

# Setting up your machine

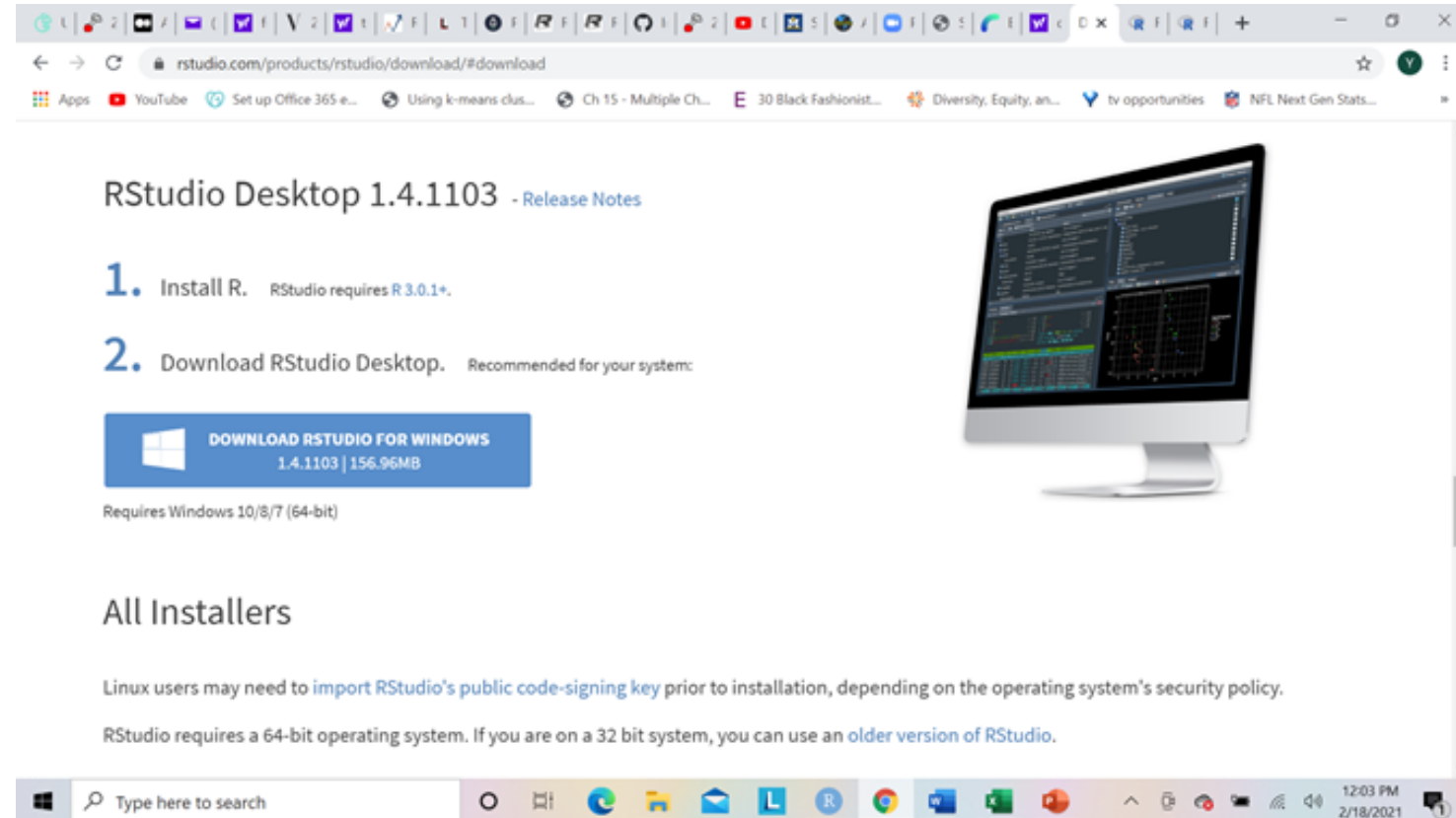
- Download a copy of [R](#) on your local computer from the Comprehensive R Archive Network (CRAN). You can choose between binaries for Linux, Mac and Windows.
- Install one of R's integrated development environment (IDE), [RStudio](#), which makes R coding much easier and faster as it allows you to type multiple lines of code, handle plots, install and maintain packages and navigate your programming environment much more productively.

<https://cran.r-project.org/doc/manuals/R-intro.pdf>

- Download contents located in the Github

<https://github.com/dixond2014/Bevera-R>

- <https://www.r-project.org/>
- <https://rstudio.com/products/rstudio/download/#download>
- <https://cran.r-project.org/mirrors.html>





# Statistical Analysis in R Participant's Guide

March 17<sup>th</sup>, 2021

Instructor: Brian Ashford  
Yvonne Phillips

# About BeVera



- Founded in 2013
- HQ in Metro Atlanta
- Veteran-Owned Small Business
- Data Science Training and Consulting
- Technical Staffing
- Dedicated to making Data Science understandable and usable at every level of the organization
- Helping our customers become self-sufficient

# BK Ashford, Director of Data Science

- Over 22 years of Data Science solution delivery experience
- Analytical Consultant for SAS Institute for over 11 years
- Successful Data Science solution delivery across multiple industries
  - Insurance Carriers
  - Major Airlines
  - Credit Bureaus
  - Retail
- Education:
  - M.Sc. Applied Mathematics, Lehigh University
  - B.S. Mathematics, Morehouse College



# Yvonne Phillips, Instructor

- 18 years of Predictive Analytics work
- Currently Adjunct Professor, Morehouse College – Computer Science/Data Science
- Experienced analytic professional with demonstrated history in:
  - Insurance: Underwriting Auto & Property, Motor Vehicle Reporting Analytics
  - Government: Federal and State/Local government customer solutions needs
  - Financial/Credit Cards
  - Retail
- Education:
  - M.Sc., Decision Science, Georgia State University, Robinson College of Business
  - B.S., Mathematics, Spelman College



# Read article:

Introduction to Research Statistical Analysis.pdf - Adobe Acrobat Reader DC

File Edit View Sign Window Help

Home Tools Introduction to Res... x

71 (1 of 5) 116%

**Education**

## Introduction to Research Statistical Analysis: An Overview of the Basics

Christian Vandever<sup>1</sup>

### Abstract

#### Description

This article covers many statistical ideas essential to research statistical analysis. Sample size is explained through the concepts of statistical significance level and power. Variable types and definitions are included to clarify necessities for how the analysis will be interpreted. Categorical and quantitative variable types are defined, as well as response and predictor variables. Statistical tests described include t-tests, ANOVA and chi-square tests. Multiple regression is also explored for both logistic and linear regression. Finally, the most common statistics produced by these methods are explored.

#### Keywords

statistical analysis; sample size; power; t-test; anova; chi-square; regression

#### Introduction

Statistical analysis is necessary for any research project seeking to make quantitative conclusions. The following is a primer for research-based statistical analysis. It is intended to be a high-level overview of appropriate statistical testing, while not diving too deep into any specific methodology. Some of the

Author affiliations are listed at the end of this article.

Correspondence to:  
Christian Vandever  
HCA Healthcare Graduate  
Medical Education  
2000 Health Park Drive  
Brentwood TN, 37027  
([Christian.Vandever@hca-healthcare.com](mailto:Christian.Vandever@hca-healthcare.com))

Search 'Measure'

Edit PDF

Export PDF

**Adobe Export PDF**

Convert PDF Files to Word or Excel Online

Select PDF File

Introductio...nalysis.pdf

Convert to

Microsoft Word (\*.docx)

Document Language:  
English (U.S.) [Change](#)

Create, edit and sign PDF forms & agreements

[Start Free Trial](#)

# Module 1: R Programming Language



<https://cran.r-project.org/doc/manuals/R-intro.pdf>



RStudio is a four pane work-space for 1) creating file containing R script, 2) typing R commands, 3) viewing command histories, 4) viewing plots and more.

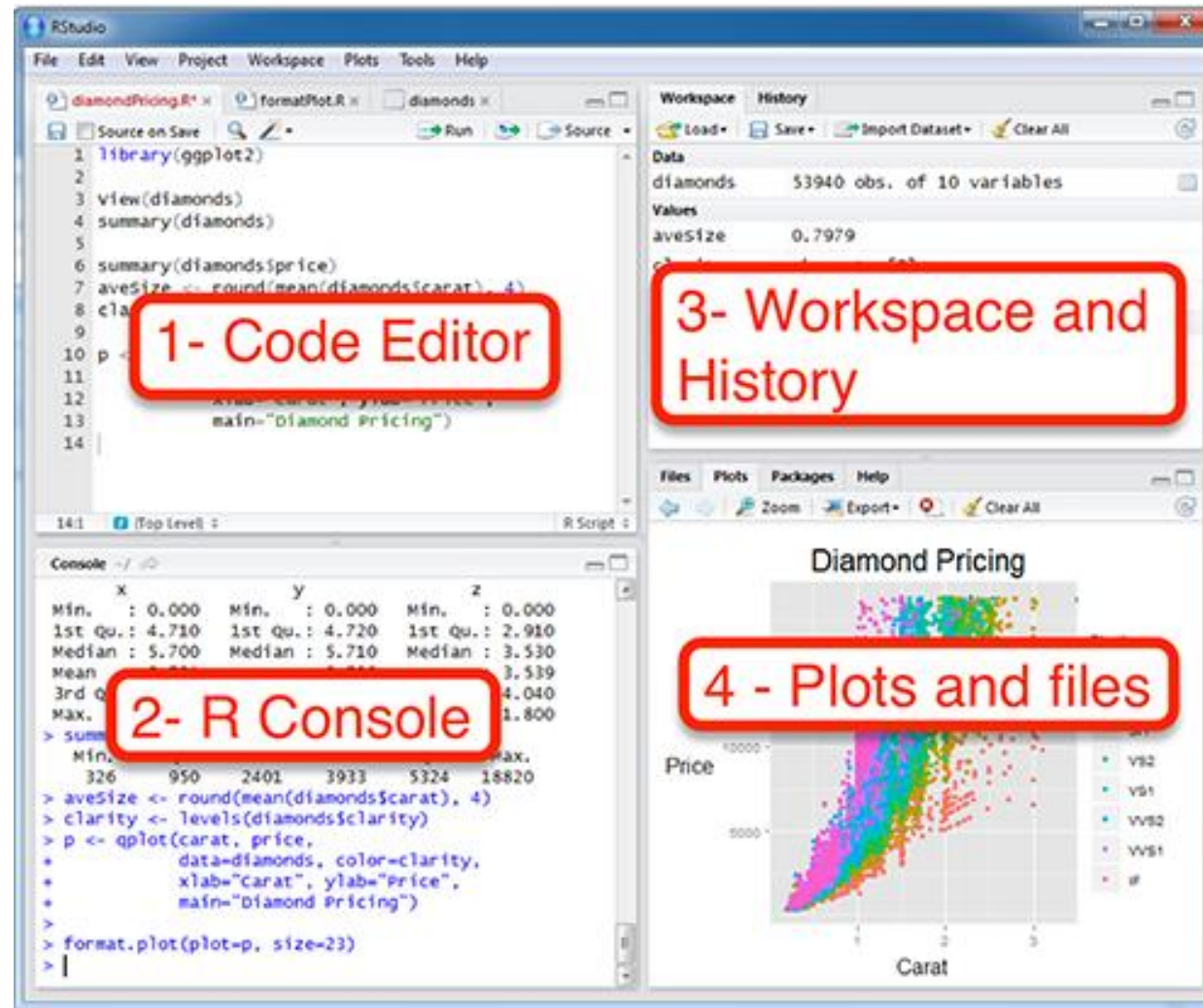
### 1.Top-left panel:

- Code editor allowing you to create and open a file containing R script.

- The R script is where you keep a record of your work. R script can be created as follow:  
File → New → R Script

### 2.Bottom-left panel:

- R console for typing R commands



### 3.Top-right panel:

- Workspace tab: shows the list of R objects you created during your R session
- History tab: shows the history of all previous commands

### 4.Bottom-right panel:

- Files tab: show files in your working directory
- Plots tab: show the history of plots you created. From this tab, you can export a plot to a PDF or an image files
- Packages tab: show external R packages available on your system. If checked, the package is loaded in R.

Labeled key	Ctrl-key combination	Effect
Up arrow	Ctrl-P	Recall previous command by moving backward through the history of commands
Down arrow	Ctrl-N	Move forward through the history of commands
Backspace	Ctrl-H	Delete the character to the left of cursor
Delete	Ctrl-D	Delete the character to the right of cursor
Home	Ctrl-A	Move cursor to the start of the line
End	Ctrl-E	Move cursor to the end the line
Rsish arrow	Ctrl-F	Move cursor right (forward) one character
Left arrow	Ctrl-B	Move cursor left (back) one character
	Ctrl-K	Delete everything from the cursor position to the end of the line.
	Ctrl-U	Clear the whole line and start over
Tab		Name completion (used on some platforms)

# Getting help on a function

> `help(functionname)`

> `args(functionname)`

> `example(functionname)`

# The Working Directory

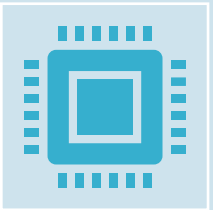


The working directory is the default location for all input and output:

Reading and writing data files

Opening and saving script files

Saving workspace image



From the command line,

`getwd()` – reports the working directory

`setwd()` – changes the working directory

# Search Path –

list of packages currently loaded into the memory

```
> search( )
```

```
[1] ".GlobalEnv"      "tools:rstudio"    "package:stats"  
[4] "package:graphics" "package:grDevices" "package:utils"  
[7] "package:datasets" "package:methods"  "Autoloads"  
[10] "package:base"
```



# Using packages

**1**

```
install.packages("readr")
```

Downloads files to computer

**1 x per computer**

**2**

```
library("readr")
```

Loads package

**1 x per R Session**

## Types

Converting between common data types in R. Can always go from a higher value in the table to a lower value.

as.logical	TRUE, FALSE, TRUE	Boolean values (TRUE or FALSE).
as.numeric	1, 0, 1	Integers or floating point numbers.
as.character	'1', '0', '1'	Character strings. Generally preferred to factors.
as.factor	'1', '0', '1', Levels: '1', '0'	Character strings with preset levels. Needed for some statistical models.

## Maths Functions

log(x)	Natural log.	sum(x)	Sum.
exp(x)	Exponential.	mean(x)	Mean.
max(x)	Largest element.	median(x)	Median.
min(x)	Smallest element.	quantile(x)	Percentage quantiles.
round(x, n)	Round to n decimal places.	rank(x)	Rank of elements.
signif(x, n)	Round to n significant figures.	var(x)	The variance.
cor(x, y)	Correlation.	sd(x)	The standard deviation.

## Variable Assignment

```
> a <- 'apple'
> a
[1] 'apple'
```

## The Environment

ls()	List all variables in the environment.
rm(x)	Remove x from the environment.
rm(list = ls())	Remove all variables from the environment.

You can use the environment panel in RStudio to browse variables in your environment.

