

# BasicR Course

## 15 – 19 February 2021

DAY 2

Molecular Biotechnology  
- Master



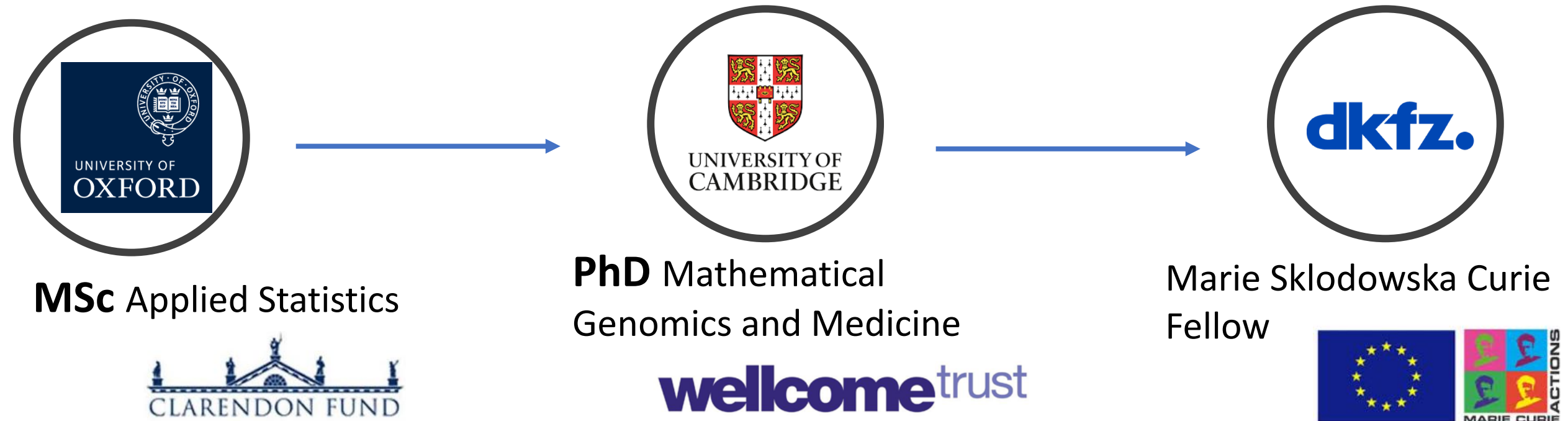
UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

dkfz.



# COURSE INSTRUCTOR

**RAJBIR NATH BATRA**



# DAY 2 AND 3

Data Import

Wrangling

- Tidy + Manipulating
- Summarizing
- Cleaning

Exploration

- Visualization
- Descriptive Statistics

Statistical Inference

- Foundation of inference
- Basic statistical tests
- Linear regression



**Day 2**



**Day 3**

# DAY 2 AND 3

Data Import

Wrangling

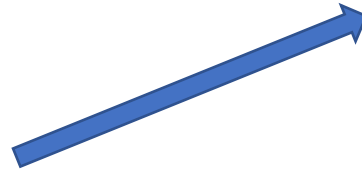
- Tidy + Manipulating
- Summarizing
- Cleaning

Exploration

- Visualization
- Descriptive Statistics

Statistical Inference

- Foundation of inference
- Basic statistical tests
- Linear regression



**FOCUS IS ON APPLICATION IN R**

## **NOT A STATISTICS COURSE**

Course does not cover:

- Principles of study design
  - Types of experiments/ studies
  - Reducing bias in study design
- Statistical theory
  - Probability and random variables
  - Statistical distributions
  - Central Limit Theorem
  - Hypothesis testing
  - Type1 and Type 2 error
  - Test statistic and standard error
  - Confidence intervals and p-values

# DAY 2

tidyverse



- Data Import
- Wrangling
  - Tidying, Cleaning, Summarizing
- Exploration
  - Visualization and Descriptive Statistics

tidyverse

- All packages within tidyverse use the same language/ grammar
- Designed for streamlined data exploration and analysis
- Rigid and expects data to be in specific format

Can also use Base R

- More flexible than tidyverse

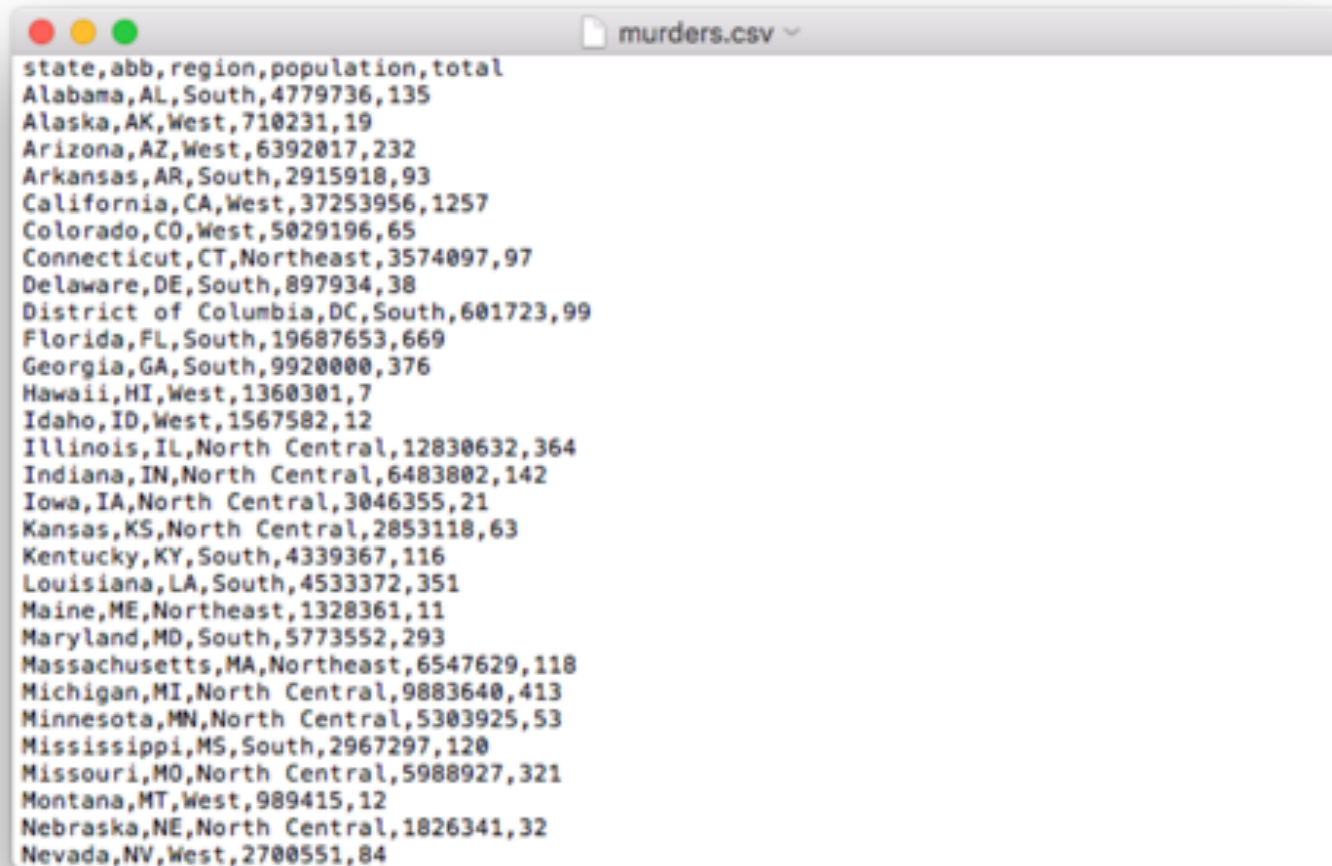
# **IMPORTING DATA**

# METHODS OF DATA IMPORT

- Import electronic spreadsheets
- Datasets stored as R objects (via packages)
- Download files from the internet using R
- ...

