SD); n/N (%)	
<sup>1</sup> Median (IQR); n (%)					

# Part 3

## Analytical statistics – compare proportions

# Cross-tabulate using Base

```
> # Cross-tabulate Sex with Smoking (Compare proportions
> table1 <- table(Smoking, Sex)
> table1
   Sex
Smoking Female Male
  0     41    19
  1     33    27
>
> margin.table(table1, 1) # Row (Smoking) total
Smoking
  0  1
60 60
> margin.table(table1, 2) # Column (Sex) total
Sex
Female  Male
  74    46
>
> # Results in %
> prop.table(table1, 1) # By Smoking status
   Sex
Smoking Female      Male
  0 0.6833333 0.3166667
  1 0.5500000 0.4500000
> prop.table(table1, 2) # By Sex
   Sex
Smoking Female      Male
  0 0.5540541 0.4130435
  1 0.4459459 0.5869565
>
```

```
> # Results in % & rounded to 1 decimal point
> round(prop.table(table1, 1)*100, digit=1)
   Sex
Smoking Female Male
  0     68.3 31.7
  1     55.0 45.0
> round(prop.table(table1, 2)*100, digit=1)
   Sex
Smoking Female Male
  0     55.4 41.3
  1     44.6 58.7
>
> # Chi-square test
> chisq.test(table1, correct=FALSE) # Do continuity corr
expected count < 5

Pearson's Chi-squared test

data: table1
X-squared = 2.2562, df = 1, p-value = 0.1331

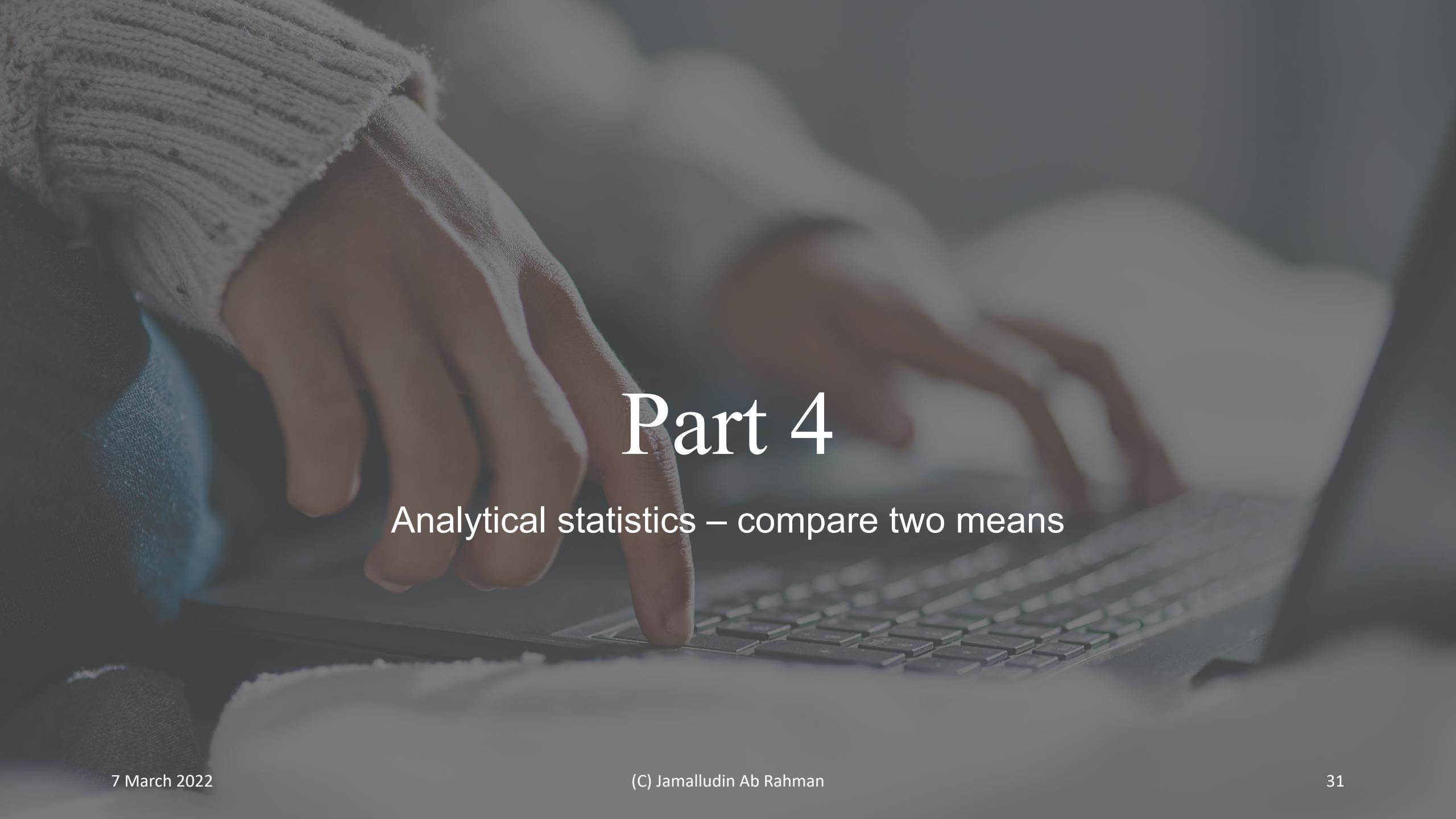
>
> # To obtain the observed & expected count
> chisq.test(table1)$observed
   Sex
Smoking Female Male
  0     41    19
  1     33    27
> chisq.test(table1)$expected
   Sex
Smoking Female Male
  0     37    23
  1     37    23
> |
```