## Matrices

```
m <- matrix(x, nrow = 3, ncol = 3)
# Create a matrix from x.
```



m[2, ] - Select a row



m[, 1] - Select a column



m[2, 3] - Select an element

t(m)

Transpose

m %\*% n

Matrix Multiplication

solve(m, n)

Find x in: m \* x = n

## Lists

```
l <- list(x = 1:5, y = c('a', 'b'))
# A list is a collection of elements which can be of different types.
```

l[[2]]

Second element of l.

l[[1]]

New list with only the first element.

l\$x

Element named x.

l['y']

New list with only element named y.

Also see the **dplyr** package.

## Data Frames

```
df <- data.frame(x = 1:3, y = c('a', 'b', 'c'))
# A special case of a list where all elements are the same length.
```

x	y
1	a
2	b
3	c

### List subsetting

df\$x



df[[2]]



Understanding a data frame

View(df)

See the full data frame.

head(df)

See the first 6 rows.

### Matrix subsetting

df[, 2]



df[2, ]



df[2, 2]



nrow(df)  
Number of rows.

ncol(df)  
Number of columns.

dim(df)  
Number of columns and rows.

cbind - Bind columns.



rbind - Bind rows.



## Strings

Also see the **stringr** package.

paste(x, y, sep = ' ')	Join multiple vectors together.
paste(x, collapse = ' ')	Join elements of a vector together.
grep(pattern, x)	Find regular expression matches in x.
gsub(pattern, replace, x)	Replace matches in x with a string.
toupper(x)	Convert to uppercase.
tolower(x)	Convert to lowercase.
nchar(x)	Number of characters in a string.

## Factors

factor(x)	Turn a vector into a factor. Can set the levels of the factor and the order.
cut(x, breaks = 4)	Turn a numeric vector into a factor by 'cutting' into sections.

## Statistics

lm(y ~ x, data=df) Linear model.	t.test(x, y) Perform a t-test for difference between means.	prop.test Test for a difference between proportions.
glm(y ~ x, data=df) Generalised linear model.	pairwise.t.test Perform a t-test for paired data.	aov Analysis of variance.
summary Get more detailed information out a model.		

## Distributions

	Random Variates	Density Function	Cumulative Distribution	Quantile
Normal	rnorm	dnorm	pnorm	qnorm
Poisson	rpois	dpois	ppois	qpois
Binomial	rbinom	dbinom	pbinom	qbinom
Uniform	runif	dunif	punif	qunif

## Plotting

Also see the **ggplot2** package.



plot(x)  
Values of x in order.



plot(x, y)  
Values of x against y.



hist(x)  
Histogram of x.

## Dates

See the **lubridate** package.



# Two Branches of Statistics

## Descriptive Statistics

- as a science, involves the collection, organization, summarization, and presentation of data
- involves raw data, as well as graphs, tables, and numerical summaries
- “Just the facts”
- Refer to sample without making any assumptions about the population

## Inferential Statistics

- as a science, involves using descriptive statistics to estimate population parameters
- deals with interpretation of the information collected
- usually used in conjunction with descriptive statistics within a statistical study



# Data Structures in



VECTORS

1

MATRIX

2

ARRAY

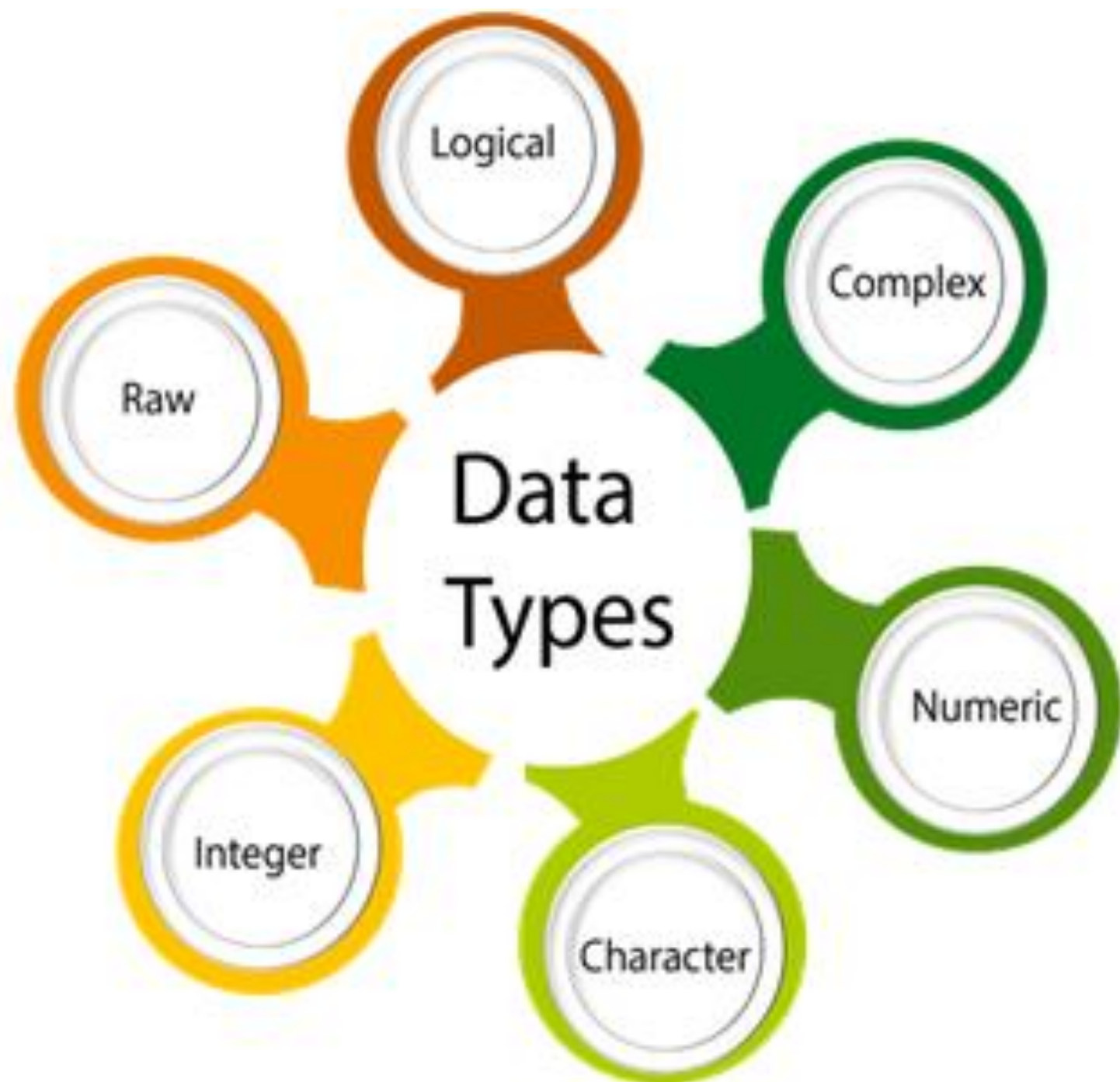
3

LIST

4

DATA FRAME

5



# Data Types in R

```
> typeof(letters)
```

```
> typeof(1:10)
```

```
> x <- list("a", "b", 1:10)
```

```
> length(x)
```

## Vectors

### Atomic vectors

Logical

Numeric

Integer

Double

Character

List

NULL

# Basic Arithmetic Operators



Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
$\wedge$ or **	Exponentiation

# Examples:

# A multiplication

$3*9$

Output: ## [1] 27

# A division

$(5+13)/2$

Output: ## [1] 9

# Exponentiation

$-3^2$

Output: ## [1] 9

# Logical Operators return values inside the vector based on logical conditions.

Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Exactly equal to
!=	Not equal to
!x	Not x
x	y
x & y	x AND y
isTRUE(x)	Test if X is TRUE

You can add many conditional statements, but we need to include them in a parenthesis. Follow this structure to create a conditional statement:

```
variable_name[(conditional_statement)]
```

# Example:

# Create a vector from 1 to 8

```
logical_vector <- c(1:8)
```

```
logical_vector > 6
```

Output: ## [1] FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE

# Example:

```
logical_vector <- c(1:8)
```

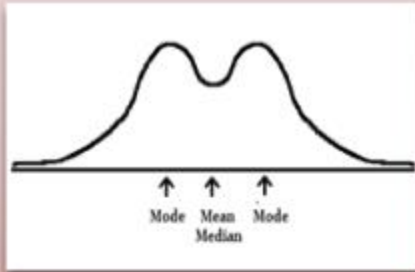
```
logical_vector[(logical_vector>3) & (logical_vector<5)]
```

Output: ## [1] 4



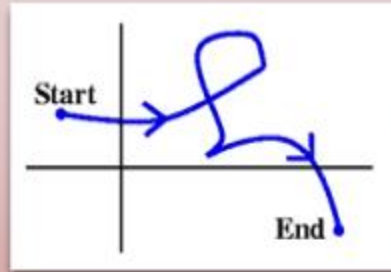
# Module 2: Descriptive Statistics





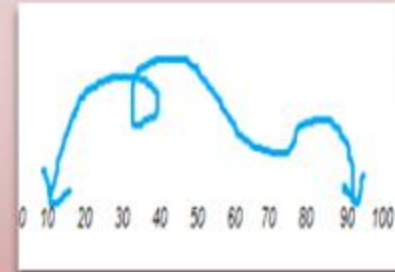
### Central Tendency

Mean, Median, Mode, Outliers



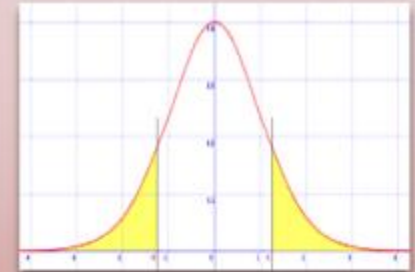
### Measures of Spread

Range, Standard deviation, Variance, Quartiles



### Percentiles

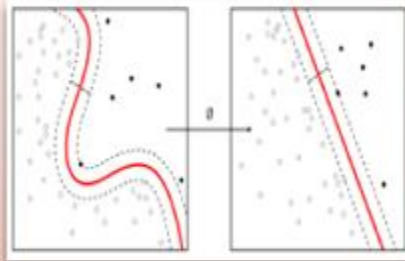
Position of data, percentile rank, percentile range



### Probability Distributions:

Uniform, normal (Gaussian), Poisson

## Basic Probability and Statistics



### Dimensionality reduction

Pruning, PCA



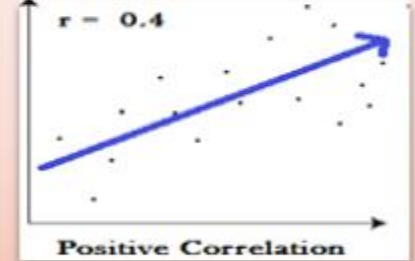
### Sampling

SRS, Reservoir, Undersampling, Oversampling,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

### Bayesian statistics

Measuring belief or confidence



### Covariance & correlation

How data is related

## More Advanced Probability and Statistics

## Descriptive Statistics

Variable	<u>Obs</u>	Mean	<u>Std.Dev.</u>	Min	Max
price	74	6165.257	2949.496	3291	15906
mpg	74	21.297	5.786	12	41
rep78	69	3.406	.99	1	5
headroom	74	2.993	.846	.846	5
trunk	74	13.757	4.277	5	23
weight	74	3019.459	777.194	1760	4840
length	74	187.932	22.266	142	233
turn	74	39.649	4.399	31	51
displacement	74	197.297	91.837	79	425
<u>gear_ratio</u>	74	3.015	.456	2.19	3.89
foreign	74	.297	.46	0	1

# R Functions for Computing Descriptive Statistics

Description	R function
Mean	<code>mean()</code>
Standard deviation	<code>sd()</code>
Variance	<code>var()</code>
Minimum	<code>min()</code>
Maximum	<code>maximum()</code>
Median	<code>median()</code>
Range of values (minimum and maximum)	<code>range()</code>
Sample quantiles	<code>quantile()</code>
Generic function	<code>summary()</code>
Interquartile range	<code>IQR()</code>

# Measure of Central Tendency

Mean	Average value
Median	Middle value
Mode	Most frequent value

When to use mean, median and mode?

- Mean – When your data is not skewed i.e., normally distributed. In other words, there are no extreme values present in the data set.
- Median – When your data is skewed or you are dealing with ordinal (ordered categories) data (e.g., Likert scale 1. Strongly dislike 2. Dislike 3. Neutral 4. Like 5. Strongly like)
- Mode - When dealing with nominal (unordered categories) data.

# Measures of Dispersion:

Measures of variability gives how “spread out” the data

Range	Difference between max and min in a distribution
Interquartile range	Correspondes to the difference between the first and third quartiles
Standard Deviation	Average distance of scores in a distribution from their mean
Variance	Square of the standard deviation
Skewness	Degree to which scores in a distribution are spread out
Kurtosis	Flatness of peakness of the curve

# More Vector Arithmetic Statistical operations on numeric vectors

- In studying data, you will make frequent use of sum, which gives the sum of the entries, max, min, mean.

Function	Example	Result
sum(x), product(x)	sum(1:20)	210
min(x), max(x)	min(1:20)	1
mean(x), median(x)	mean(1:20)	10.5
sd(x), var(x), range(x)	sd(1:20)	5.91608
quantile(x, probs)	quantile(1:20, probs = .2)	20%, 4.8
summary(x)	summary(1:20)	Min = 1.00. 1st Qu. = 5.75, Median = 10.50, Mean = 10.50, 3rd Qu. = 15.25, Max = 20.0

> (i.e.,  $\text{Sum}((x - \text{mean}(x))^2)/(\text{length}(x) - 1)$ )

- A useful function for quickly getting properties of a vector: summary(y)

# More Vector Arithmetic Statistical operations on continuous vectors

Function	Description	Example	Result
<code>round(x, digits)</code>	Round elements in x to digits digits	<code>round(c(3.712, 3.1415), digits = 1)</code>	3.7, 3.1
<code>ceiling(x), floor(x)</code>	Round elements x to the next highest (or lowest) integer	<code>ceiling(c(2.6, 8.1))</code>	3, 9
<code>x %% y</code>	Modular arithmetic (ie. $x \bmod y$ )	<code>8 %% 4</code>	0

# Vector Arithmetic Statistical operations on discrete vectors

Function	Description	Example	Result
unique(x)	Returns a vector of all unique values.	unique(c(3, 3, 4,5,12))	3, 4, 5, 12
table(x, exclude)	Returns a table showing all the unique values as well as a count of each occurrence. To include a count of NA values, include the argument exclude = NULL	table(c("x", "x", "y", "z"))	x y z 2 1 1



# Descriptive Statistics in R for Data Frames

---

`Max(frame)` – Returns the largest value in the entire data frame.

---

`Min(frame)` – Returns the smallest value in the entire data frame.

---

`Sum(frame)` – Returns the sum of the entire data frame.

---

`Fivenum(frame)` – Returns the Tukey summary values for the entire data frame.

---

`Length(frame)` – Returns the number of columns in the data frame.

---

`Summary(frame)` – Returns the summary for each column.

# Return the First and/or Last Parts of an Object

Checking your data: head() and tail()

Shows rows from the head and tail of a data frame or matrix.

headtail(x, n = 3L, which = NULL, addrownums = TRUE, ...)