# IMPORTING SPREADSHEETS

Txt/ csv file



```
state,abb,region,population,total
Alabama,AL,South,4779736,135
Alaska,AK,West,710231,19
Arizona,AZ,West,6392017,232
Arkansas,AR,South,2915918,93
California,CA,West,37253956,1257
Colorado,CO,West,5029196,65
Connecticut,CT,Northeast,3574097,97
Delaware,DE,South,897934,38
District of Columbia,DC,South,601723,99
Florida,FL,South,19687653,669
Georgia,GA,South,9920000,376
Hawaii,HI,West,1360301,7
Idaho,ID,West,1567582,12
Illinois,IL,North Central,12830632,364
Indiana,IN,North Central,6483802,142
Iowa,IA,North Central,3046355,21
Kansas,KS,North Central,2853118,63
Kentucky,KY,South,4339367,116
Louisiana,LA,South,4533372,351
Maine,ME,Northeast,1328361,11
Maryland,MD,South,5773552,293
Massachusetts,MA,Northeast,6547629,118
Michigan,MI,North Central,9883640,413
Minnesota,MN,North Central,5303925,53
Mississippi,MS,South,2967297,120
Missouri,MO,North Central,5988927,321
Montana,MT,West,989415,12
Nebraska,NE,North Central,1826341,32
Nevada,NV,West,2700551,84
```

Excel file

	A	B	C	D	E	F	G	H
1	state	abb	region	population	total			
2	Alabama	AL	South	4779736	135			
3	Alaska	AK	West	710231	19			
4	Arizona	AZ	West	6392017	232			
5	Arkansas	AR	South	2915918	93			
6	California	CA	West	37253956	1257			
7	Colorado	CO	West	5029196	65			
8	Connecticut	CT	Northeast	3574097	97			
9	Delaware	DE	South	897934	38			
10	District of Co	DC	South	601723	99			
11	Florida	FL	South	19687653	669			
12	Georgia	GA	South	9920000	376			
13	Hawaii	HI	West	1360301	7			
14	Idaho	ID	West	1567582	12			
15	Illinois	IL	North Centra	12830632	364			
16	Indiana	IN	North Centra	6483802	142			
17	Iowa	IA	North Centra	3046355	21			
18	Kansas	KS	North Centra	2853118	63			
19	Kentucky	KY	South	4339367	116			
20	Louisiana	LA	South	4533372	351			
21	Maine	ME	Northeast	1328361	11			
22	Maryland	MD	South	5773552	293			
23	Massachuset	MA	Northeast	6547629	118			
24	Michigan	MI	North Centra	9883640	413			
25	Minnesota	MN	North Centra	5303925	53			
26	Mississippi	MS	South	2967297	120			
27	Missouri	MO	North Centra	5988927	321			
28	Montana	MT	West	989415	12			
29	Nebraska	NE	North Centra	1826341	32			

HEADER - The first row contains column names rather than data.



# IMPORTING SPREADSHEETS

## readr library



The following functions are available to read-in spreadsheets:

Function	Format	Typical suffix
read_table	white space separated values	txt
read_csv	comma separated values	csv
read_csv2	semicolon separated values	csv
read_tsv	tab delimited separated values	tsv
read_delim	general text file format, must define delimiter	txt

<https://github.com/rafalab/dsbook>↔

# IMPORTING SPREADSHEETS

## readxl library

The package provides functions to read-in Microsoft Excel formats:

Function	Format	Typical suffix
read_excel	auto detect the format	xls, xlsx
read_xls	original format	xls
read_xlsx	new format	xlsx

<https://github.com/rafalab/dsbook>↔

**WRANGLING –**

**TIDY +**

**MANIPULATING**

# TIDY FORMAT

## Tidy format

Hadley Wickham defines "tidy data" for data storage by analysts

## DOs

1. Each variable forms a column, and that column contains one "type" of data
2. Each observation forms a row
3. Each type of observational unit forms a table

## DON'Ts

- Column headers contain values, rather than names
- Multiple variables are stored in a single column
- Variables are stored in both rows and columns
- Multiple observational types are stored in a single table
- A single observational unit is stored in multiple tables.

<http://vita.had.co.nz/papers/tidy-data.pdf>

# TIDY FORMAT

country	year	m014	m1524	m2534	m3544	m4554	m5564	m65	mu	f014
AD	2000	0	0	1	0	0	0	0	—	—
AE	2000	2	4	4	6	5	12	10	—	3
AF	2000	52	228	183	149	129	94	80	—	93
AG	2000	0	0	0	0	0	0	1	—	1
AL	2000	2	19	21	14	24	19	16	—	3
AM	2000	2	152	130	131	63	26	21	—	1
AN	2000	0	0	1	2	0	0	0	—	0
AO	2000	186	999	1003	912	482	312	194	—	247
AR	2000	97	278	594	402	419	368	330	—	121
AS	2000	—	—	—	—	1	1	—	—	—

Table 9: Original TB dataset. Corresponding to each 'm' column for males, there is also an 'f' column for females, **f1524**, **f2534** and so on. These are not shown to conserve space. Note the mixture of 0s and missing values (—). This is due to the data collection process and the distinction is important for this dataset.

<http://vita.had.co.nz/papers/tidy-data.pdf>

# TIDY FORMAT

country	year	column	cases	country	year	sex	age	cases
AD	2000	m014	0	AD	2000	m	0-14	0
AD	2000	m1524	0	AD	2000	m	15-24	0
AD	2000	m2534	1	AD	2000	m	25-34	1
AD	2000	m3544	0	AD	2000	m	35-44	0
AD	2000	m4554	0	AD	2000	m	45-54	0
AD	2000	m5564	0	AD	2000	m	55-64	0
AD	2000	m65	0	AD	2000	m	65+	0
AE	2000	m014	2	AE	2000	m	0-14	2
AE	2000	m1524	4	AE	2000	m	15-24	4
AE	2000	m2534	4	AE	2000	m	25-34	4
AE	2000	m3544	6	AE	2000	m	35-44	6
AE	2000	m4554	5	AE	2000	m	45-54	5
AE	2000	m5564	12	AE	2000	m	55-64	12
AE	2000	m65	10	AE	2000	m	65+	10
AE	2000	f014	3	AE	2000	f	0-14	3

(a) Molten data

(b) Tidy data

Table 10: Tidying the TB dataset requires first melting, and then splitting the `column` column into two variables: `sex` and `age`.

<http://vita.had.co.nz/papers/tidy-data.pdf>

# TIBBLE

Introduced to **data.frame**

```
> as.data.frame(dot)
```

	state	abb	region	population	total
1	Alabama	AL	South	4779736	135
2	Alaska	AK	West	710231	19
3	Arizona	AZ	West	6392017	232
4	Arkansas	AR	South	2915918	93
5	California	CA	West	37253956	1257
6	Colorado	CO	West	5829196	65
7	Connecticut	CT	Northeast	3574097	97
8	Delaware	DE	South	897934	38
9	District of Columbia	DC	South	601723	99
10	Florida	FL	South	19687653	669
11	Georgia	GA	South	9920000	376
12	Hawaii	HI	West	1360301	7
13	Idaho	ID	West	1567582	12
14	Illinois	IL	North Central	12830632	364
15	Indiana	IN	North Central	6483802	142
16	Iowa	IA	North Central	3046355	21
17	Kansas	KS	North Central	2853118	63
18	Kentucky	KY	South	4339367	116
19	Louisiana	LA	South	4533372	351
20	Maine	ME	Northeast	1328361	11
21	Maryland	MD	South	5773552	293
22	Massachusetts	MA	Northeast	6547629	118
23	Michigan	MI	North Central	9883640	413
24	Minnesota	MN	North Central	5303925	53
25	Mississippi	MS	South	2967297	120
26	Missouri	MO	North Central	5988927	321
27	Montana	MT	West	989415	12
28	Nebraska	NE	North Central	1826341	32
29	Nevada	NV	West	2700551	84

<https://github.com/rafalab/dsbook>

# TIBBLE

Introduced to `data.frame`

```
36      Ohio OH North Central 11536504 310
37    Oklahoma OK      South 3751351 111
38      Oregon OR      West 3831074 36
39    Pennsylvania PA Northeast 12702379 457
40    Rhode Island RI Northeast 1052567 16
41    South Carolina SC      South 4625364 207
42    South Dakota SD North Central 814180 8
43      Tennessee TN      South 6346105 219
44        Texas TX      South 25145561 805
45        Utah UT      West 2763885 22
46      Vermont VT Northeast 625741 2
47      Virginia VA      South 8001024 250
48    Washington WA      West 6724540 93
49    West Virginia WV      South 1852994 27
50      Wisconsin WI North Central 5686986 97
51        Wyoming WY      West 563626 5
> |
```

`tibble` are like `data.frame`



More properties

- Displays better

```
> as_tibble(dat)
# A tibble: 51 x 5
  state      abb region population total
  <chr>    <chr> <chr>      <dbl> <dbl>
1 Alabama AL    South    4779736 135
2 Alaska  AK    West     710231 19
3 Arizona AZ    West    6392017 232
4 Arkansas AR    South    2915918 93
5 California CA    West   37253956 1257
6 Colorado CO    West    5029196 65
7 Connecticut CT    Northeast 3574097 97
8 Delaware DE    South     897934 38
9 District of Columbia DC    South     601723 99
10 Florida FL    South   19687653 660
# ... with 41 more rows
```

- Subsets of tibbles are tibbles
- Tibbles can have complex entries
- Tibbles can be grouped

<https://github.com/rafalab/dsbook>↵



# WRANGLING with dplyr

**dplyr** is a grammar of data manipulation

- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.