# Chi-square test using `gtsummary`

```
# Using gtsummary to compare
proportions
training %>%
  select(Smoking2, Sex) %>%
 tbl_summary(by = Sex) %>%
  add_p(pvalue_fun = ~
style_pvalue(.x, digits = 3))
```

Characteristic	Female, N = 74 <sup>1</sup>	Male, N = 46 <sup>1</sup>	p-value <sup>2</sup>
Smoking2	33 (45%)	27 (59%)	0.133

<sup>1</sup> n (%)  
<sup>2</sup> Pearson's Chi-squared test



# Part 4

## Analytical statistics – compare two means

# Compare two means

# To check the variance assumption

```
var.test(FBS ~ Smoking2)
```

# By default, equal variance assumed

```
t.test(FBS ~ Smoking2)
```

# If equal variance can't be assumed

```
t.test(FBS ~ Smoking2,  
var.equal = FALSE)
```

```
> # Compare mean FBS by Smoking  
>  
> var.test(FBS ~ Smoking2) # To check the variance assumption, if P > 0.05 the equal variance can be assumed  
  
F test to compare two variances  
  
data: FBS by Smoking2  
F = 1.4246, num df = 59, denom df = 59, p-value = 0.1771  
alternative hypothesis: true ratio of variances is not equal to 1  
95 percent confidence interval:  
 0.8509299 2.3849163  
sample estimates:  
ratio of variances  
 1.424569  
  
> t.test(FBS ~ Smoking2) # By default, equal variance assumed  
  
Welch Two Sample t-test  
  
data: FBS by Smoking2  
t = -0.48065, df = 114.49, p-value = 0.6317  
alternative hypothesis: true difference in means between group No and group Yes is not equal to 0  
95 percent confidence interval:  
 -0.7255158 0.4421825  
sample estimates:  
mean in group No mean in group Yes  
 6.523333 6.665000  
  
> t.test(FBS ~ Smoking2, var.equal = FALSE) # If equal variance can't be assumed  
  
Welch Two Sample t-test  
  
data: FBS by Smoking2  
t = -0.48065, df = 114.49, p-value = 0.6317  
alternative hypothesis: true difference in means between group No and group Yes is not equal to 0  
95 percent confidence interval:  
 -0.7255158 0.4421825  
sample estimates:  
mean in group No mean in group Yes  
 6.523333 6.665000
```