> head()

> tail()

> headtail(iris)

> headtail(iris,10)

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
148	6.5	3.0	5.2	2.0	virginica
149	6.2	3.4	5.4	2.3	virginica
150	5.9	3.0	5.1	1.8	virginica

# Computing an overall summary

## Summary of a single variable.

Summary() : Provides back five values.

```
> summary(cars$speed)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4.0   12.0   15.0   15.4   19.0   25.0
```

## Summary of a data frame.

Summary() : The function automatically is applied to each column

```
> summary(cars)
  Min. speed      Min. dist
   4.0      : 4.0    2.00
 1st Qu.:12.0    1st Qu.: 26.00
 Median :15.0    Median : 36.00
  Mean  :15.4     Mean  : 42.98
3rd Qu.:19.0    3rd Qu.: 56.00
  Max.  :25.0     Max.   :120.00
```



# R Row Summary Commands

The row summary commands in R work with row data.

- *rowmeans()* command gives the mean of values in the row
- while *rowsums()* command gives the sum of values in the row.



# R Special Summary Commands

There are two types of special summary commands:

- Row Summary Commands – Applied to work with row data. Two commands here are *rowmeans()* and *rowsums()*.
- Column Summary Commands – Also, applied to work with row data but the two commands here are *colmeans()* and *colsums()*.

Q1	Q2	Q3	Q4	Q5
0	0	0	1	1
0	1	1	1	0
1	0	0	1	1
0	0	1	1	0
1	1	1	1	1

```
quiz <- data.frame("q1" = c(0, 0, 1, 0, 1),
  "q2" = c(0, 1, 0, 0, 1),
  "q3" = c(0, 1, 0, 1, 1),
  "q4" = c(1, 1, 1, 1, 1),
  "q5" = c(1, 0, 1, 0, 1))
```

```
rowMeans(quiz)
rowSums(quiz)
colMeans(quiz)
colSums(quiz)
```

# Descriptive Statistics in R for Matrix Objects

Let's create a 4 by 5 matrix

```
> set.seed(231)
```

```
> mat <- matrix(rnorm(50), nrow=4, ncol=5)
```

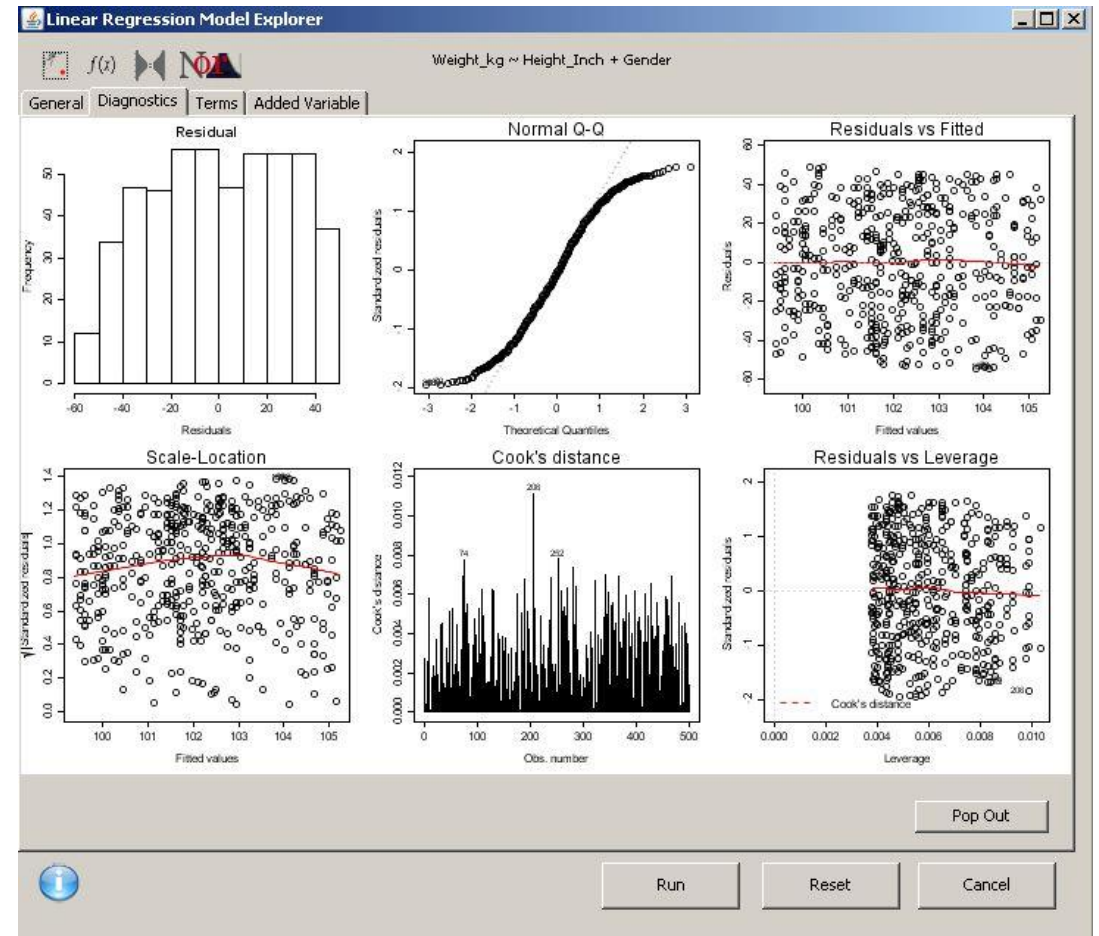
	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	-0.5331	-0.47336	0.56369	-1.18714	1.005845
[2,]	-2.31166	0.217397	-0.03433	-0.57976	-0.52605
[3,]	-0.9542	0.062922	-0.22631	1.401331	-0.1247
[4,]	0.262516	-0.87783	1.386578	0.319622	1.650254

```
> mean(mat[,2]) <- column
```

```
> mean(mat[2,]) <- row
```

# Deducer package: frequencies

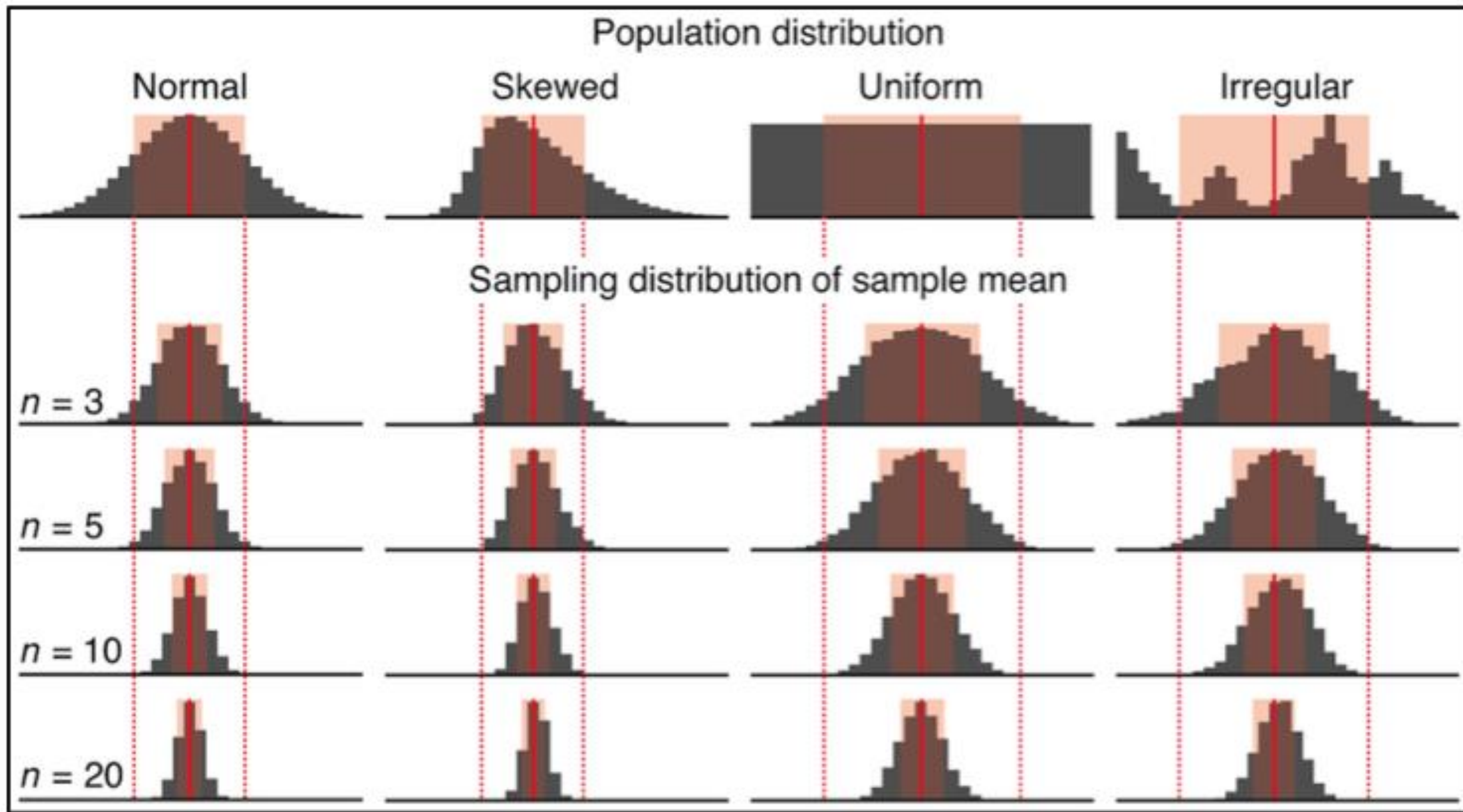
- Adds menu driven analysis and plotting
  - `install.packages("Deducer")`
  - `library("Deducer")`
- frequency functions



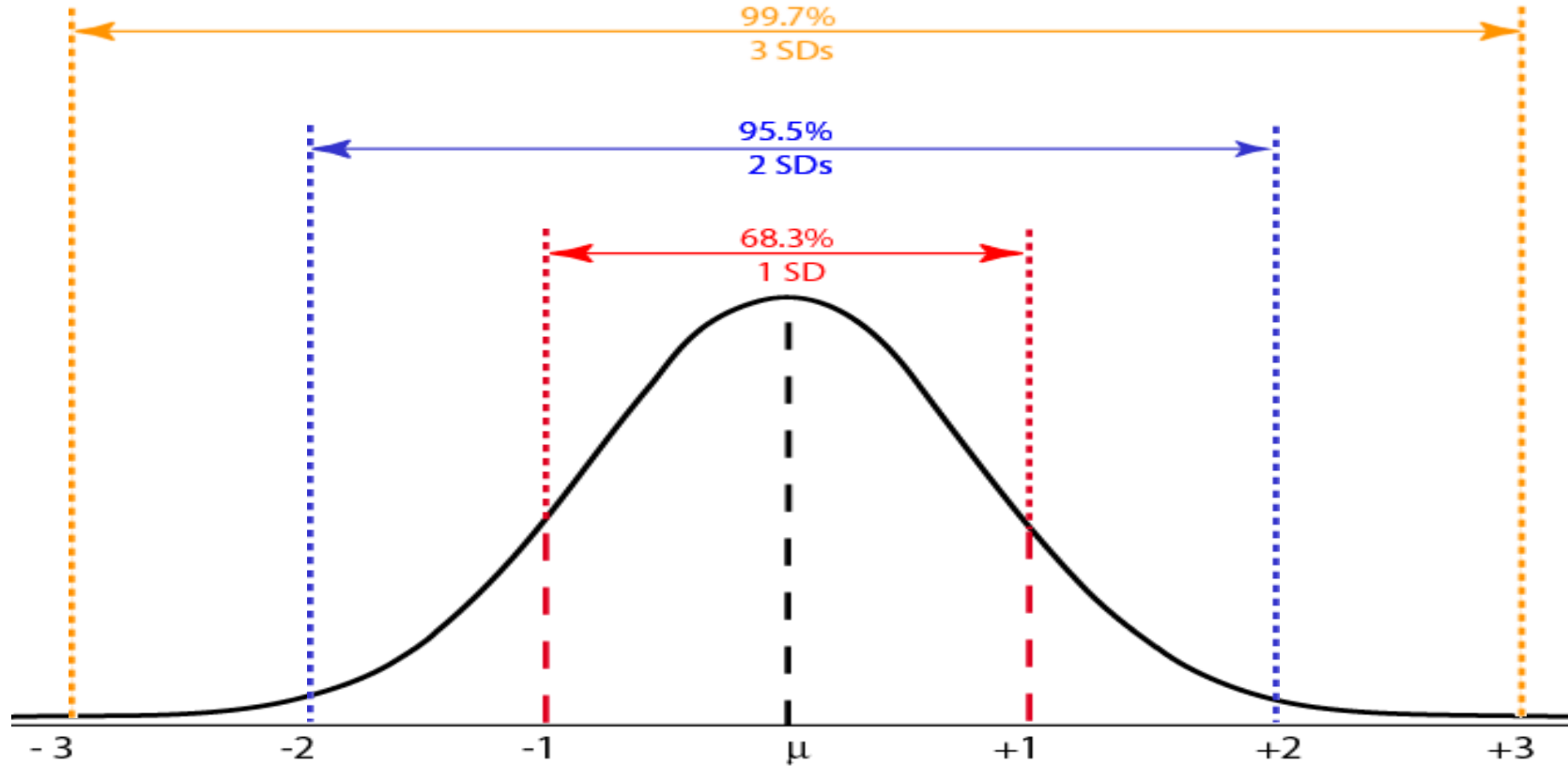


- `install.packages("Hmisc")`
- `library("Hmisc")`
- Similar to the `summary` function is the `describe` function.

41



# Empirical Rule



- 68% of data lie within  $1\sigma$  of the mean  $\mu$
- 95% of data lie within  $2\sigma$  of the mean  $\mu$
- 99.7% of data lie within  $3\sigma$  of the mean  $\mu$

# Random Variables

1. Generate random numbers from uniform distribution

**runif**(n, min=a, max=b)

2. Generate random numbers from normal distribution

**rnorm**(n, mean=a, sd=b)

3. Generate random numbers from binomial distribution

**rbinom** (# observations, # trials/observation, probability of success )

4. Generate random numbers from bernoulli distribution

**rbinom**(10, 1,.5)

# Methods of Standardization and Normalization

S.NO.	Normalisation	Standardisation
1	Minimum and maximum value of features are used for scaling	Mean and standard deviation is used for scaling.
2	It is used when features are of different scales.	It is used when we want to ensure zero mean and unit standard deviation.
3	Scales values between $[0, 1]$ or $[-1, 1]$ .	It is not bounded to a certain range.
4	It is really affected by outliers.	It is much less affected by outliers.
5	This transformation squishes the n-dimensional data into an n-dimensional unit hypercube.	It translates the data to the mean vector of original data to the origin and squishes or expands.
6	It is useful when we don't know about the distribution	It is useful when the feature distribution is Normal or Gaussian.
7	It is an often called as Scaling Normalization	It is an often called as Z-Score Normalization.