**pipe: %>%**

dataset

→ select

→ filter

*in R*

dataset %>% select %>% filter

<https://dplyr.tidyverse.org/>

# WRANGLING with dplyr

## One table verbs

<code>arrange()</code>	Arrange rows by column values
<code>count()</code> <code>tally()</code> <code>add_count()</code> <code>add_tally()</code>	Count observations by group
<code>distinct()</code>	Subset distinct/unique rows
<code>filter()</code>	Subset rows using column values
<code>mutate()</code> <code>transmute()</code>	Create, modify, and delete columns
<code>pull()</code>	Extract a single column
<code>relocate()</code>	Change column order
<code>rename()</code> <code>rename_with()</code>	Rename columns
<code>select()</code>	Subset columns using their names and types
<code>summarise()</code> <code>summarize()</code>	Summarise each group to fewer rows
<code>slice()</code> <code>slice_head()</code> <code>slice_tail()</code> <code>slice_min()</code> <code>slice_max()</code> <code>slice_sample()</code>	Subset rows using their positions

## Two table verbs

<code>bind_rows()</code> <code>bind_cols()</code>	Efficiently bind multiple data frames by row and column
<code>reexports</code>	Objects exported from other packages
<code>inner_join()</code> <code>left_join()</code> <code>right_join()</code> <code>full_join()</code>	Mutating joins
<code>nest_join()</code>	Nest join
<code>semi_join()</code> <code>anti_join()</code>	Filtering joins

## Grouping

<code>group_by()</code> <code>ungroup()</code>	Group by one or more variables
<code>group_cols()</code>	Select grouping variables
<code>rowwise()</code>	Group input by rows

<https://dplyr.tidyverse.org>

dplyr cheatsheet - <https://rstudio.com/resources/cheatsheets/>

# SUMMARISE – 1 NUMERICAL VARIABLE

## HISTOGRAM – graphical

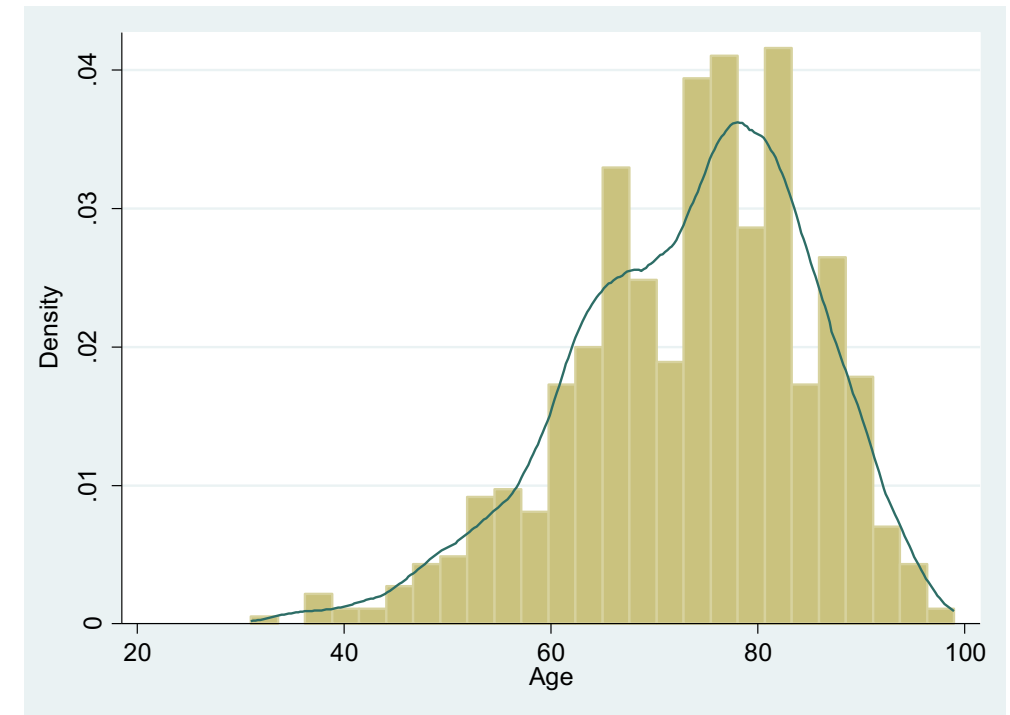
- Histograms provide a view of the ***data density***. Higher bars represent where the data are relatively more common.
- Histograms are especially convenient for describing the ***shape*** of the data distribution.
- The chosen ***bin width*** can alter the story the histogram is telling.

Can use summarise() function for numerical summaries like

Mean, Standard deviation

<https://www.openintro.org/book/os/>

Age (years) in patients



# SUMMARISE – 1 CATEGORICAL VARIABLE

## TABLE

- A table summarizes data for one categorical variable with frequency and proportions (%)

In general  
would you say  
your health is:

	Frequency	Percentage	Cumulative %
Excellent	31	2.46	2.46
Very good	155	12.3	14.76
Good	494	39.21	53.97
Fair	430	34.13	88.1
Poor	150	11.9	100
Total	1,260	100	

Can use `table()` function for numerical summaries

## BAR PLOT - graphical

- A bar plot is a common way to display a single categorical variable.
- Pie chart is not recommended since we cannot compare areas as accurately as heights.



# EXERCISE

Day2\_1.ImportingandWrangling\_Exercise.Rmd

# **WRANGLING – CLEANING DATA**

# DATA CLEANING

- Follows a ***tidy data*** structure
- Remove duplicate rows/values
- Error-free (e.g. free of misspellings)
- Variables should have appropriate data type
  - e.g. numeric, character, factor etc
- Factors (categorical/ ordinal) should have relevant levels
- Remove incorrect/ non-relevant outliers
- Missing data should be set as **NA**

# EXERCISE

Day2\_2\_Cleaning\_Exercise.Rmd



# **EXPLORATION – VISUALISATION WITH GGPLOT2**

# VISUALISATION with ggplot2



3 main building blocks to ggplot

## 1. Data

US Gun Murders in 2010

## 2. Geometric object (type of plot)

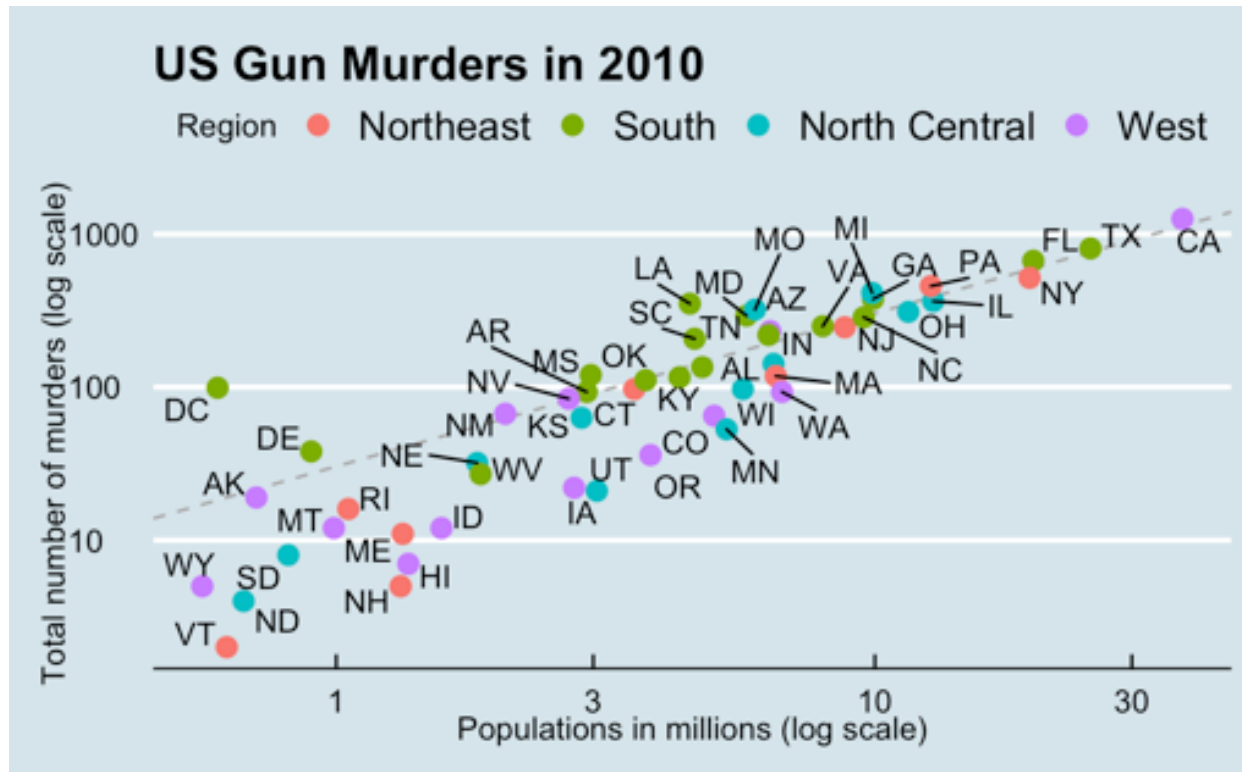
Scatter plot (x and y)

## 3. Aesthetic mapping

- 2 layers - points + labels of states
- colored by Region

## 4. Other elements

- Scale – logged
- Dotted Line of best fit
- Legend
- Style and background theme



<https://github.com/rafalab/dsbook>

# **EXPLORATION – DESCRIPTIVE STATISTICS**

# Data Visualization with ggplot2

## Cheat Sheet



### Basics

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



To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.