# Compare means using gtsummary

```
training %>%  
  select(FBS, Smoking2) %>%  
 tbl_summary(by = Smoking2,  
             statistic = FBS ~ c("{mean} ({sd})")) %>%  
  add_p(test.args = all_tests("t.test") ~ list(var.equal = TRUE),  
        pvalue_fun = ~style_pvalue(.x, digits = 3)) %>%  
  modify_spanning_header(all_stat_cols() ~ "***Smoking***")
```

Smoking			
Characteristic	No, N = 60 <sup>1</sup>	Yes, N = 60 <sup>1</sup>	p-value <sup>2</sup>
FBS	6.52 (1.75)	6.66 (1.47)	0.467

<sup>1</sup> Mean (SD)  
<sup>2</sup> Wilcoxon rank sum test

# Visualising means by group

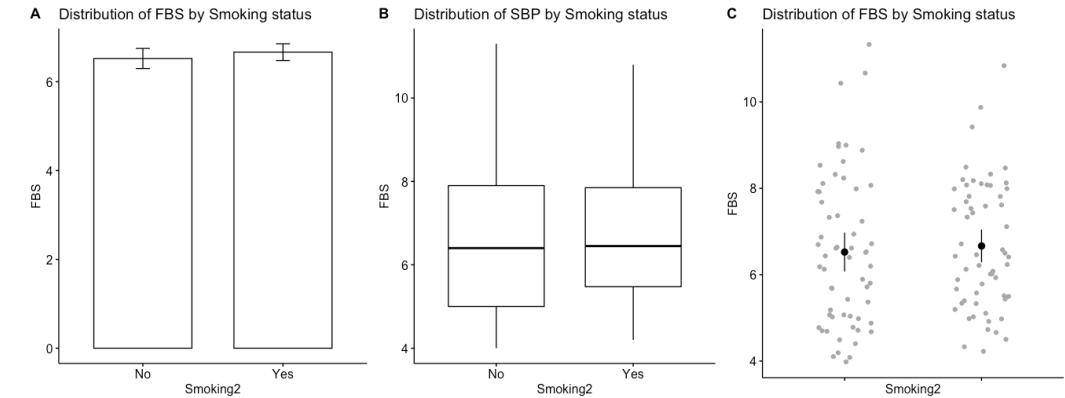
```
library(ggpubr)

A <- ggbarplot(data = training,
                x = "Smoking2",
                y = "FBS",
                main = "Distribution of FBS by Smoking status",
                add = "mean_se")
A

B <- ggboxplot(data = training,
                x = "Smoking2",
                y = "FBS",
                main = "Distribution of SBP by Smoking status")
B

C <- ggerrorplot(data = training, x="Smoking2", y="FBS",
                  main = "Distribution of FBS by Smoking status",
                  desc_stat = "mean_ci",
                  add = "jitter",
                  add.params = list(color = "darkgray"))
C

ggarrange(A, B, C + rremove("x.text"),
          labels = c("A", "B", "C"),
          ncol = 3, nrow = 1)
```



# Part 5

Analytical statistics – compare more than two means

# One-way ANOVA

```
> # Compare means FBS by BMI Status  
> aov(training$FBS ~ training$BMI_Status)  
Call:  
aov(formula = training$FBS ~ training$BMI_Status)
```

Terms:

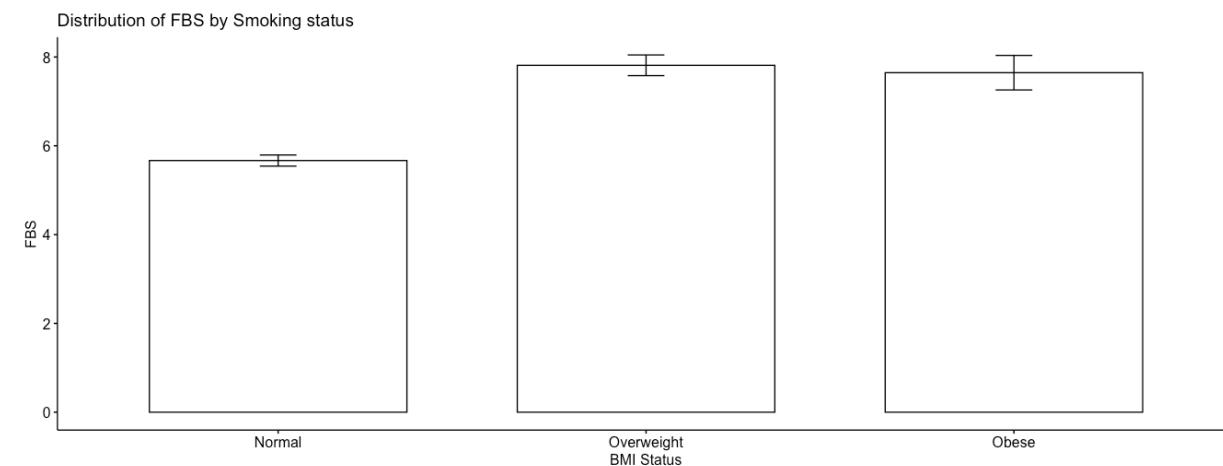
	training\$BMI_Status	Residuals
Sum of Squares	130.6574	177.4685
Deg. of Freedom	2	117

Residual standard error: 1.231594

Estimated effects may be unbalanced

```
> summary(aov(training$FBS ~ training$BMI_Status))  
      Df Sum Sq Mean Sq F value    Pr(>F)  
training$BMI_Status  2 130.7   65.33  43.07 9.62e-15 ***  
Residuals          117 177.5    1.52  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Visualising FBS by BMI Status  
ggbarplot(data = training,  
           x = "BMI_Status",  
           y = "FBS",  
           main = "Distribution of FBS by Smoking status",  
           add = "mean_se",  
           xlab = "BMI Status")
```



```

> # Testing homogeneity assumption
> library(car)
>
> leveneTest(FBS ~ BMI_Status, training, center=mean)
Levene's Test for Homogeneity of Variance (center = mean)
  Df F value Pr(>F)
group  2  1.5498 0.2166
      117

>
> # Post-hoc test if equal variance
> pairwise.t.test(training$FBS, training$BMI_Status, p.adj = "bonf")

  Pairwise comparisons using t tests with pooled SD

data: training$FBS and training$BMI_Status

Normal Overweight
Overweight 1.4e-13 -
Obese      3.8e-07 1

P value adjustment method: bonferroni
>
> # Post-hoc test if can't assume equal variance
> TukeyHSD(aov(training$FBS ~ training$BMI_Status))
  Tukey multiple comparisons of means
  95% family-wise confidence level

Fit: aov(formula = training$FBS ~ training$BMI_Status)

`$`training$BMI_Status`  

  diff      lwr      upr      p adj
Overweight-Normal 2.1459937 1.552252 2.7397356 0.0000000
Obese-Normal     1.9795025 1.144369 2.8146364 0.0000004
Obese-Overweight -0.1664912 -1.058015 0.7250322 0.8974380