# Methods of Standardization and Normalization

1. Z-score:  $z = \frac{x - \text{mean}}{\text{std.dev}}$

2. Min-Max Scaling:  $x - \min(x) / \max(x) - \min(x)$

3. Standard Deviation Method:  $x / \text{stdev}(x)$

4. Range:  $x / (\max(x) - \min(x))$

5. Centering: Subtracting a constant value from every value of a variable. The constant value can be average, min or max.

## *# Creating a sample data*

```
set.seed(272)  
X = data.frame(k1 = sample(100:1000, 1000, replace=TRUE),  
               k2 = sample(10:100, 1000, replace=TRUE))  
X.scaled = scale(X, center= TRUE, scale=TRUE)
```

*#Check Mean and Variance of Standardized Variable*

```
colMeans(X.scaled)  
var(X.scaled)
```

# Exercise: Cars

- Load sample dataset: cars
- Basic scale() command description
- Standardize data in R
- Visualization of standardized data in R



# Standard probability density function for the binomial distribution:

#If we flip a fair coin 10 times, what is the probability of getting exactly 5 heads? (a fair coin ( $p(\text{head})=.5$ ))

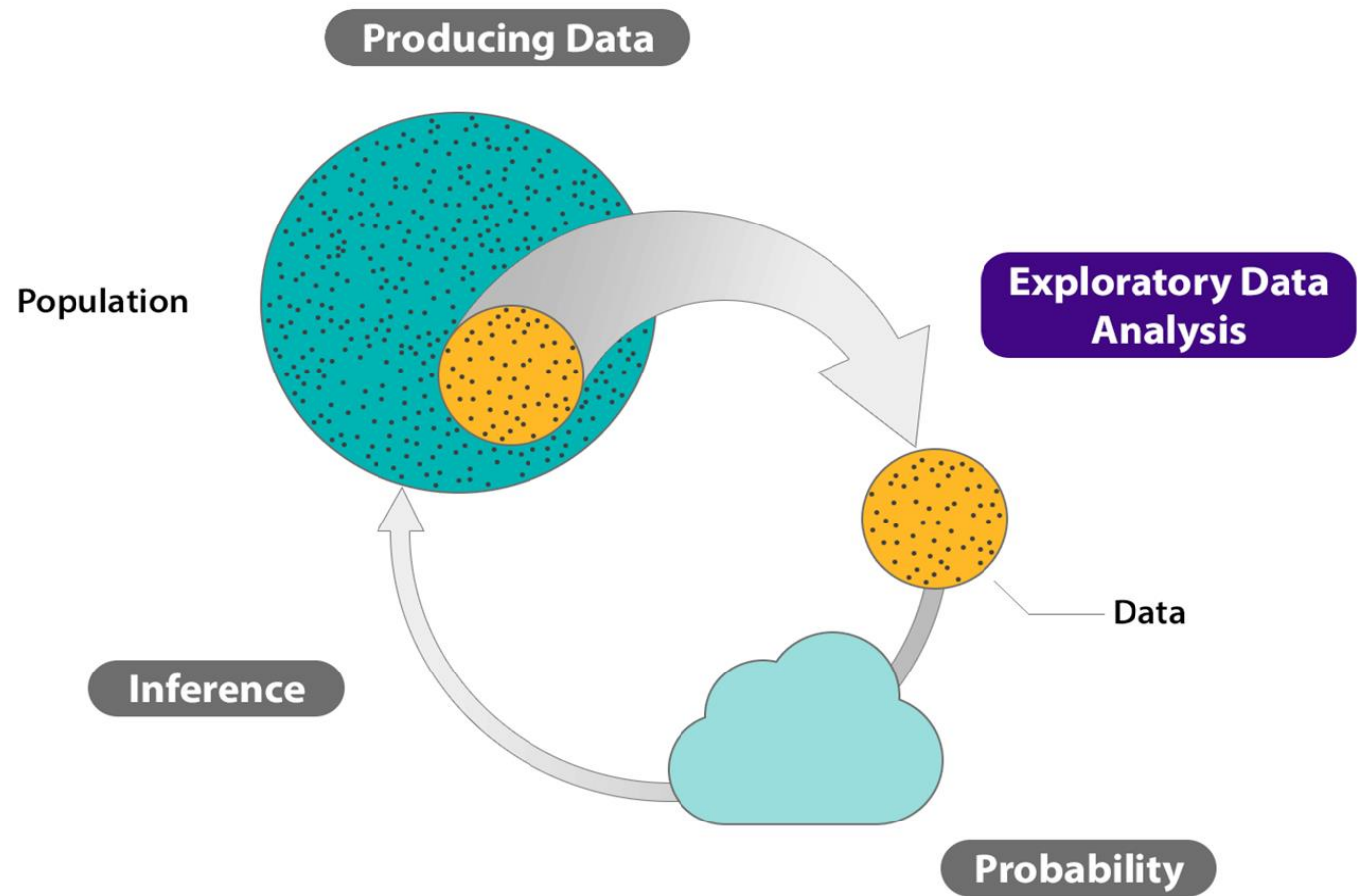
```
> dbinom(5, size=10, prob=0.5) #calculate binomial probability
```

# cumulative probability of getting X successes

# If we flip a fair coin 10 times, what is the probability of getting 5 or less heads?

```
> pbinom(5,10,0.5)
```

There is a difference between pbinom and dbinom!!



**Exploratory Data Analysis:** an approach to analyzing data sets to summarize their main characteristics, often with visual methods. - Wikipedia



# “Get to Know” the dataset

- Doing so upfront will make the rest of the project much smoother, in 3 main ways:
  1. You’ll gain valuable hints for [Data Cleaning](#).
  2. You’ll think of ideas for [Feature Engineering](#).
  3. You’ll get a "feel" for the dataset, which will help you communicate results and deliver greater impact.
- EDA should be **quick, efficient, and decisive**... not long and drawn out!
- You see, there are infinite possible plots, charts, and tables, but you only need a **handful** to "get to know" the data well enough to work with it.

# What is EDA?

An approach for data analysis that employs a variety of techniques

1. Maximize insight into a data set
2. Uncover underlying structure
3. Extract important variables
4. Detect outliers and anomalies
5. Test underlying assumptions
6. Develop parsimonious models and
7. Determine optimal factor settings

# EDA is a data approach.

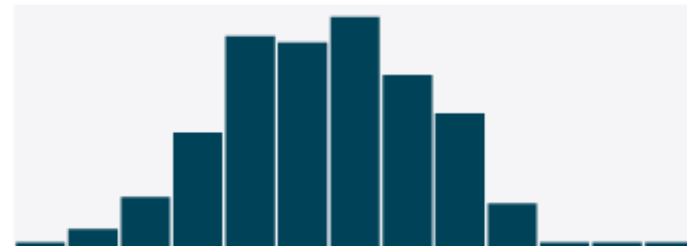
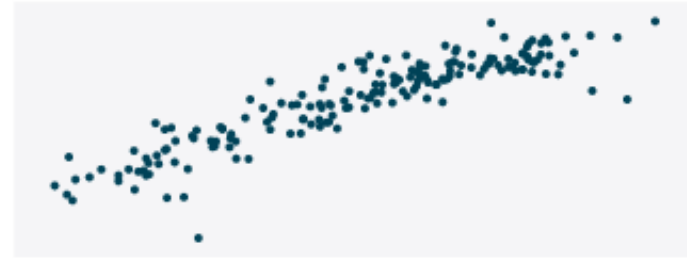
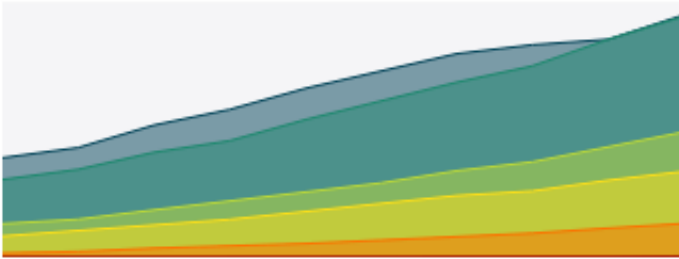
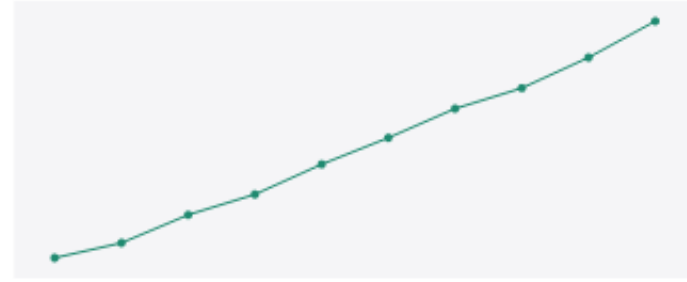
The EDA sequence is:

Problem => Data => Analysis=> Model=> Conclusions

As opposed for a classical approach:

Problem => Data => Model=> Analysis=> Conclusions

# EDA Techniques are generally graphical



# EDA is majorly performed using the following methods:

Univariate visualization – provides summary statistics for each field in the raw data set

Bivariate visualization – is performed to find the relationship between each variable in the dataset and the target variable of interest

Multivariate visualization – is performed to understand interactions between different fields in the dataset

Dimensionality reduction – helps to understand the fields in the data that account for the most variance between observations and allow for the processing of a reduced volume of data.



# Here is a list of all graph types that are illustrated:

- Barplot
- Boxplot
- Density Plot
- Heatmap
- Histogram
- Line Plot
- Pairs Plot
- Polygon Plot
- QQplot
- Scatterplot
- Venn Diagram

Barplot	A barplot (or barchart; bargraph) illustrates the association between a numeric and a categorical variable. The barplot represents each category as a bar and reflects the corresponding numeric value with the bar's size.	barplot(x)
Boxplot	Displays the distribution of a numerical variable based on five summary statistics: minimum non-outlier; first quartile; median; third quartile; and maximum non-outlier. Furthermore, boxplots show the positioning of outliers and whether the data is skewed.	boxplot(x)
Density Plot	A density plot (or kernel density plot; density trace graph) shows the distribution of a numerical variable over a continuous interval. Peaks of a density plot visualize where the values of numerical variables are concentrated.	plot(density(x))
Heatmap	A heatmap (or shading matrix) visualizes individual values of a matrix with colors. More common values are typically indicated by brighter reddish colors and less common values are typically indicated by darker colors.	heatmap(cbind(x, y))
Histogram	A histogram groups continuous data into ranges and plots this data as bars. The height of each bar shows the number of observations within each range.	hist(x)

Line Plot	A line plot, visualizes values along a sequence (e.g., over time). Line plots consist of an x-axis and a y-axis. The x-axis usually displays the sequence and the y-axis the values corresponding to each point of the sequence.	<code>plot(1:length(y), y, type = "l")</code>
Pairs Plot	A pairs plot is a plot matrix, consisting of scatterplots for each variable-combination of a data frame.	<code>pairs(data.frame(x, y))</code>
Polygon Plot	A polygon plot displays a plane geometric figure (i.e., a polygon) within the plot	<code>plot(1,1 col = "white", xlab="X", ylab = "Y") polygon(x= c(0.7, 1.3, 1.3, 0.8), y=c(0.6, 1.0, 1.4, 1.3), col = "#353436")</code>
QQplot	Quantile-Quantile plot; Quantile-Quantile diagram) determines whether two data sources come from a common distribution. QQ plots draw the quantiles of the two numerical data sources against each other. If both data sources come from the same distribution, the points fall on a 45-degree angle.	<code>qqplot(x,y)</code>
Scatterplot	A scatterplot (or scatter plot; scatter graph; scatter chart; scattergram; scatter diagram) displays two numerical variables with points, whereby each point represents the value of one variable on the x-axis and the value of the other variable on the y-axis.	<code>plot(x,y)</code>
Venn Diagram	A venn diagram (or primary diagram; set diagram; logic diagram) illustrates all possible logical relations between certain data characteristics. Each characteristic is represented as a circle, whereby overlapping parts of the circles illustrate elements that have both characteristics at the same time.	<code>Install.packages("VennDiagram") library("VennDiagram") plot.new() draw.single.venn(area = 10)</code>

# Graphical Display of Distributions: pie chart

- graphically depicting groups of numerical data through their quartiles

The basic syntax to create a pie chart in R:

```
pie(x, labels, radius, main, col, clockwise)
```

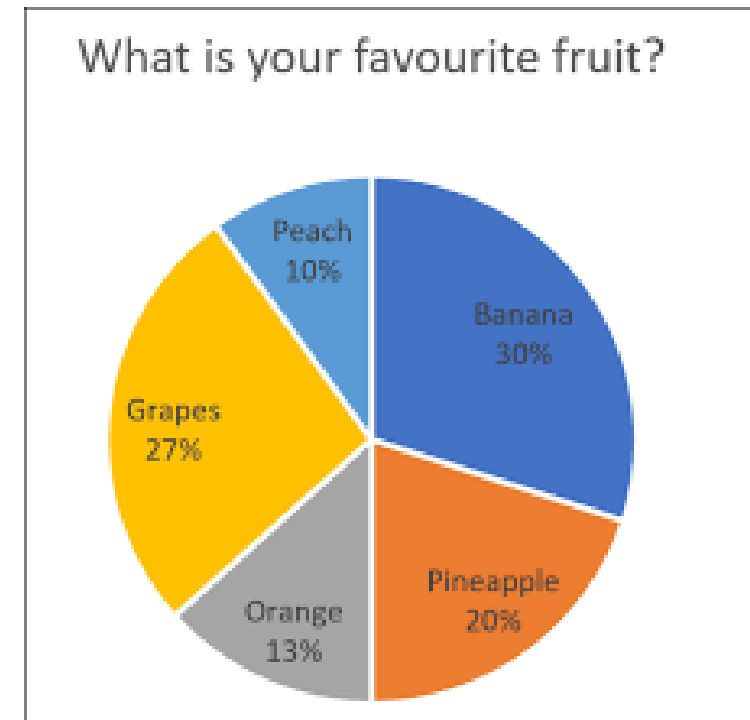
```
# Create data for the graph.
```

```
x <- c(21, 62, 10, 53)
```

```
labels <- c("London", "New York", "Singapore", "Mumbai")
```

```
pie(x,labels)
```

```
.
```



# Graphical Display of Distributions: boxplot

- graphically depicting groups of numerical data through their quartiles

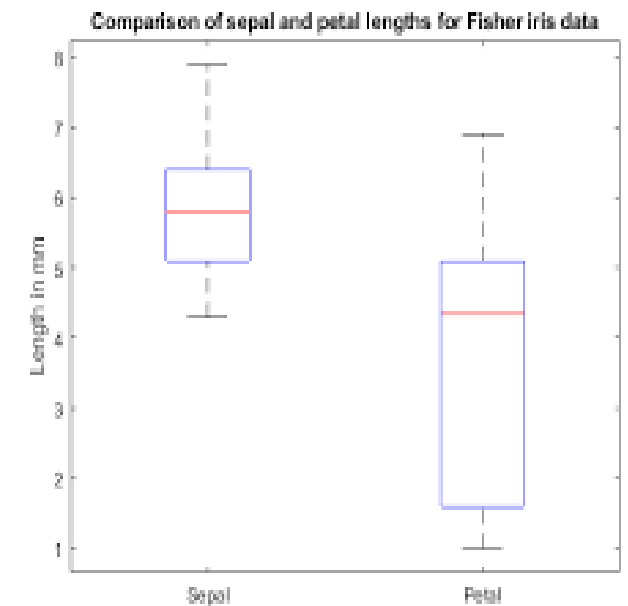
The basic syntax to create a boxplot in R:

```
boxplot(x, data, notch, varwidth, names, main)
```

```
#Use the default “cars” dataset
```

```
#Plot the chart.
```

```
boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders",  
        ylab = "Miles Per Gallon", main = "Mileage Data")
```



# Graphical Display of Distributions: histogram

- Histograms show the number of observations that fall within specified divisions (i.e., bins).