Build a graph with **qplot()** or **ggplot()**

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

**ggplot()** `ggplot(data = mpg, aes(x = cyl, y = hwy))`  
Begins a plot that you finish by adding layers to. No defaults, but provides more control than qplot().

**add layers, elements with:**  
`ggplot(mpg, aes(hwy, cyl)) +  
 geom_point(aes(color = cyl)) +  
 geom_smooth(method = "lm") +  
 scale_color_manual(values =  
 c("red", "blue", "green")) +  
 theme_minimal()`

Add a new layer to a plot with a **geom\_\*** or **stat\_\*** function. Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.

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

### One Variable

#### Continuous

`a = ggplot(mpg, aes(hwy))`

- geom\_area**(stat = "bin")  
x, y, alpha, color, fill, linetype, size  
b = `geom_area(aes(y = ..density..), stat = "bin")`
- geom\_density**(kernel = "gaussian")  
x, y, alpha, color, fill, linetype, size, weight  
b = `geom_density(aes(y = ..density..), county)`
- geom\_dotplot**  
x, y, alpha, color, fill
- geom\_freqpoly**  
x, y, alpha, color, linetype, size  
b = `geom_freqpoly(aes(y = ..density..))`
- geom\_histogram**(binwidth = 5)  
x, y, alpha, color, fill, linetype, size, weight  
b = `geom_histogram(aes(y = ..density..))`

#### Discrete

`b = ggplot(mpg, aes(cyl))`

- geom\_bar**  
x, alpha, color, fill, linetype, size, weight

### Graphical Primitives

`c = ggplot(mpg, aes(long, lat))`

- geom\_polygon**(aes(group = group))  
x, y, alpha, color, fill, linetype, size

`d = ggplot(economics, aes(date, unemploy))`

- geom\_path**(lineend = "butt", linejoin = "round", linewidth = 1)  
x, y, alpha, color, linetype, size
- geom\_ribbon**(aes(ymin = unemploy - 900, ymax = unemploy + 900))  
x, y, alpha, color, fill, linetype, size

`e = ggplot(seals, aes(x = long, y = lat))`

- geom\_segment**(aes(xend = long + delta\_long, yend = lat + delta\_lat))  
x, y, alpha, color, linetype, size
- geom\_rect**(aes(xmin = long, ymin = lat, xmax = long + delta\_long, ymax = lat + delta\_lat))  
x, y, alpha, color, fill, linetype, size

### Two Variables

#### Continuous X, Continuous Y

`f = ggplot(mpg, aes(cty, hwy))`

- geom\_blank**
- geom\_jitter**  
x, y, alpha, color, fill, shape, size
- geom\_point**  
x, y, alpha, color, fill, shape, size
- geom\_quantile**  
x, y, alpha, color, linetype, size, weight
- geom\_rug**(sides = "b")  
alpha, color, linetype, size
- geom\_smooth**(model = lm)  
x, y, alpha, color, fill, linetype, size, weight
- geom\_text**(aes(label = cty))  
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

#### Discrete X, Continuous Y

`g = ggplot(mpg, aes(class, hwy))`

- geom\_bar**(stat = "identity")  
x, y, alpha, color, fill, linetype, size, weight
- geom\_boxplot**  
lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight
- geom\_dotplot**(binaxis = "y", stacked = "center")  
x, y, alpha, color, fill
- geom\_violin**(scale = "area")  
x, y, alpha, color, fill, linetype, size, weight

#### Discrete X, Discrete Y

`h = ggplot(diamonds, aes(cut, color))`

- geom\_jitter**  
x, y, alpha, color, fill, shape, size

#### Continuous Bivariate Distribution

`i = ggplot(movies, aes(year, rating))`

- geom\_bin2d**(binwidth = c(5, 0.5))  
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight
- geom\_density2d**  
x, y, alpha, color, linetype, size
- geom\_hex**  
x, y, alpha, color, fill, size

#### Continuous Function

`j = ggplot(economics, aes(date, unemploy))`

- geom\_area**  
x, y, alpha, color, fill, linetype, size
- geom\_line**  
x, y, alpha, color, linetype, size
- geom\_step**(direction = "hv")  
x, y, alpha, color, linetype, size

#### Visualizing error

`df = data.frame(grp = c("A", "B"), fit = 4.5, se = 1.2)`  
`k = ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))`

- geom\_crossbar**(latten = 2)  
x, y, ymax, ymin, alpha, color, fill, linetype, size
- geom\_errorbar**  
x, ymax, ymin, alpha, color, linetype, size, width (also `geom_errorbarh()`)
- geom\_linerange**  
x, ymin, ymax, alpha, color, linetype, size
- geom\_pointrange**  
x, y, ymin, ymax, alpha, color, fill, linetype, shape, size

#### Maps

`data = data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))`  
`map = map_data("state")`  
`l = ggplot(data, aes(fill = murder))`  
`i = geom_map(aes(map_id = state, map = map)) +  
 expand_limits(x = map$long, y = map$lat)`  
`map_id, alpha, color, fill, linetype, size`

### Three Variables

`seals2 = with(seals, sqrt(delta_long^2 + delta_lat^2))`  
`m = ggplot(seals, aes(long, lat))`

- geom\_contour**(aes(z = z))  
x, y, z, alpha, color, linetype, size, weight

- geom\_raster**(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE)  
x, y, alpha, fill
- geom\_tile**(aes(fill = z))  
x, y, alpha, color, fill, linetype, size

# EXERCISE

Day2\_3\_Visualisationwithggplot2\_Exercise.Rmd

# NUMERICAL - DESCRIPTIVE STATISTICS

## DISTRIBUTIONS

What is a distribution

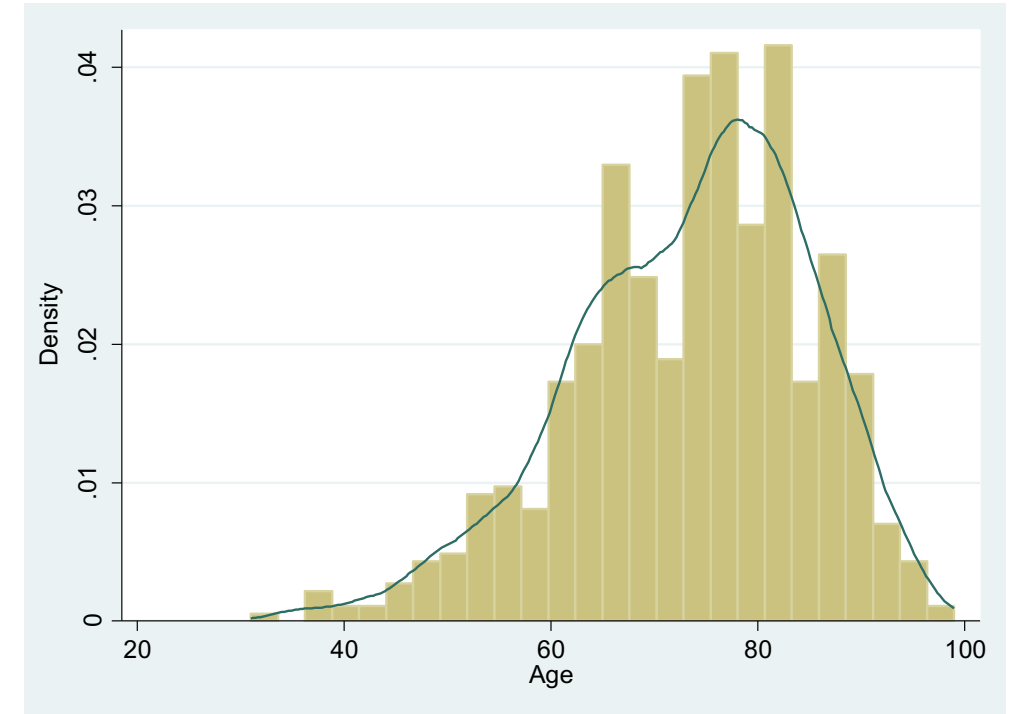
- describes the frequency (or probability) of occurrence for a given value
- describes the shape of the data