```

# Post hoc test

```

# Compare means using gtsummary
training %>%
  select(FBS, BMI_Status) %>%
 tbl_summary(by = BMI_Status,
  statistic = FBS ~ c("{mean} ({sd})"),
  digits = FBS ~ 1)%>%
add_p(FBS ~ "aov",
  pvalue_fun = ~style_pvalue(.x, digits = 3)) %>%
modify_spanning_header(all_stat_cols() ~ "**BMI Status**")

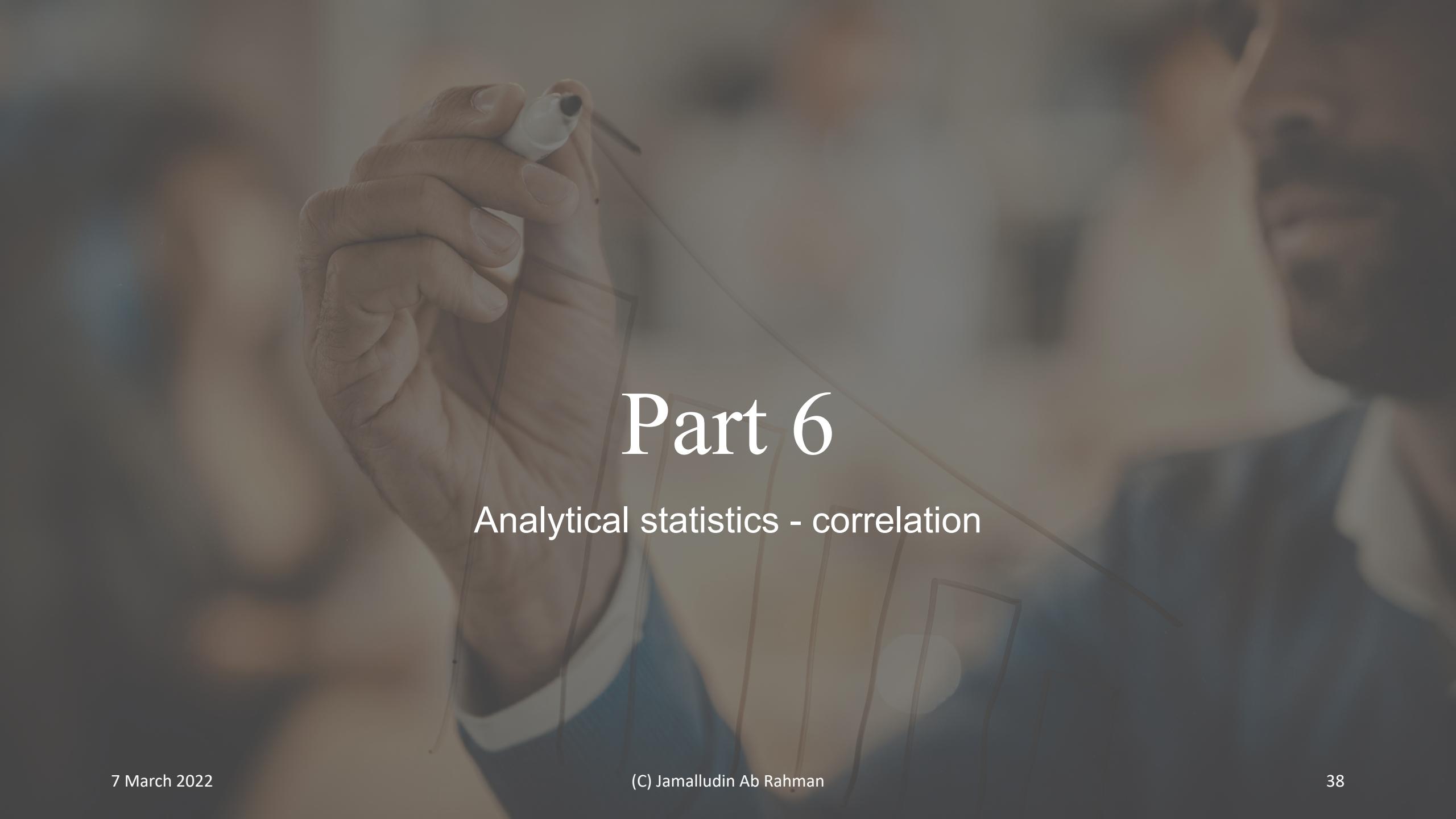
```

---

	BMI Status			
Characteristic	Normal, N = 67 <sup>1</sup>	Overweight, N = 38 <sup>1</sup>	Obese, N = 15 <sup>1</sup>	p-value <sup>2</sup>
FBS	5.7 (1.0)	7.8 (1.4)	7.6 (1.5)	<0.001

<sup>1</sup> Mean (SD)  
<sup>2</sup> One-way ANOVA

---

A blurred background image of a person's hands holding a whiteboard marker and writing on a whiteboard. The person is wearing a dark t-shirt. The whiteboard has some faint, illegible markings.