The basic syntax for creating a histogram using R:

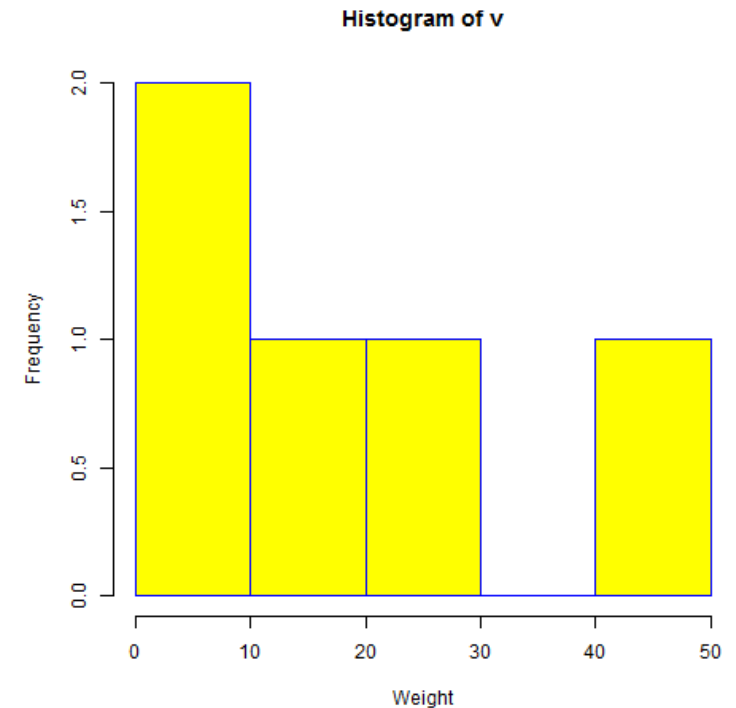
```
hist(v,main,xlab,xlim,ylim,breaks,col,border)
```

```
# Create data for the graph.
```

```
v <- c(9,13,21,8,36,22,12,41,31,33,19)
```

```
# Create the histogram.
```

```
hist(v,xlab = "Weight",col = "yellow",border = "blue")
```



# Graphical Display of Distributions: scatterplot

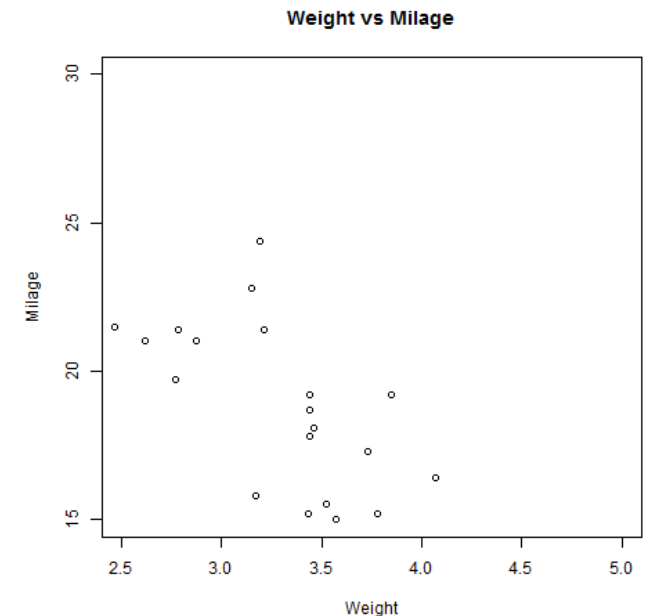
- Scatterplots show many points plotted in the Cartesian plane. Each point represents the values of two variables. One variable is chosen in the horizontal axis and another in the vertical axis.

The basic syntax for creating a scatterplot using R:

```
plot(x, y, main, xlab, ylab, xlim, ylim, axes)
```

```
# Plot the chart for cars with weight between 2.5 to 5  
and mileage between 15 and 30.
```

```
plot(x = input$wt, y = input$mpg,  
     xlab = "Weight", ylab = "Milage",  
     xlim = c(2.5,5), ylim = c(15,30),  
     main = "Weight vs Milage")
```



# Graphical Display of Distributions: barplot

#Bars displayed with values on top

```
set.seed(27222)
```

# Create random example

```
data <- data.frame(x = sample(LETTERS[1:5], 100, replace = TRUE))
```

```
head(data)
```

# Print first lines of data

```
install.packages(ggplot2)
```

```
library(ggplot2)
```

```
data_srz <- as.data.frame(table(data$x))
```

# Summarize data

```
data_srz
```

# Print summarized data

```
ggplot(data_srz, aes(x = Var1, y = Freq, fill = Var1)) + geom_bar(stat = "identity") +  
geom_text(aes(label = Freq), vjust = 0)
```

#Plot with values on top



# Graphical Display of Distributions: ecdf

- ECDF is the fraction of data smaller than or equal to  $x$ . The basic syntax is `ecdf(x)`.

```
> set.seed(19191)    # Set seed for reproducibility
> x <- rnorm(50)      # Normal distribution with 50 values
> ecdf(x)             # Compute ecdf values
> plot(ecdf(x))       # Create ecdf plot in R
```

# Graphical Display of Distributions: Q-Q plots

- Quantile-quantile plots is used to check whether the data is normally distributed. A Q-Q plot is a scatterplot created by plotting two sets of quantiles against one another.

In R, there are two functions to create Q-Q plots: `qqnorm` and `qqplot`.

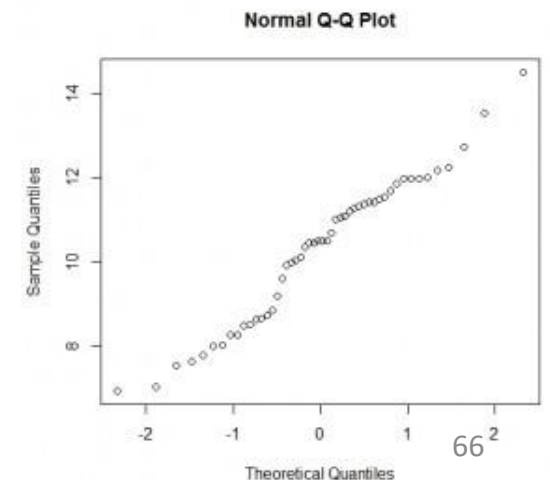
# Creates a standard Normal distribution from 0.01 to 0.99 by increments of 0.01

```
qnorm(seq(0.01,0.99,0.01))
```

```
quantile(rnorm(200),probs = seq(0.01,0.99,0.01))
```

```
y <- qunif(ppoints(length(randu$x)))
```

```
qqplot(randu$x,y)
```



# Frequency tables: Used to describe categorical variables

- You can generate contingency (frequency) tables using:
  - the `table( )` function
  - tables of proportions using the `prop.table( )` function, and
  - marginal frequencies using `margin.table( )`
- Compute table margins and relative frequency
  - `table(x)`
  - `margin.table(x, margin = NULL)`
  - `prop.table(x, margin = NULL)`

# Frequency tables: Used to describe categorical variables

#Example:

```
> m <- matrix(1:4, 2)
```

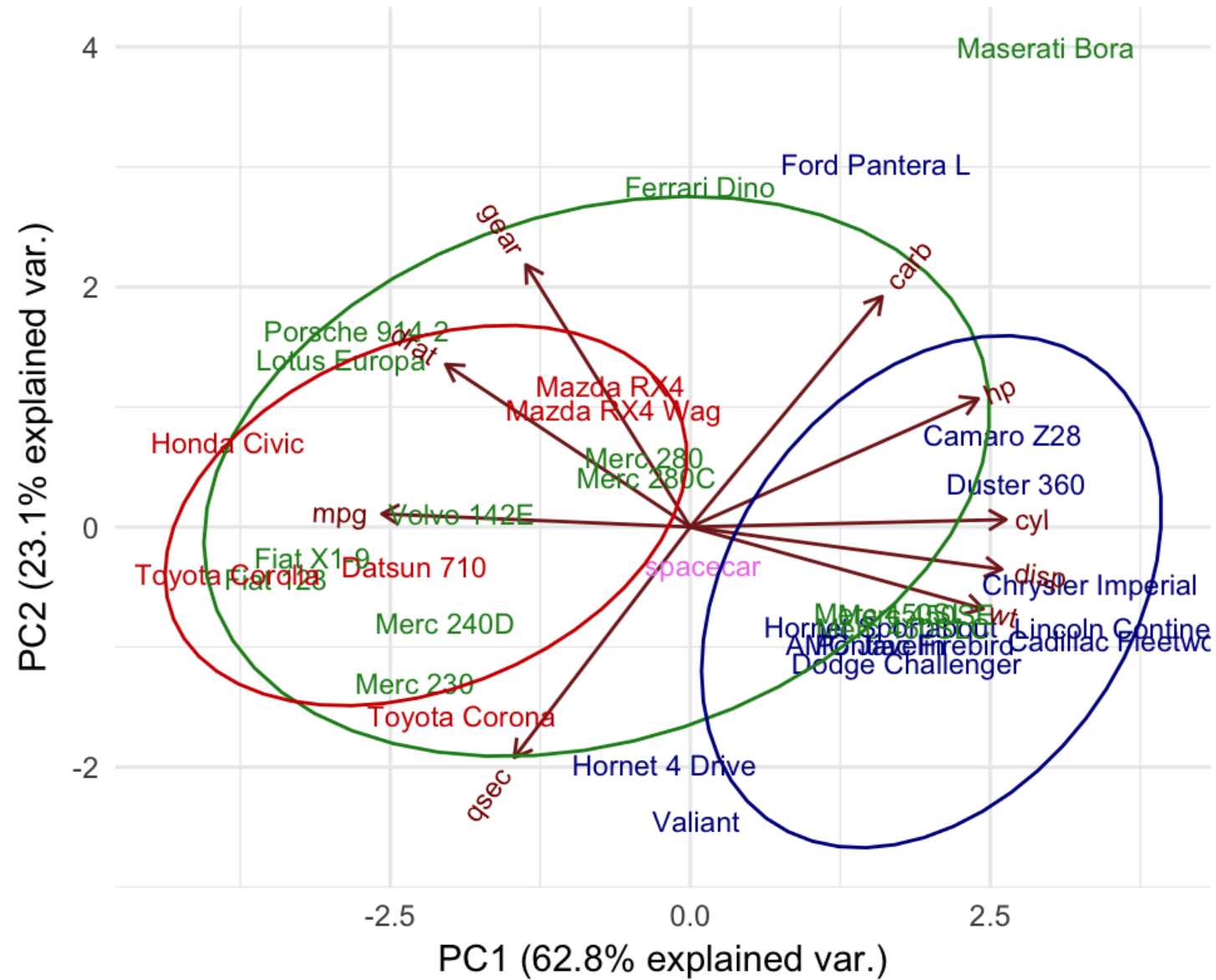
```
> m
```

```
> margin.table(m, 1)
```

```
> margin.table(m, 2)
```

```
> prop.table(m, 1)
```