Probability distributions for Continuous variables

e.g. Normal, skewed

Frequency distributions for Discrete variables

e.g. Poisson, Binomial



**Vital in determining which statistical tests are applicable**

**PARAMETRIC (based on the specific distributional assumptions) – easy to model**

**NON-PARAMETRIC – no assumptions on distribution – not easy to model**

# TYPES OF DISTRIBUTIONS

## MODALITY

unimodal



bimodal



multimodal



uniform



## SKEWNESS

right skew



left skew



symmetric

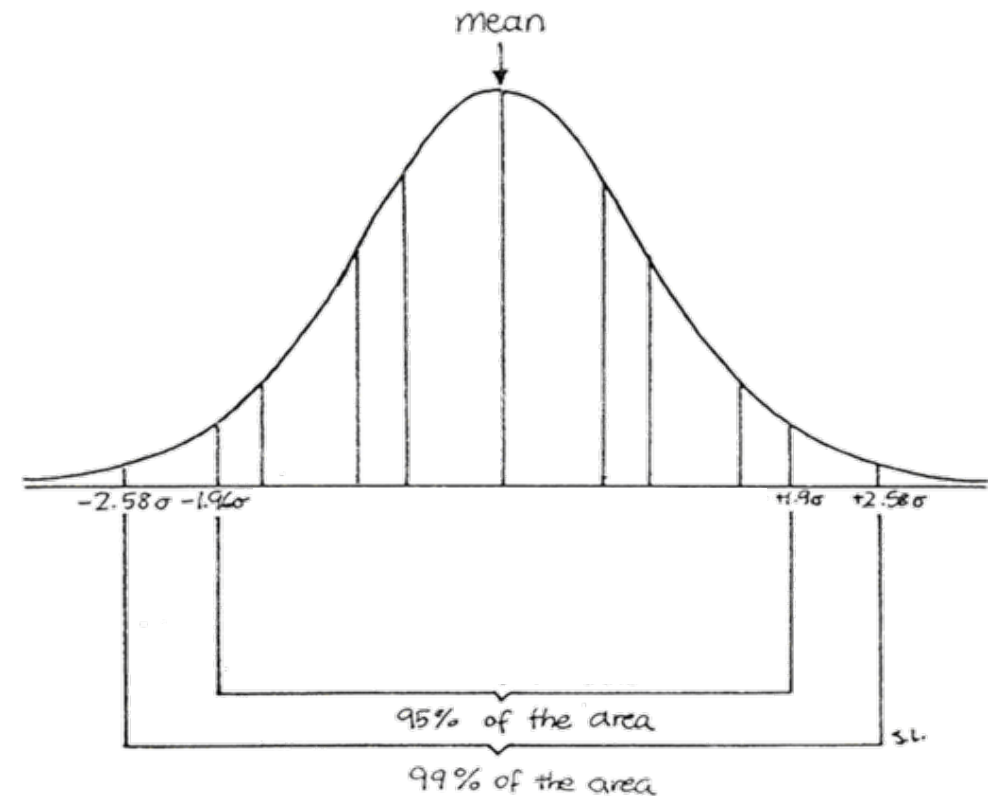


<https://www.openintro.org/book/os/>

# NORMAL DISTRIBUTION

- Unimodal and symmetric, bell shaped curve
- 2 parameters
  - i. MEAN ( $\mu$ ) – measure of central tendency
  - ii. STANDARD DEVIATION ( $\sigma$ ) - measure of spread