# Part 6

## Analytical statistics - correlation

# Correlation coefficient tests

```
> # Correlation test  
> cor.test(training$FBS, training$SBP, method = "pearson") # If Normal
```

Pearson's product-moment correlation

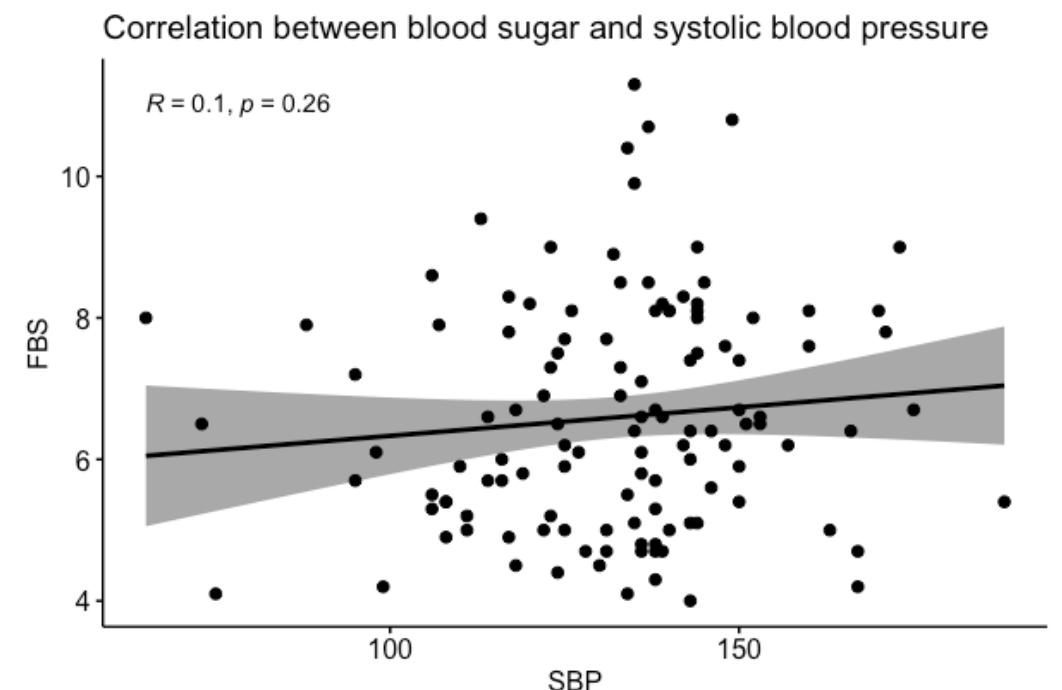
```
data: training$FBS and training$SBP  
t = 1.1321, df = 118, p-value = 0.2599  
alternative hypothesis: true correlation is not equal to 0  
95 percent confidence interval:  
-0.07701355 0.27773965  
sample estimates:  
cor  
0.1036587
```

```
> cor.test(training$FBS, training$SBP, method = "spearman", exact=FALSE) # If not Normal
```

Spearman's rank correlation rho

```
data: training$FBS and training$SBP  
S = 248981, p-value = 0.1403  
alternative hypothesis: true rho is not equal to 0  
sample estimates:  
rho  
0.1354242
```

```
ggscatter(data = training,  
y = "FBS",  
x = "SBP",  
add = "reg.line", conf.int = TRUE, cor.coef = TRUE,  
main = "Correlation between blood sugar and systolic blood pressure")
```



# Part 7

## Analytical statistics – non-parametric tests

# Wilcoxon Rank Sum Test

```
> # Compare two numericals  
> # independent 2-group Mann-Whitney U Test  
> wilcox.test(training$FBS ~ training$BP)
```

Wilcoxon rank sum test with continuity correction

```
data: training$FBS by training$BP  
W = 2091.5, p-value = 0.07831  
alternative hypothesis: true location shift is not equal to 0
```

```
training %>%  
  select(FBS, BP) %>%  
 tbl_summary(  
    by = BP,  
    statistic = FBS ~ "{median} ({p25}, {p75})"  
) %>%  
  add_p(all_continuous() ~ "wilcox.test") %>%  
  modify_spanning_header(all_stat_cols() ~ "**Blood Pressure**")
```