# PCA of mtcars dataset, with extra sample projected



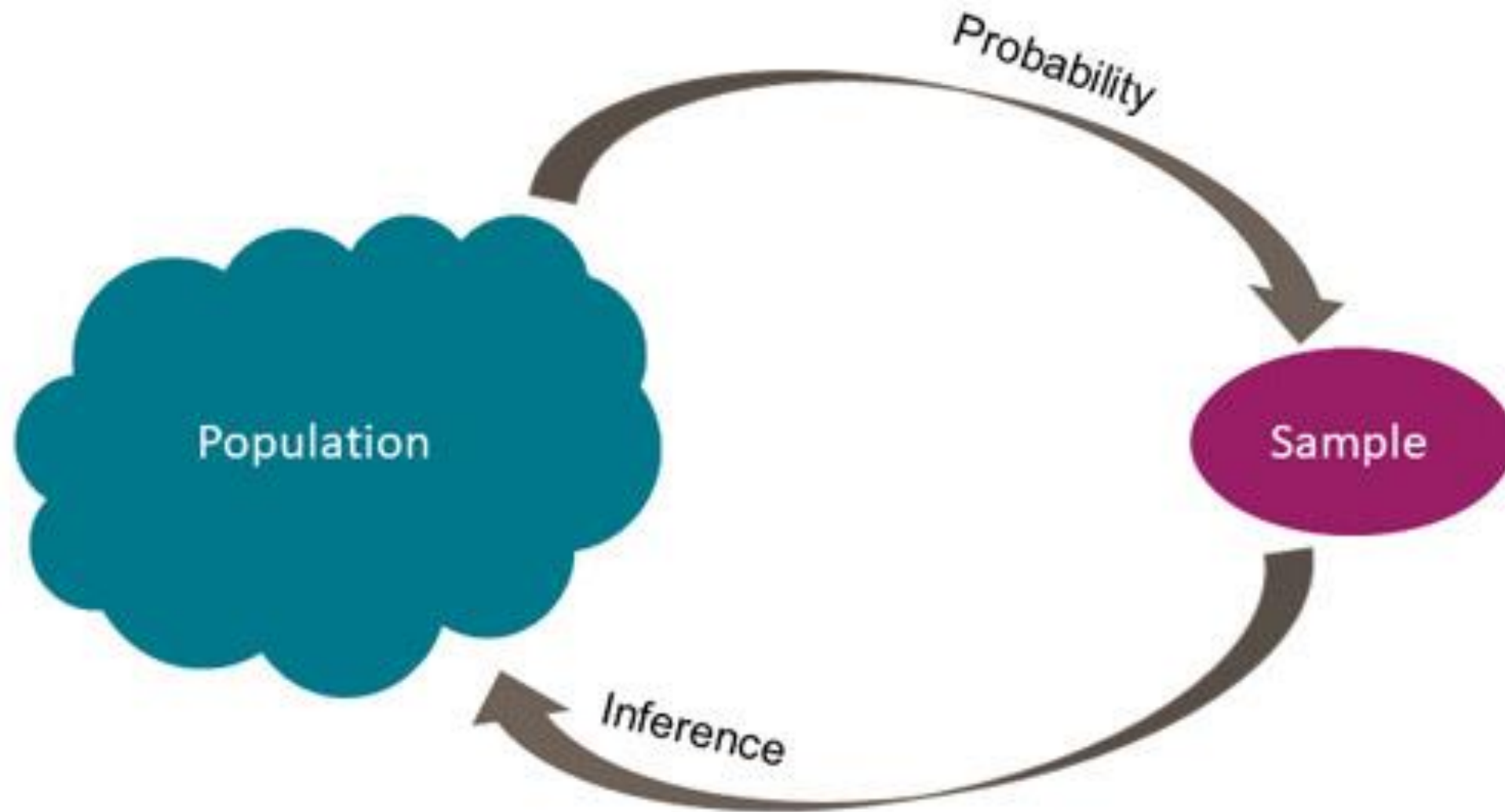
Origin — Europe — Japan — Jupiter — US

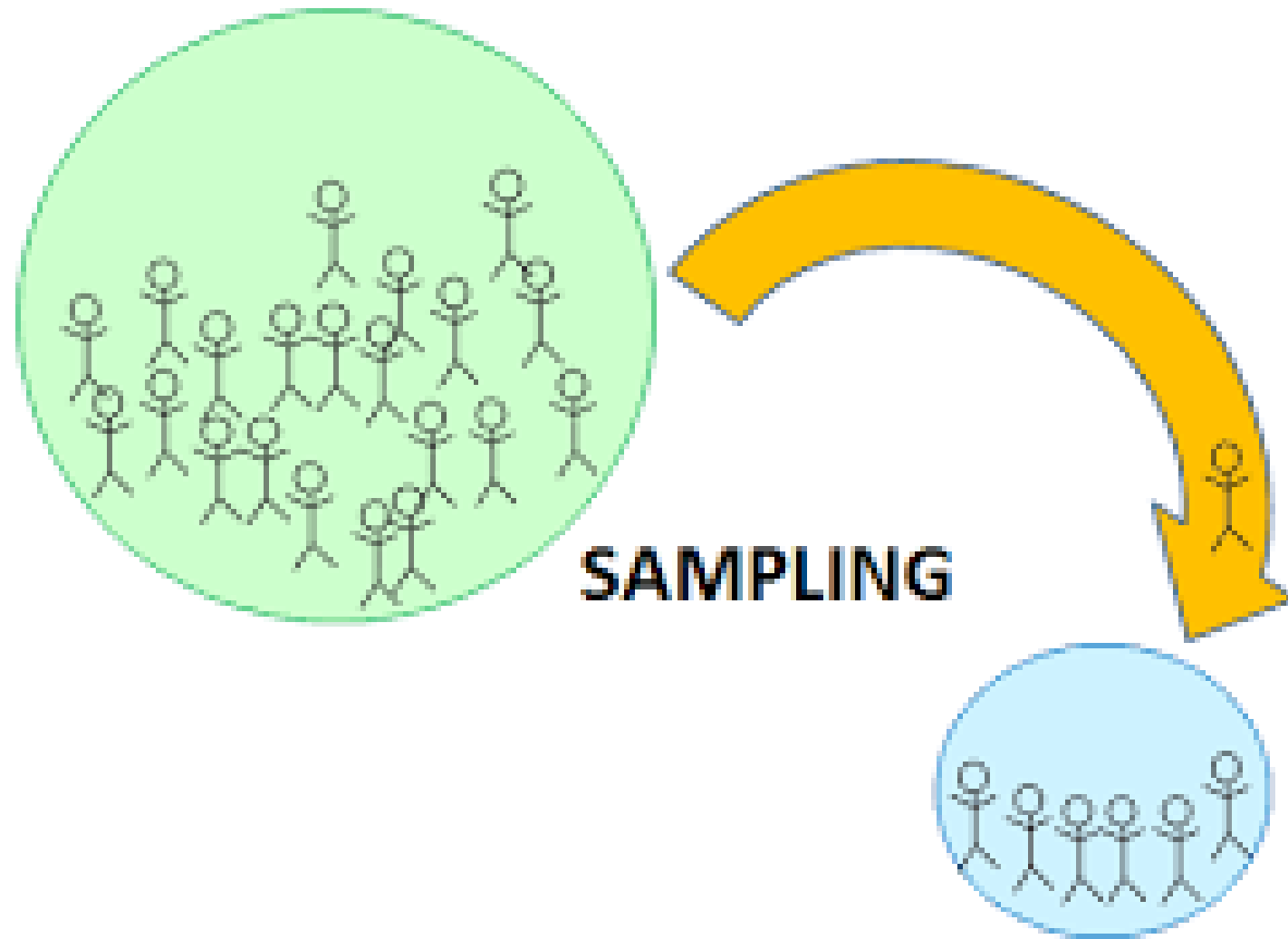
The area of *descriptive statistics* is concerned with meaningful and efficient ways of presenting data.

When it comes to *inferential statistics*, though, our goal is to make some statement about a characteristic of a population based on what we know about a sample drawn from that population.

S. No	Descriptive Statistics	Inferential Statistics
1	Concerned with the describing the target population	Make inferences from the sample and generalize them to the population.
2	Organize, analyze and present the data in a meaningful manner	Compares, test and predicts future outcomes.
3	Final results are shown in form of charts, tables and Graphs	Final result is the probability scores.
4	Describes the data which is already known	Tries to make conclusions about the population that is beyond the data available.
5	Tools- Measures of central tendency (mean/median/ mode), Spread of data (range, standard deviation etc.)	Tools- hypothesis tests, Analysis of variance etc.

# Inferential statistics







# Ex. Dimension of Matrix or Data Frame

```
> set.seed(8212)
```

```
> N <- 500
```

```
# Set Seed for reproducibility
```

```
# Sample size
```

```
> x1 <- round(rnorm(N, 1, 20))
```

```
> x2 <- round(runif(N, 5, 10))
```

```
> x3 <- round(runif(N, 1, 4), 1)
```

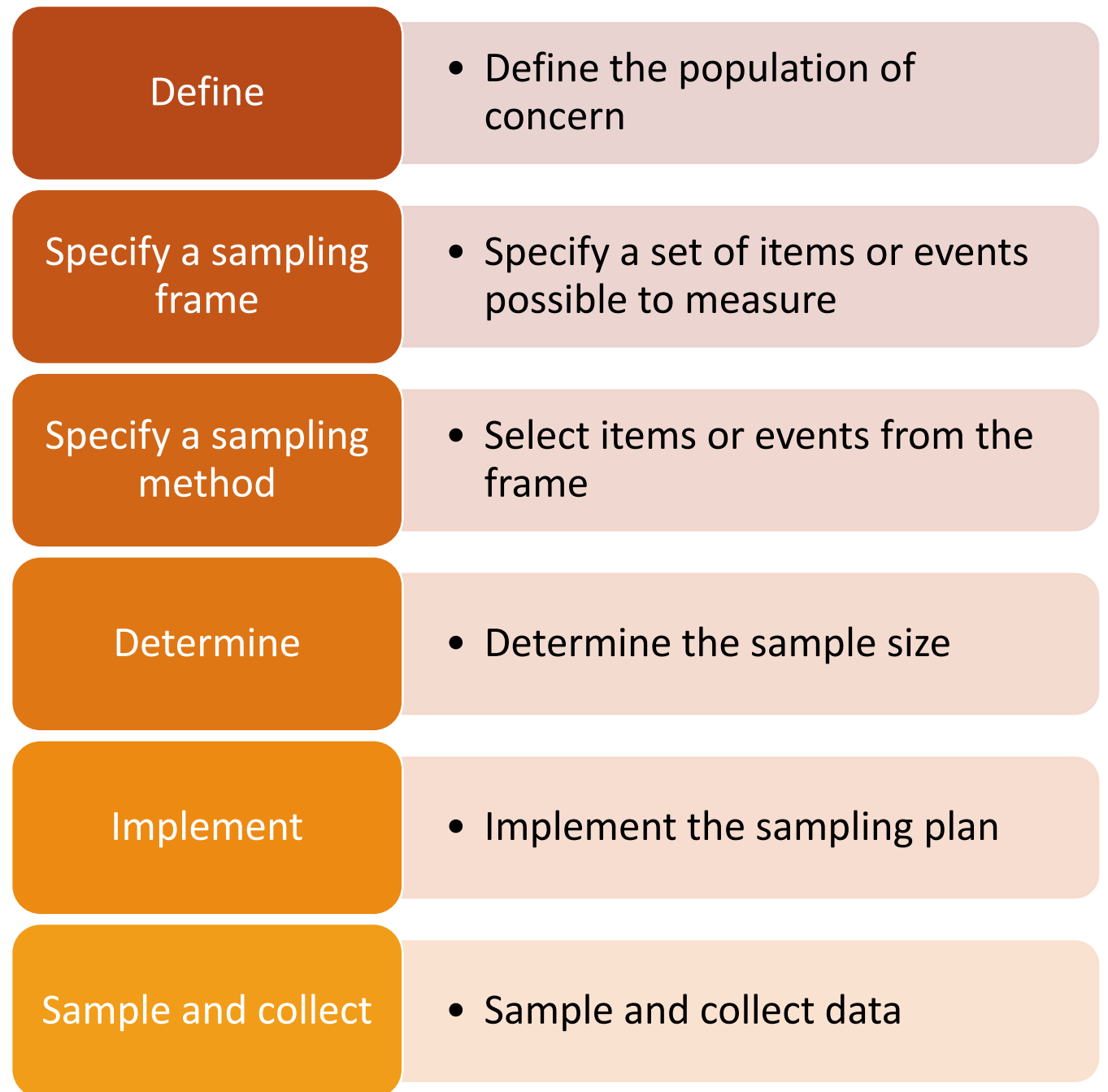
```
> x4 <- round(runif(N, 5, 50))
```

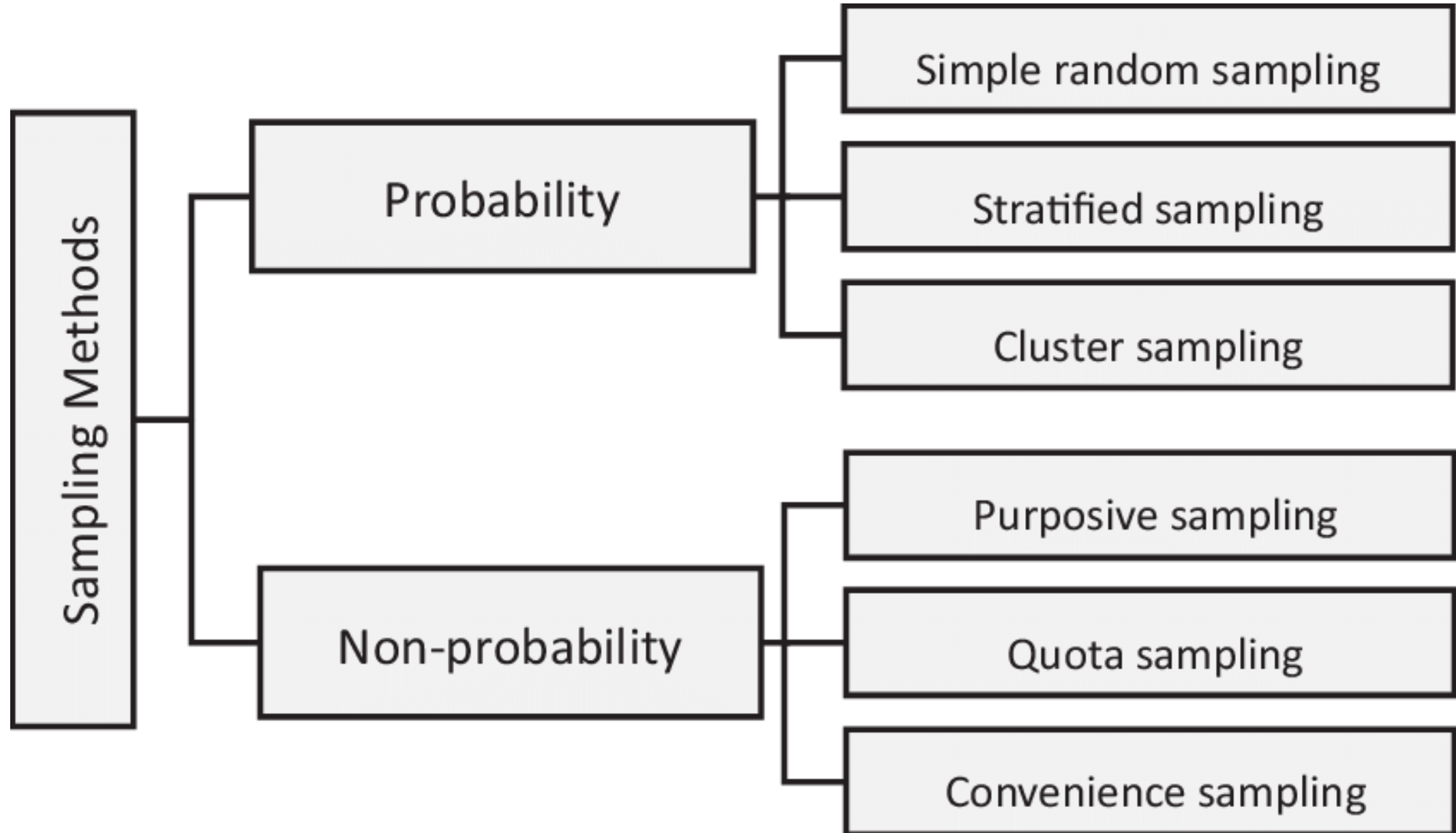
```
> x5 <- rpois(N, 5)
```

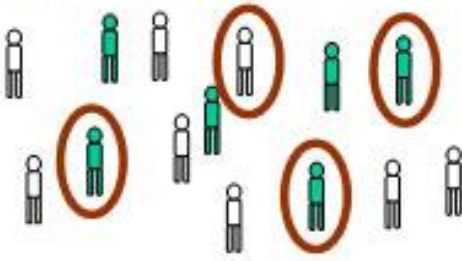
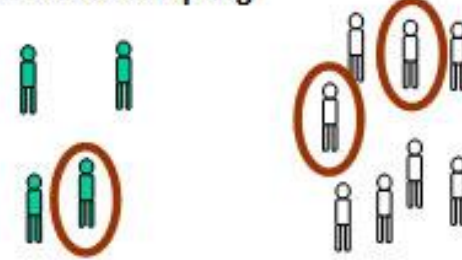
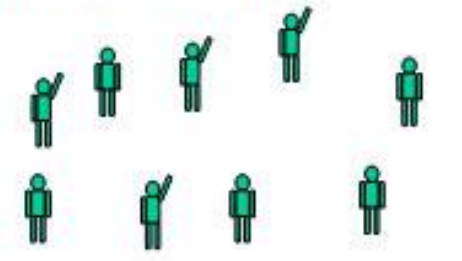
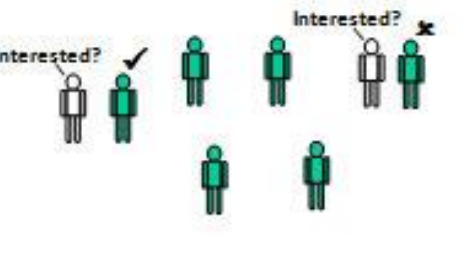
```
# Create 5 random variables
```

	x1	x2	x3	x4	x5
1	29	6	2.5	6	8
2	26	8	1.1	47	6
3	-7	6	3.9	38	5
4	-11	6	1.1	33	7
5	17	7	1.7	27	4
6	-11	7	3.0	41	6