**These characteristics allow the use of parametric statistical tests on normal distributions**

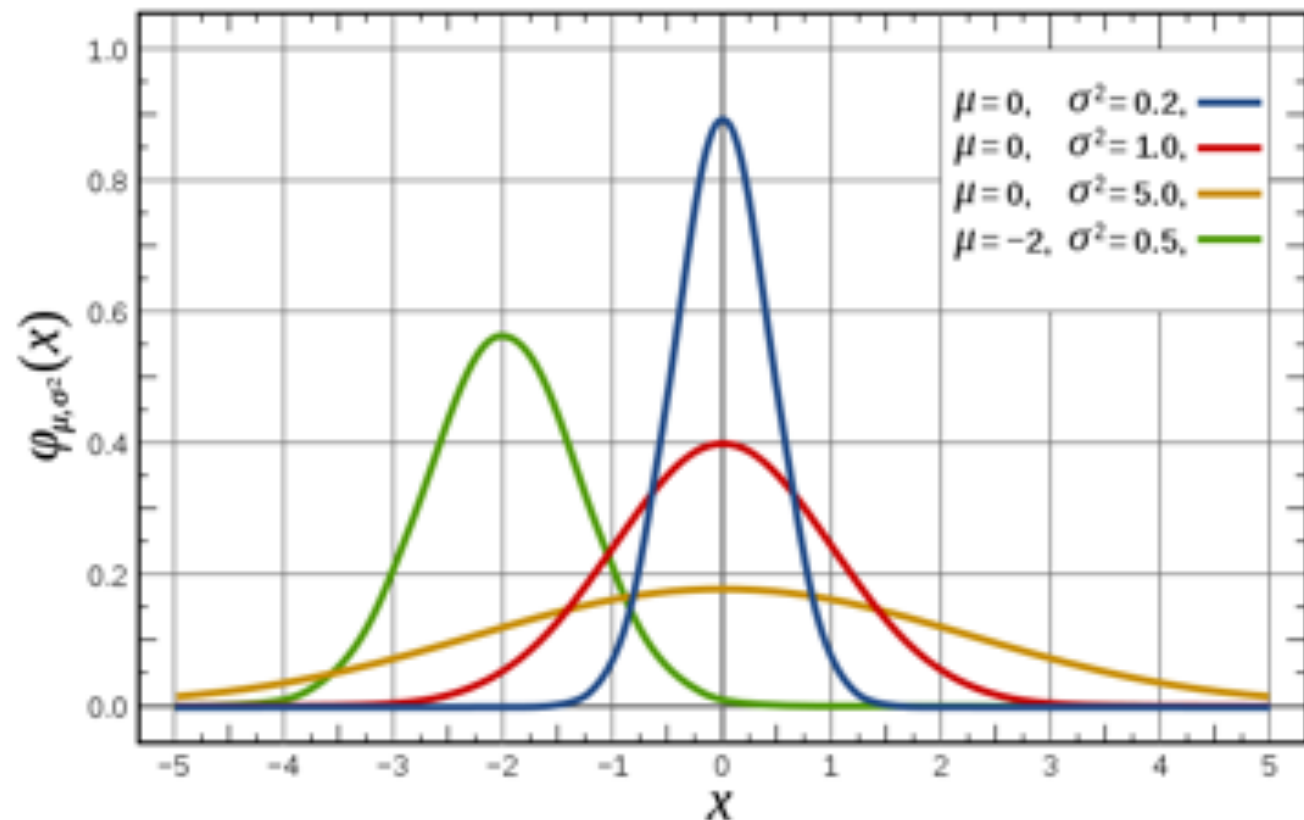




# DIFFERENT NORMAL DISTRIBUTIONS

2 parameters can be different to give different shapes

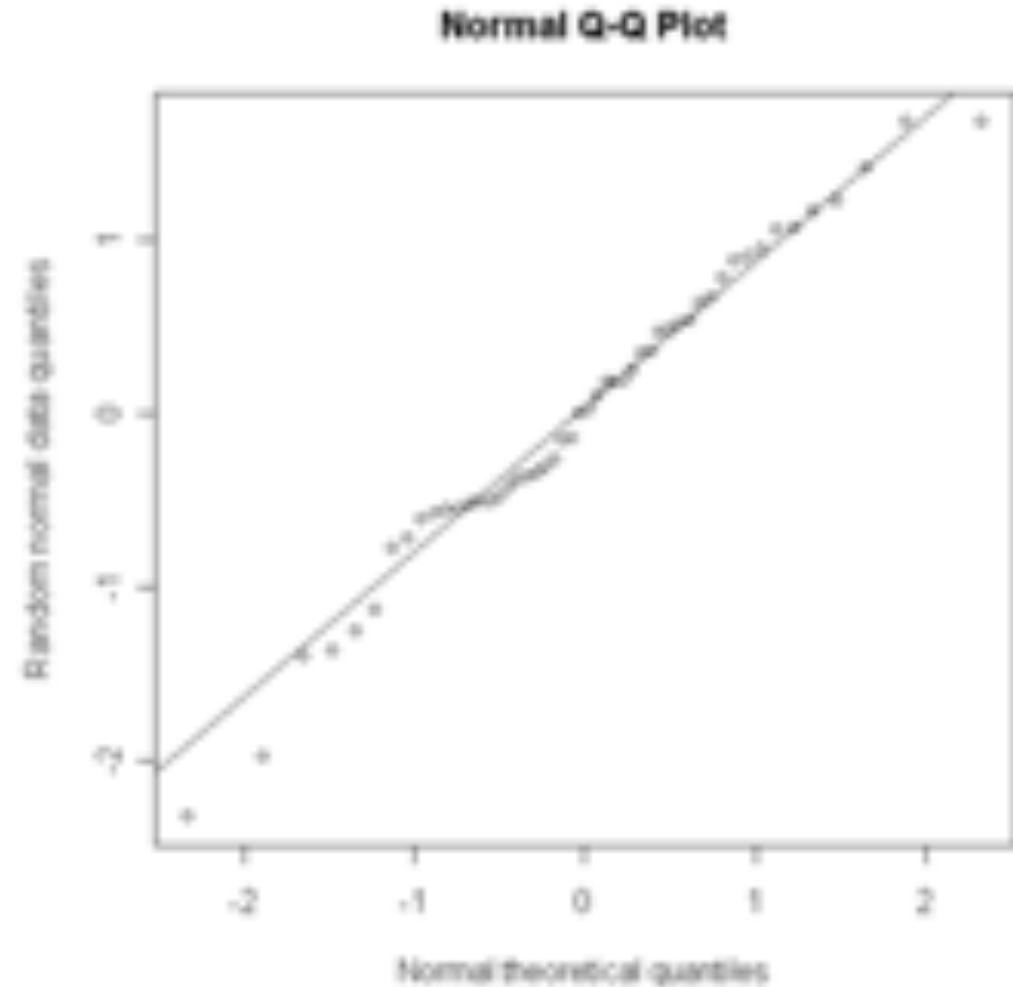
- i. MEAN ( $\mu$ ) – measure of central tendency
- ii. STANDARD DEVIATION ( $\sigma$ ) - measure of spread



# TESTING FOR NORMALITY

## QQ PLOT

Graphically determines if a data sets come from a specified distribution  
e.g. *Normal distribution*



# 4. Quantile-Quantile Plot

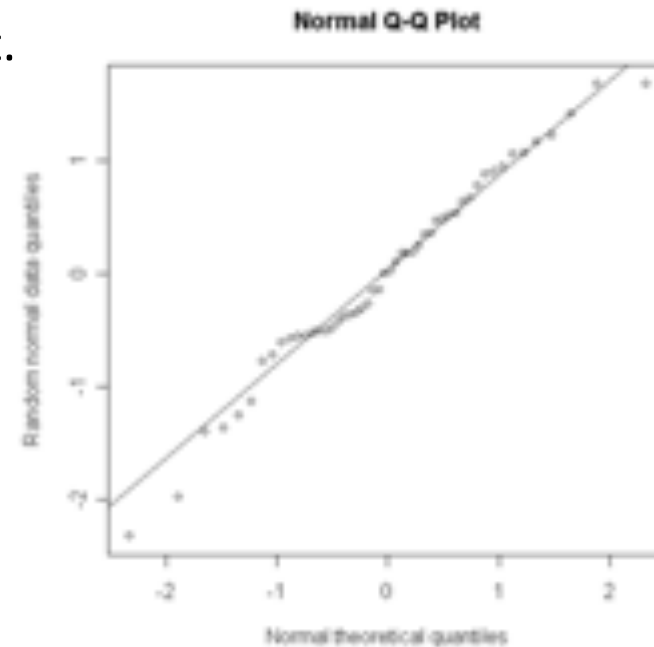
## QQ-plot

- Graphically determines if two data sets come from populations with a common distribution.
- Used to test for normality

Quantile is the fraction (or percent) of points below a given value. i.e. 0.2 quantile has 20% of value below this point and 80% of values above this point.

### Advantage

- the sample sizes do not need to be equal.
- reveals distributional aspects such as shifts in scale, location, change in symmetry.



# NON-NORMAL DISTRIBUTIONS

MEDIAN - value that splits the data in half. 50<sup>th</sup> percentile

Q1 – 1<sup>st</sup> quartile – 25<sup>th</sup> percentile

Q3 – 3<sup>rd</sup> quartile – 75<sup>th</sup> percentile

INTERQUARTILE RANGE (IQR)

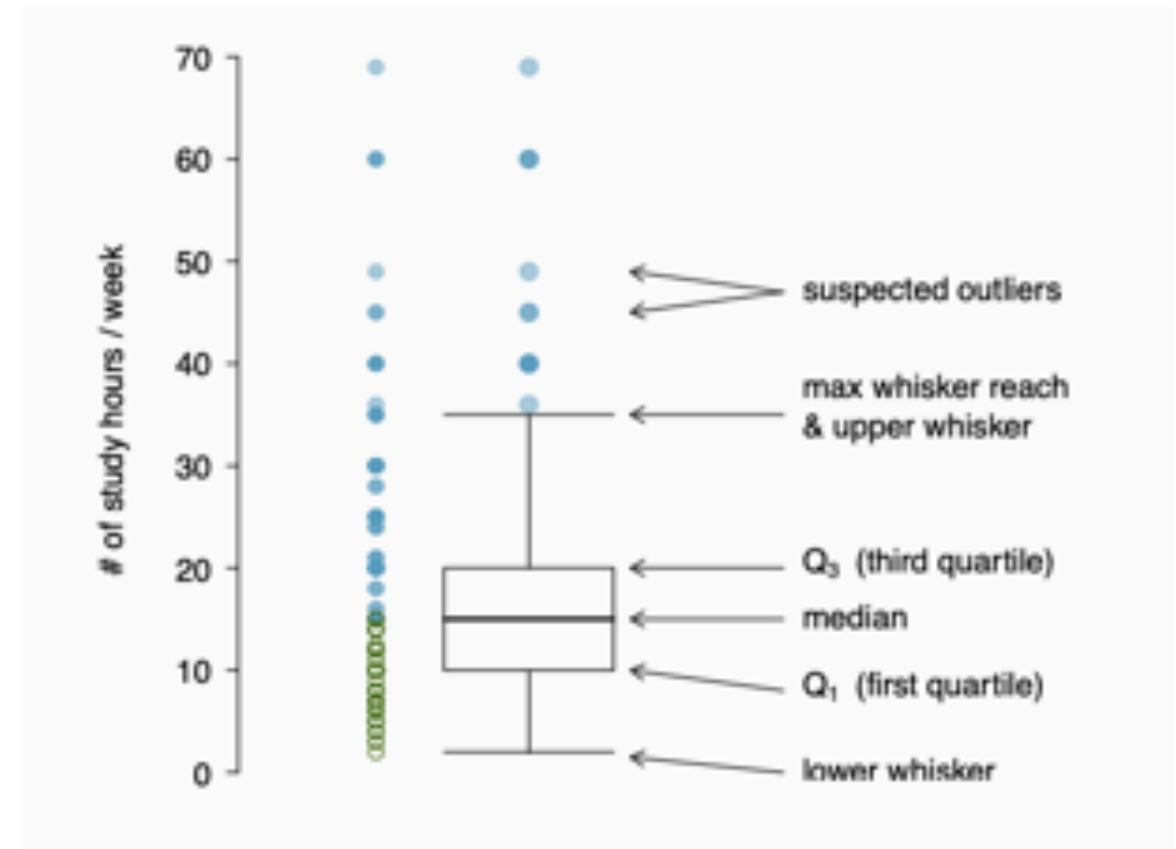
Between Q1 and Q3 is the middle 50% of the data.

$IQR = Q3 - Q1$

OUTLIERS

- Identify extreme skew in the distribution.
- Identify data collection and entry errors.
- Provide insight into interesting features of the data.