Characteristic	Blood Pressure		
	High, N = 69 <sup>1</sup>	Normal, N = 51 <sup>1</sup>	p-value <sup>2</sup>
FBS	6.50 (5.40, 8.10)	6.10 (5.00, 7.15)	0.078

<sup>1</sup> Median (IQR)  
<sup>2</sup> Wilcoxon rank sum test

# Kruskall Wallis Test

```
> # Kruskal Wallis Test One Way Anova by Ranks
> kruskal.test(training$FBS ~ training$BMI_Status)

Kruskal-Wallis rank sum test

data: training$FBS by training$BMI_Status
Kruskal-Wallis chi-squared = 51.204, df = 2, p-value = 7.605e-12

>
> # Dunn Test, a post-hoc test for Kruskall Wallis
> library(FSA)
> dunnTest(FBS ~ BMI_status,
+           data = training,
+           method = "bonferroni")
Dunn (1964) Kruskal-Wallis multiple comparison
p-values adjusted with the Bonferroni method.

Comparison          Z      P.unadj      P.adj
1 Normal - Obese -4.4597645 8.204978e-06 2.461493e-05
2 Normal - Overweight -6.5550513 5.562277e-11 1.668683e-10
3 Obese - Overweight -0.1878895 8.509633e-01 1.000000e+00
> |
```

```
training %>%
  select(FBS, BMI_status) %>%
 tbl_summary(
    by = BMI_status,
    statistic = FBS ~ "{median} ({p25}, {p75})"
  ) %>%
  add_p(all_continuous() ~ "kruskal.test") %>%
  modify_spanning_header(all_stat_cols() ~ "**BMI Status**")
```

---

Characteristic	BMI Status			p-value <sup>2</sup>
	Normal, N = 67 <sup>1</sup>	Overweight, N = 38 <sup>1</sup>	Obese, N = 15 <sup>1</sup>	
FBS	5.50 (4.95, 6.30)	7.85 (6.98, 8.20)	8.00 (6.60, 8.50)	<0.001

<sup>1</sup> Median (IQR)  
<sup>2</sup> Kruskal-Wallis rank sum test

---



# Part 8

## Paired samples

# Paired t-test

```
> # Paired means - compare means FBS & FBS after as overall (within changes)
> t.test(training$FBS, training$FBS_post, paired = TRUE, alternative = "two.sided")
```

Paired t-test

```
data: training$FBS and training$FBS_post
t = 47.362, df = 119, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 2.342779 2.547221
sample estimates:
mean of the differences
 2.445
```

```
# Table to show within and between changes
training %>%
  select(Intervention, FBS, FBS_post) %>%
 tbl_summary(
  by = Intervention,
  statistic = all_continuous() ~ "{mean} ({sd})",
  label = list(FBS ~ "Before", FBS_post ~ "After")) %>%
add_n() %>%
add_difference() %>%
modify_header(label ~ "***FBS***") %>%
modify_spanning_header(all_stat_cols() ~ "***Type of Intervention***")
```

FBS	N	Type of Intervention		Difference <sup>2</sup>	95% CI <sup>2,3</sup>	p-value <sup>2</sup>
		Diet, N = 58 <sup>1</sup>	Diet & Exercise, N = 62 <sup>1</sup>			
Before	120	6.76 (1.58)	6.44 (1.64)	0.32	-0.26, 0.90	0.3
After	120	4.81 (1.55)	3.53 (1.54)	1.3	0.73, 1.8	<0.001

<sup>1</sup> Mean (SD)  
<sup>2</sup> Welch Two Sample t-test  
<sup>3</sup> CI = Confidence Interval

# Self Practice

# Instruction

1. Please download the practice dataset from  
<https://raw.githubusercontent.com/profjamal/biostatistics/main/healthstatus.csv> - Use only variable age:hba1c
2. Fulfil the following objectives:
  1. Describe baseline characteristics (based on age, sex, exercise, smoking, BMI status and blood pressure status) of the study population. Summarise that in Table 1
  2. What are significant factors related to HbA1c? Summarise that in Table 2