The sampling process comprises several stages:





<p><b>Random sampling</b></p> 	<p>Every member of a population has an equal chance of being selected</p> <p>E.g. Pulling names out of a hat</p>	<p>For very large samples it provides the best chance of an unbiased representative sample</p>	<p>For large populations it is time-consuming to create a list of every individual.</p>
<p><b>Stratified sampling</b></p> 	<p>Dividing the target population into important subcategories</p> <p>Selecting members in proportion that they occur in the population</p> <p>E.g. 2.5% of British are of Indian origin, so 2.5% of your sample should be of Indian origin... and so on</p>	<p>A deliberate effort is made to make the sample representative of the target population</p>	<p>It can be time consuming as the subcategories have to be identified and proportions calculated</p>
<p><b>Volunteer sampling</b></p> 	<p>Individuals who have chosen to be involved in a study. Also called self-selecting</p> <p>E.g. people who responded to an advert for participants</p>	<p>Relatively convenient and ethical if it leads to informed consent</p>	<p>Unrepresentative as it leads to bias on the part of the participant. E.g. a daytime TV advert would not attract full-time workers.</p>
<p><b>Opportunity sampling</b></p> 	<p>Simply selecting those people that are available at the time.</p> <p>E.g. going up to people in cafés and asking them to be interviewed</p>	<p>Quick, convenient and economical. A most common type of sampling in practice</p>	<p>Very unrepresentative samples and often biased by the researcher who will likely choose people who are 'helpful'</p>

# (Simple) Random Sampling

- A sample selected in such a way that every element in the population has a equal probability of being chosen.
- Equivalently, all samples of size  $n$  have an equal chance of being selected.
- Obtained either by sampling with replacement from a finite population or by sampling without replacement from an infinite population.
- Inherent in the concept of randomness: the next result (or occurrence) is not predictable.
- Proper procedure for selecting a random sample: use a random number generator or a table of random numbers
- Assumed when performing conventional statistical analyses
- No guarantee of a representative sample
- May not be feasible (e.g., costly, impractical)

# Stratified Sampling (Proportional Sample)

- Population gets partitioned into groups based on a factor that may influence the variable that is being measured.
- These groups called strata.
- An individual group is called a stratum.
- To perform **stratified sampling**:
  - Partition the population into groups (strata)
  - Obtain a simple random sample from each group (stratum)
  - Collect data on each sampling unit that was randomly sampled from each group (stratum)
  - Works best when a heterogeneous population is split into fairly homogeneous groups.
- Generally, produces more precise estimates of the population percent than estimates that would be found from a simple random sample. More control over representativeness. Allows for intentional oversampling which permits greater statistical precision (i.e., decreases standard errors).
- Must have data on the characteristics of the population in order to select the sample.

# Cluster Sampling

- Stratifying the sampling frame and then selecting some or all of the items from some of, but not all, the strata.
  - Divide the population into groups (clusters).
  - Obtain a simple random sample of so many clusters from all possible clusters.
  - Obtain data on every sampling unit in each of the randomly selected clusters.
- Note that, unlike with the strata in stratified sampling, the clusters should be microcosms, rather than subsections, of the population.
- Each cluster should be heterogeneous.
- Statistical analysis often more complicated than stratified sampling.
- Decreases statistical precision (individuals within groups tend to be more similar so we have less unique information)

# Syntax for Sample Function in R:

```
sample(x, size, replace = FALSE, prob = NULL)
```

# A single roll of a die is a number between one and six

```
> set.seed(1)
```

```
> sample(1:6, 10, replace=TRUE) #Sample with replacement
```

```
> sample(1:6, 10, replace=TRUE)
```

```
> set.seed(123) #Setting a seed
```

```
> index <- sample(1:nrow(iris), 5)
```

```
> index
```

```
> iris[index, ]
```



```
> library(devtools)
```

```
#Stratified sampling
```

```
> set.seed(1)
```

```
> d1 <- data.frame(ID = 1:100, A = sample(c("AA", "BB", "CC", "DD", "EE"), 100,  
      replace = TRUE),  
      B = rnorm(100),  
      C = abs(round(rnorm(100), digits = 1)),  
      D = sample(c("CA", "NY", "TX"), 100, replace = TRUE),  
      E = sample(c("M", "F"), 100, replace = TRUE))
```

```
# What do the data look like in general?
```

```
> summary(d1)
```

```
> library(dplyr)
#obtain stratified sample
> strat_smp<- d1 %>% group_by("A") %>% sample_n(size=10)
> strat_smp

# Let's take a 10% sample from only 'AA' and 'BB' groups from -A- in d1
> strat_smp <- d1 %>% group_by("A", "B") %>% sample_frac(size=.10)

# Let's take 7 samples from all -D- groups in d1, specified by column# number
> strat_smp <- d1 %>% group_by(!!!5)%>% sample_n(size=7)

# Use a two-column strata: -E- and -D- -E- varies more slowly, so it is better to put that first
> strat_smp <- d1 %>% group_by("E", "D") %>% sample_frac(size= 0.20)

# Use a two-column strata (-E- and -D-) but only interested in cases where E- == "F"
> strat_smp <- d1 %>% group_by("E", "D") %>% sample_frac(size= 0.15, weight=(E=="F"))
```

# Non-probability Sampling

Sampling types that should be avoided:

- Convenience (accidental) – selected based on availability
- Quota – selected based on availability with “quotas” being selected to represent the distribution in the population.
- Judgmental – researcher selects units he/she thinks are more representative of the population. Every unit is not eligible for inclusion in the sample, personal biases
- Snowball – unit with a desired characteristic is identified. This unit then identifies other units with desired characteristics and so on.... (i.e., social networks)

These are referred to as "sampling disasters".

- Biased samples
- Based on human choice rather than random selection
- Statistical theory cannot explain how they might behave, and potential sources of bias are rampant.

# Discussion: Variable Selection

How do these limitations impact your selection of data?

- Highly predictive variables — the use of which is prohibited by legal, ethical or regulatory rules.
- Some variables might not be available or might be of poor-quality during modeling or production stages.
- The business will always have the last word and might insist that only business-sound variables are included or request monotonically increasing or decreasing effects.

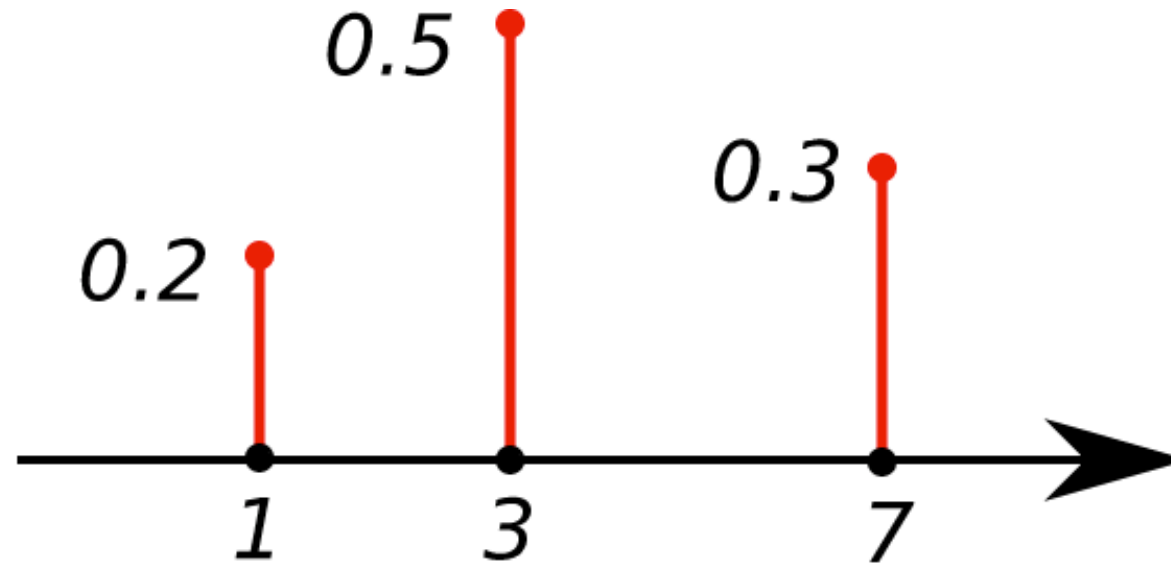
# Sampling Challenges



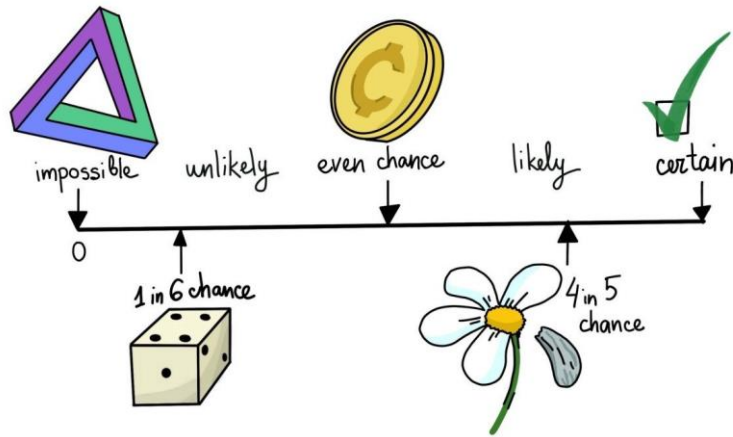
- Sampling error - discrepancies between the sample and the population on a certain parameter that are due to random differences; no fault of the researcher.
  - *Systematic error* - difference between the sample and the population that is due to a systematic difference between the two rather than random chance alone.
  - *Response rate* - sample can become self-selecting, and that there may be something about people who choose to participate in the study that affects one of the variables of interest.
  - *Coverage error* - refers to the fact that sometimes researchers mistakenly restrict their sampling frame to a subset of the population of interest
- \*The more participants a study has, the less likely the study is to suffer from sampling error.

# Distributions of Random Variables

A **probability mass function** (pmf) assigns a probability to each possible value of a **discrete** random variable



# What is a Probability Distribution?



## Discrete Probability Distributions

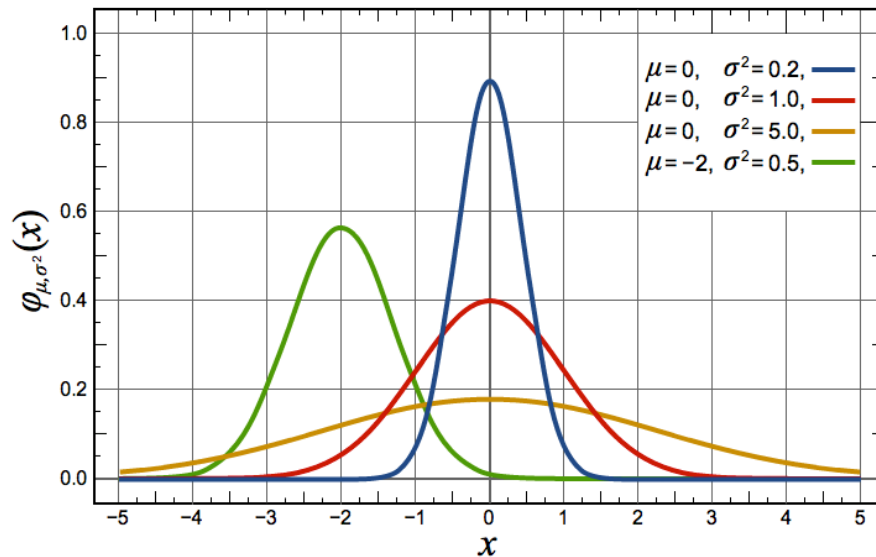
- The probability distribution of a [discrete](#) random variable can always be represented by a table.

## Continuous Probability Distributions

- The probability distribution of a [continuous](#) random variable is represented by an equation, called the **probability density function** (pdf). All probability density functions satisfy the following conditions:
- The random variable  $Y$  is a function of  $X$ ; that is,  $y = f(x)$ .
- The value of  $y$  is greater than or equal to zero for all values of  $x$ .
- The total area under the curve of the function is equal to one.

# Distribution of Random Variables

- A **continuous** random variable  $X$  is described through the **probability density function** (pdf)
- PDF describes the *relative* likelihood for  $X$  to take a given value



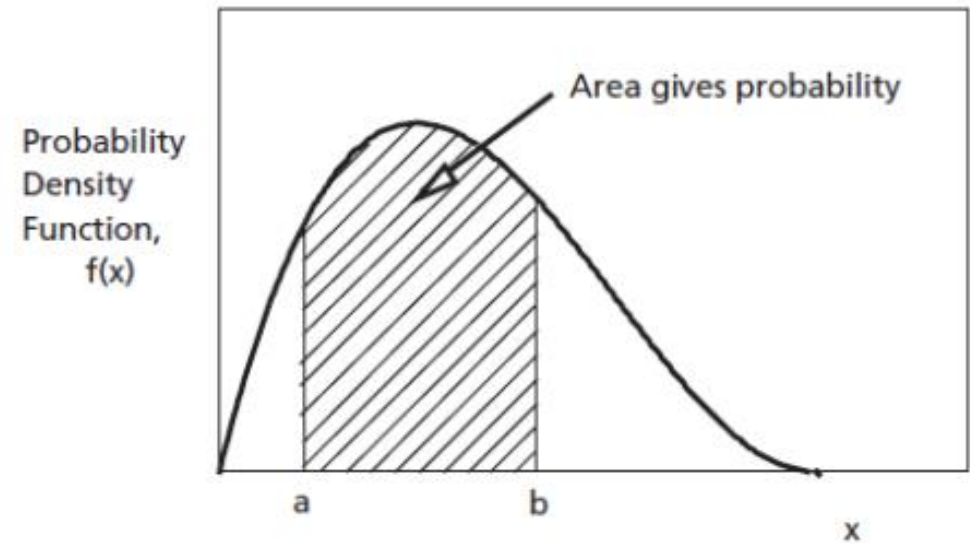
$$P(a \leq X \leq b) = \int_a^b f(x) dx$$

**What is the probability that  $X = b$ ?**



# Normal Distribution

- *dnorm()*: returns the height of the probability distribution at each point. If you only give the points it assumes you want to use a mean of zero and standard deviation of one.
- *pnorm()*: Given a number or a list it computes the probability that a normally distributed random number will be less than that number. Goes by Cumulative Distribution Function
- *qnorm()*: inverse of *pnorm*. The idea behind *qnorm* is that you give it a probability, and it returns the number whose cumulative distribution matches the probability (quantiles)
- *rnorm()*: generates random numbers whose distribution is normal





# The t Distribution

- Values are normalized to mean zero and standard deviation one
  - Specify the number of degrees of freedom
    - *dt*: Distribution function
    - *pt*: Cumulative probability function
    - *qt*: Inverse cumulative probability distribution
    - *rt*: Random
-

# Binomial Distribution

The binomial distribution requires two extra parameters, the number of trials and the probability of success for a single trial.