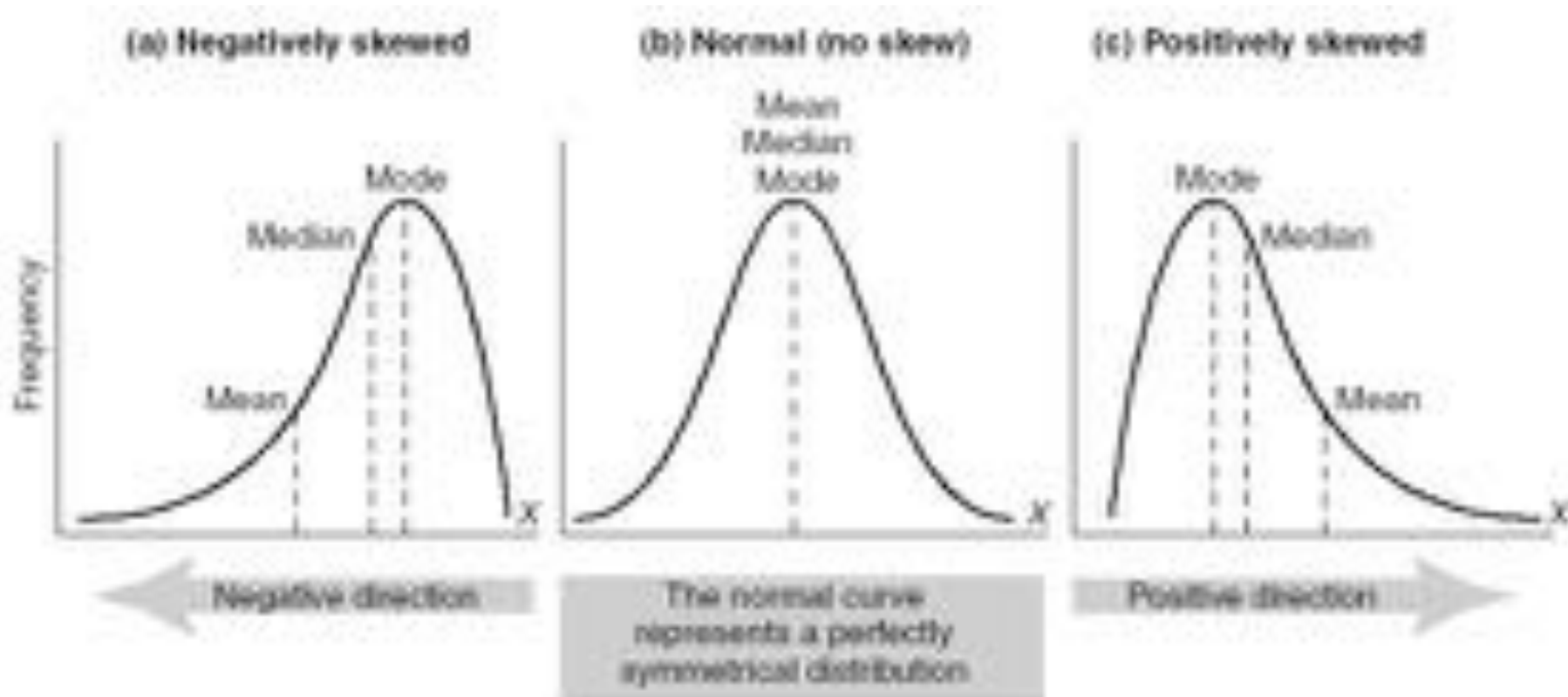
## BOX PLOT



<https://www.openintro.org/book/os/>

# ROBUST STATISTICS

Mean is affected by outliers. Median is more robust



- for symmetric distributions it is more helpful to use the mean and SD to describe the centre and spread
- for skewed distributions it is more helpful to use median and IQR to describe the centre and spread

# TRANSFORMING SKEWED DATA

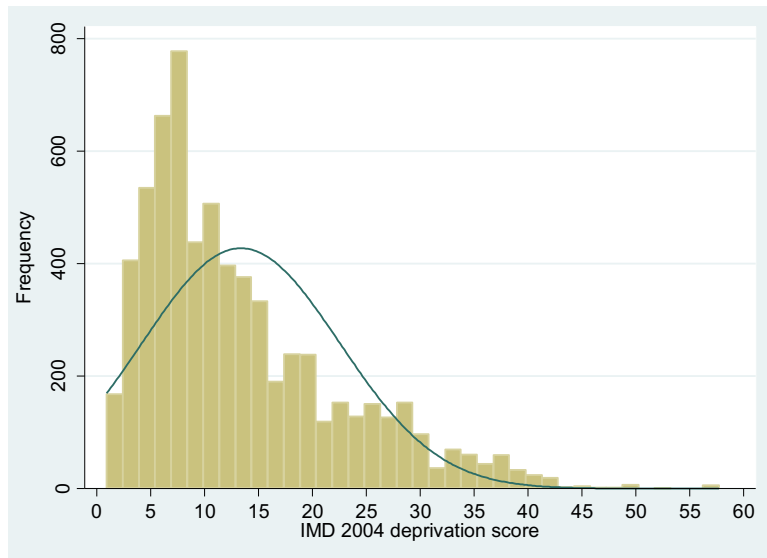
When data are extremely skewed, transforming them could

- make the data normally distributed
- PRO: allows use of parametric statistical tests that make modelling easier
- CON: However, interpretation will be trickier

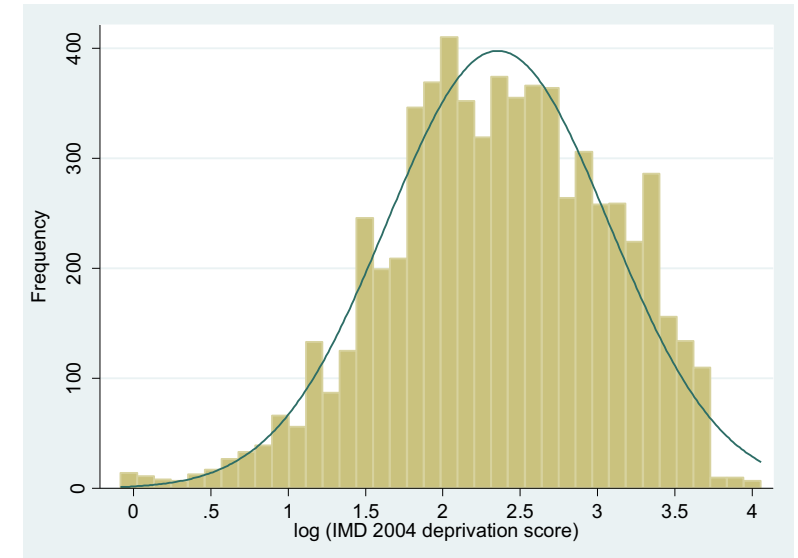
## LOG TRANSFORMATION

A common transformation is the log transformation for positively skewed data

- takes values between  $(0, \infty)$  and converts it to the range  $(-\infty, \infty)$
- the transformed data becomes symmetric about mean (closer to normal)

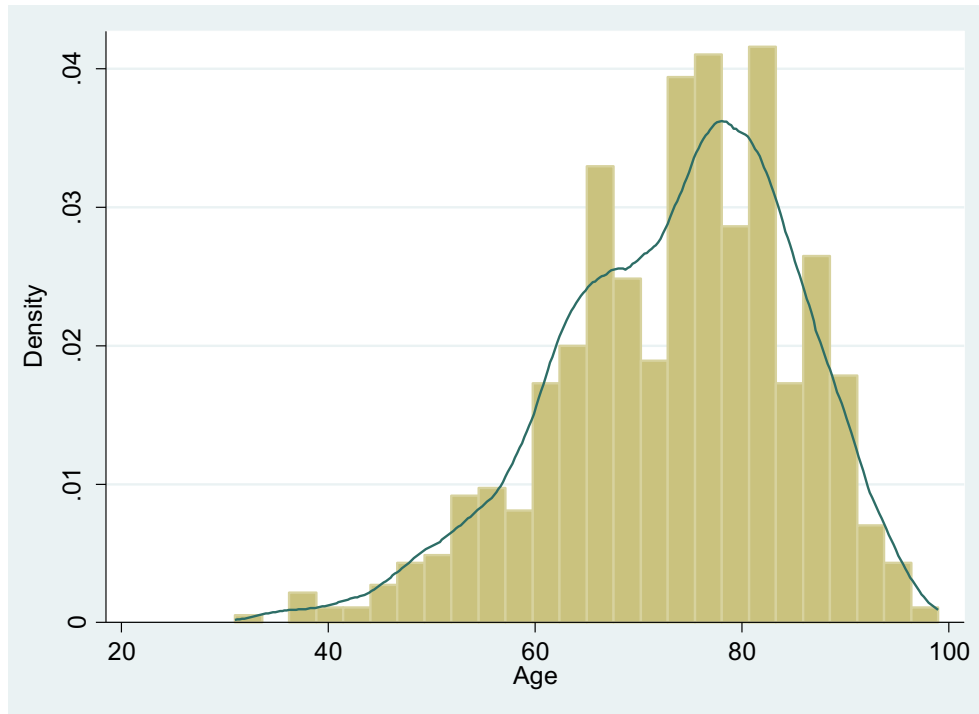


Log Transformation



# SUMMARISE – RECAP

## BAR PLOT – NUMERICAL VARIABLE



Age (years) in patients

## BAR PLOT – CATEGORICAL VARIABLE

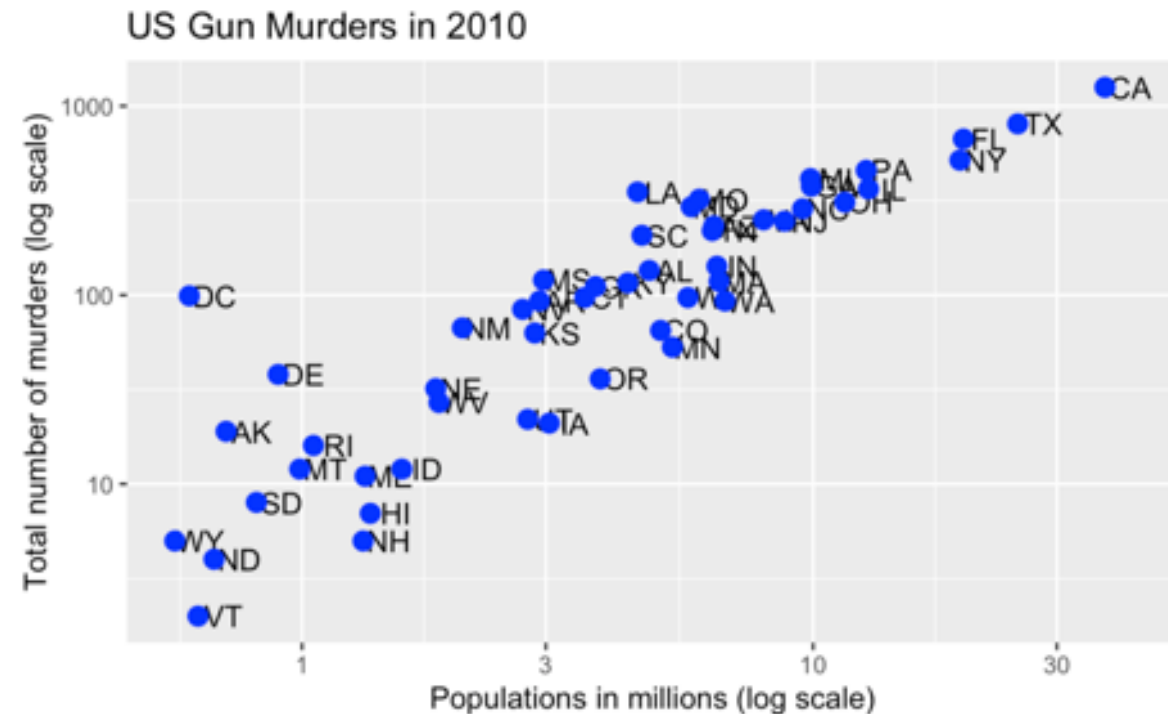


# ASSOCIATION – 2 NUMERICAL VARIABLES

## SCATTER PLOT

are useful for visualizing the relationship between two numerical variables.

Association between total number of murders and population of US states



Appear to be linearly and positively associated: as population increases, total number of murders increases.



# ASSOCIATION – 2 CATEGORICAL VARIABLES

## CONTINGENCY TABLE

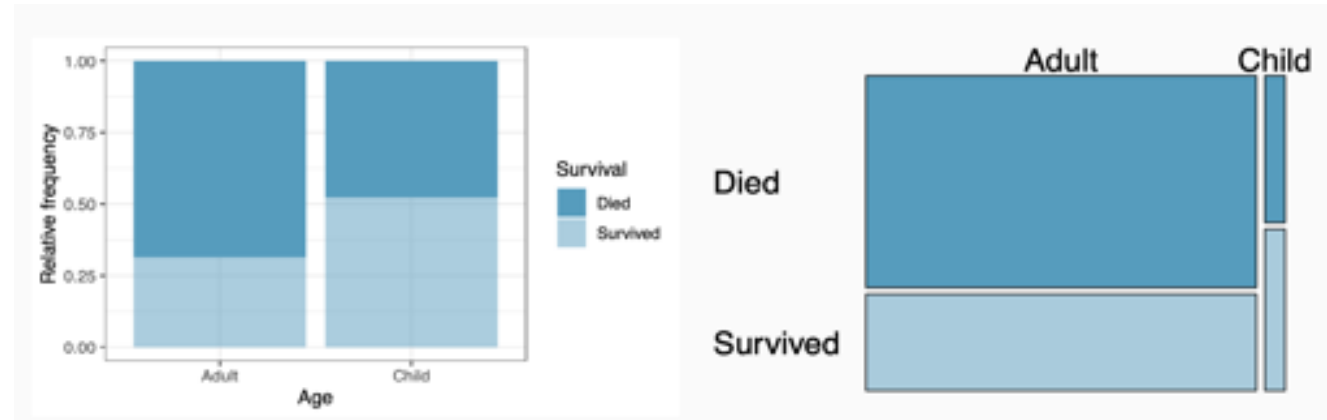
- A table that summarizes data for two categorical variables is called a contingency table.

The contingency table below shows the distribution of survival and ages of passengers on the Titanic.

		Survival		Total
		Died	Survived	
Age	Adult	1438	654	2092
	Child	52	57	109
Total		1490	711	2201

## BAR PLOT/ MOSAIC

- A bar plot is a common way to display a single categorical variable.
- A bar plot where proportions instead of frequencies are shown is called a relative frequency bar plot.
- A mosaic plot has width in proportion to the marginal total (row or column).



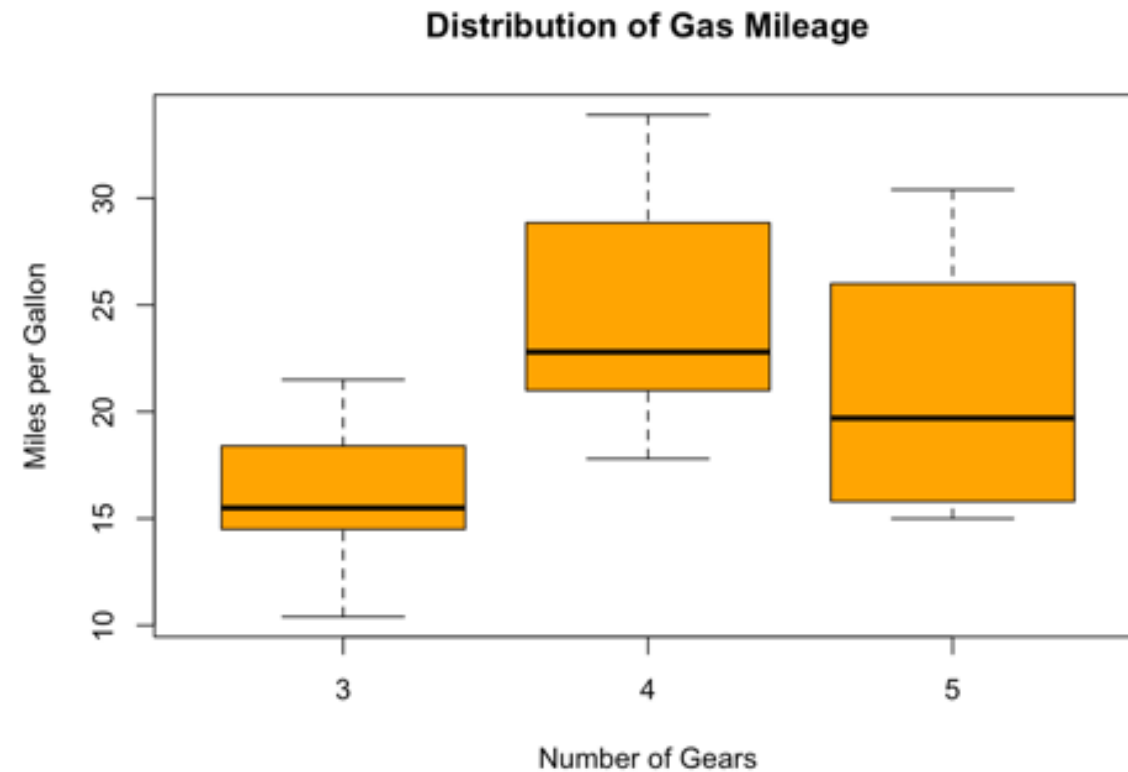
<https://www.openintro.org/book/os/>

# ASSOCIATION – 1 NUMERICAL AND 1 CATEGORICAL VARIABLE

## SIDE-BY-SIDE BOX PLOTS

A **Boxplot** is a method for graphically depicting groups of numerical data through their quartiles

Association between gas mileage and number of gears in the car



# EXERCISE

Day2\_4\_DescriptiveStatistics\_Exercise.Rmd

# DAY 2 AND 3

Data Import

Wrangling

- Tidy + Manipulating
- Summarizing
- Cleaning

Exploration

- Visualization
- Descriptive Statistics

Statistical Inference

- Foundation of inference
- Basic statistical tests
- Linear regression



**Day 2**



**Day 3**