- *dbinom*:
- *pbinom*: Cumulative probability distribution function
- *qbinom*: Inverse cumulative probability distribution
- *rbinom*: random numbers

# Conditional Probability

Here we can define, 2 events:

- Event A is the probability of the event we're trying to calculate.
- Event B is the condition that we know or the event that has happened.
- We can write the conditional probability as ,  $P\left(\frac{A}{B}\right)$ , the probability of the occurrence of event A given that B has already happened.

$$P\left(\frac{A}{B}\right) = \frac{P(A \text{ and } B)}{P(B)} = \frac{\text{Probability of the occurrence of both A and B}}{\text{Probability of B}}$$

Suppose we have a test for the flu that is positive 90% of the time when tested on a flu patient ( $P(\text{test} + \mid \text{flu}) = 0.9$ ) and is negative 95% of the time when tested on a healthy person ( $P(\text{test} - \mid \text{no flu}) = 0.95$ ). We also know that the flu is affecting about 1% of the population ( $P(\text{flu})=0.01$ ). You go to the doctor and test positive. What is the chance that you truly have the flu?

```
flu <- sample(c('No','Yes'), size=100000, replace=TRUE, prob=c(0.99,0.01))
```

```
test <- rep(NA, 100000) #create a dummy variable first
```

```
test[flu=='No'] <- sample(c('Neg','Pos'), size=sum(flu=='No'), replace=TRUE,  
prob=c(0.95,0.05))
```

```
test[flu=='Yes'] <- sample(c('Neg','Pos'), size=sum(flu=='Yes'), replace=TRUE,  
prob=c(0.1, 0.9))
```

# Bayesian

THE PROBABILITY OF "B"  
BEING TRUE GIVEN THAT  
"A" IS TRUE

↓

THE PROBABILITY  
OF "A" BEING  
TRUE

↙

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

↑

THE PROBABILITY  
OF "A" BEING TRUE  
GIVEN THAT "B" IS  
TRUE

↖

THE PROBABILITY  
OF "B" BEING  
TRUE

# Bayes' Theorem in Machine Learning

- Bayes' theorem tells us how to gradually update our knowledge on something as we get more evidence about that something.
- Bayes' theorem is used in machine learning; both in regression and classification, to incorporate previous knowledge into our models and improve them.

# Confidence Intervals

```
> a <- 5
```

**<- Sample Mean**

```
> s <- 2
```

**<- The Standard Deviation**

```
> n <- 20
```

**<- Sample Size**

```
> error <- qnorm(0.975)*s/sqrt(n)
```

**<- Calculate the error term**

```
> left <- a-error
```

**<- Left (Lower Bound)**

```
> right <- a+error
```

**<- Right (Upper Bound)**

```
> left
```

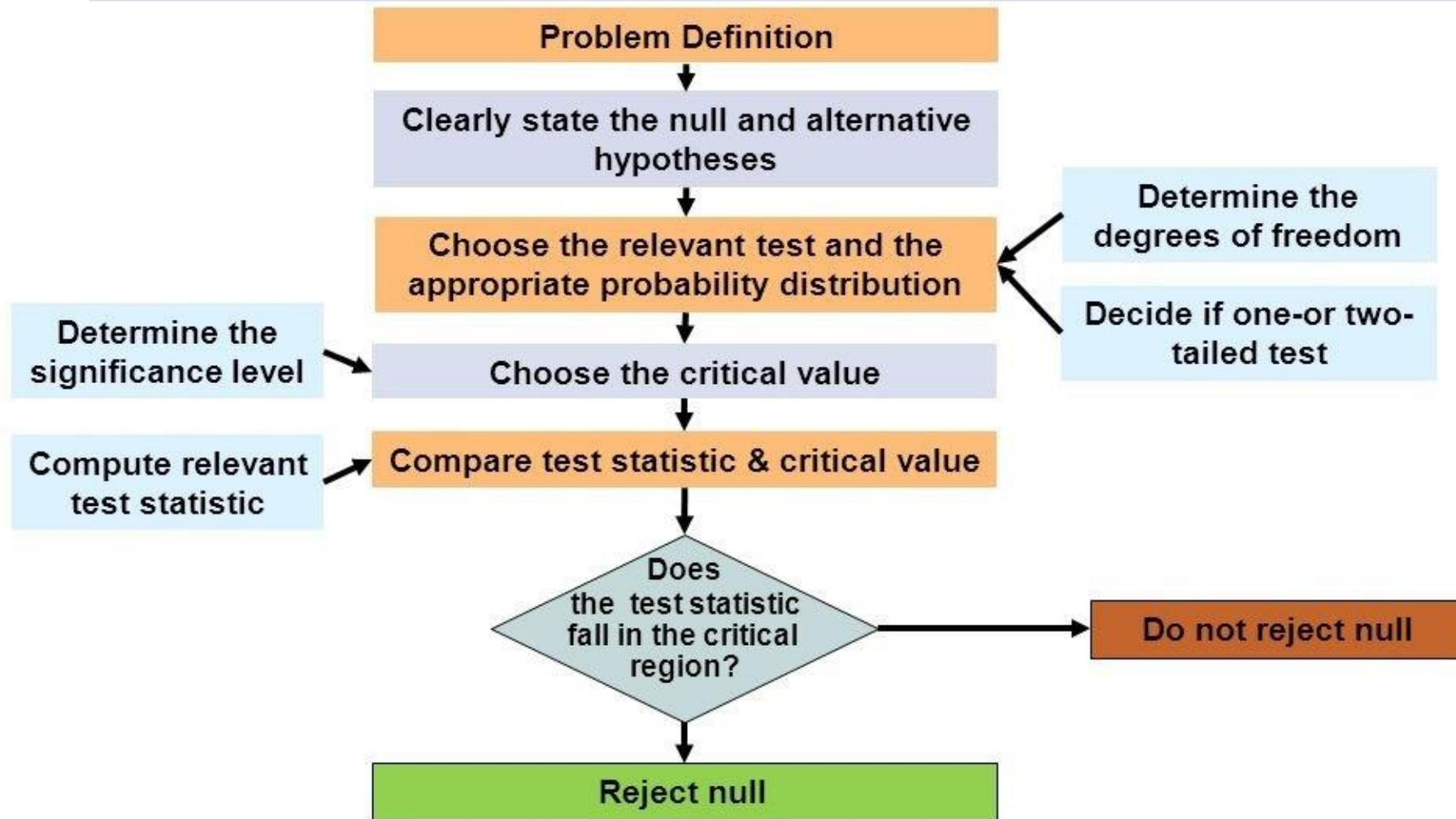
```
[1] 4.123477
```

```
> right
```

```
[1] 5.876523
```



# Hypothesis Testing Process



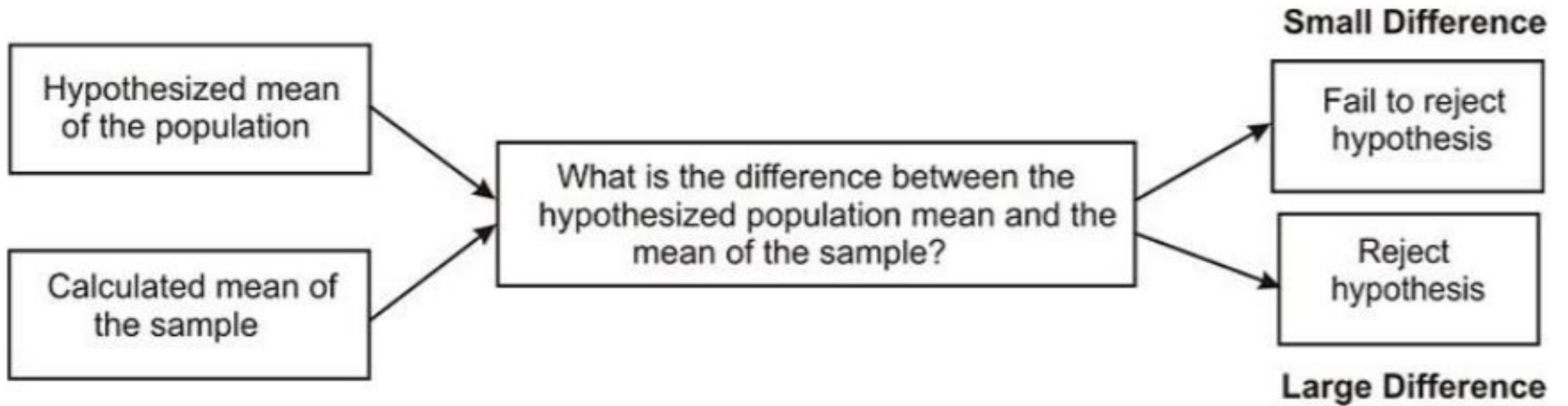
# Example

Consider a person on trial for a “criminal” offense in the United States. Under the US system a jury (or sometimes just the judge) must decide if the person is innocent or guilty while in fact the person may be innocent or guilty.

**Person Is:**

		Innocent	Guilty
		No Error	Error
Jury Says:	Innocent	No Error	Error
	Guilty	Error	No Error

There is a favored assumption, an initial bias. The jury is instructed to assume the person is innocent.



# Hypothesis Testing

- You have sample data, and you are asked to assess the credibility of a statement about population using sample data.
- Suppose we have a website that has a white background and the average engagement on the website by any user is around 20 minutes. Now we are proposing a new website with a yellow background that might increase the average engagement by any user to more than 20 minutes. So, we state the null and alternate hypothesis as:
  1.  $H_0: \mu = 20 \text{ min after the change} \mid H_a: \mu > 20 \text{ min after the change}$
  2. Significance Level :  $\alpha = 0.05$

Recall the **Central Limit Theorem**:

- Using this, we determine if our assumption for the null hypothesis (**H0**) is reasonable or not. If it is unlikely, by the **Rare Event rule**, our hypothesis is probably incorrect (i.e., reject **H0**).

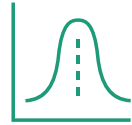
In general, we use the following:

- If the test sample yields an unlikely result, it is probably incorrect (reject **H0** (**large difference**))
- If the test sample yields a likely result, it is probably correct (fail to reject **H0** (**small difference**))

# P – values and Significance tests



There is an extremely close relationship between confidence intervals and hypothesis testing.



When a 95% confidence interval is constructed, all values in the interval are considered plausible values for the parameter being estimated. Values outside the interval are rejected as relatively implausible.



The confidence interval tells you **more than just the possible range around the estimate**. It also tells you about **how stable the estimate is**.



If exact p-value is reported, then the relationship **between confidence intervals and hypothesis testing** is very close.



However, the objective of the two methods is different: **Hypothesis testing** relates to a single conclusion of statistical significance vs. no statistical significance.



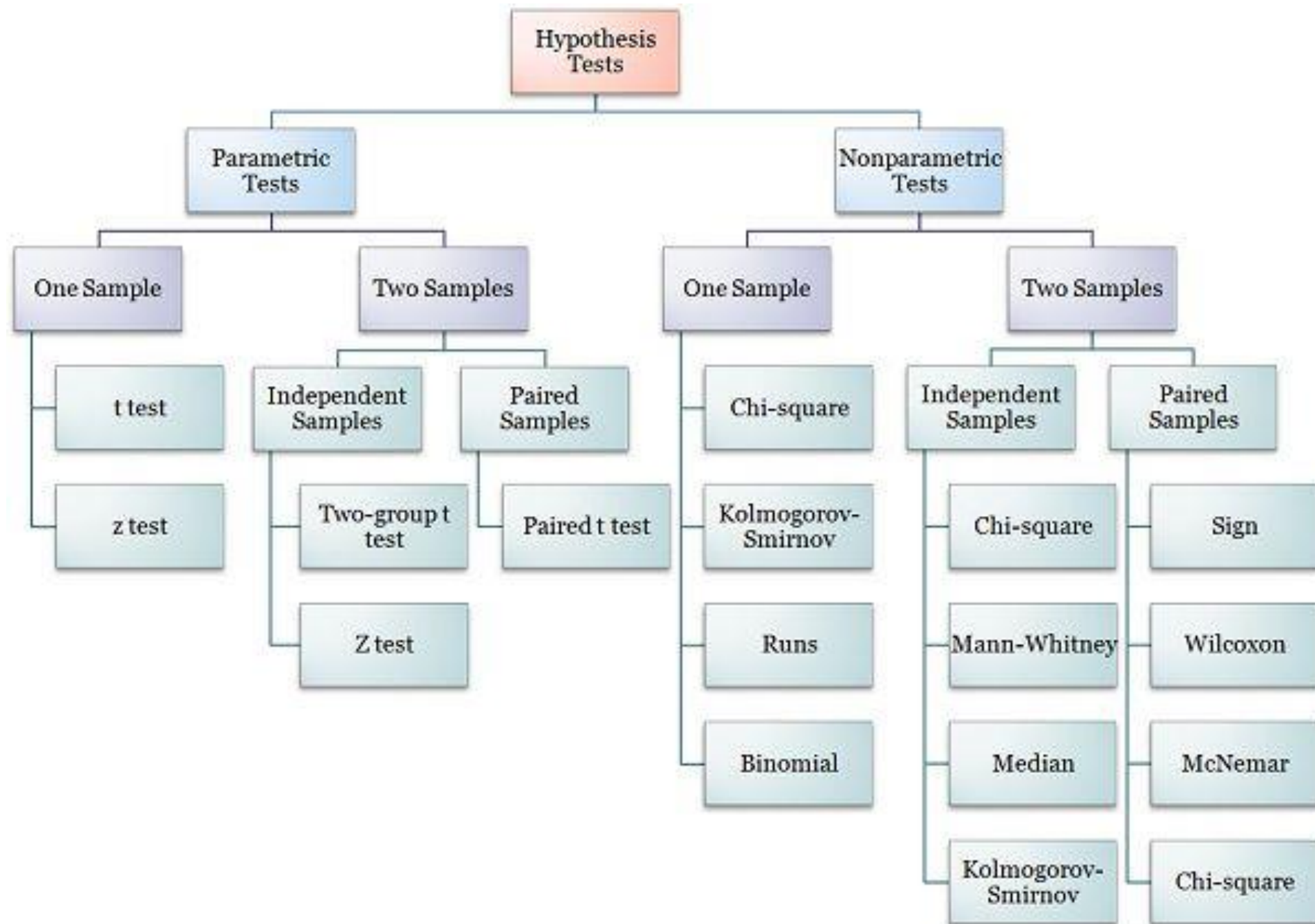
Probability theory: Allows us to calculate the exact *probability* that chance was the real reason for the relationship.



Probability theory allows us to produce test statistics (using mathematical formulas)



A test statistic is a number that is used to decide whether to accept or reject the null hypothesis.





# Applications of Hypothesis Testing

1. When you want to compare the sample mean with the population mean. For example – You would like to determine if the average life of a bulb from brand X is 12 years or not.  
In this case, when you want to check if the sample mean represents the population mean, then you should run **One Sample t-test**.
2. When you want to compare the means of two independent variables. One of which can be a categorical variable. In this case, we run **Two sample t-test**.
3. When you want to compare the before and after-effects of an experiment or a treatment. Then, in that case, we run **Paired t-test**.
4. When you want to compare more than two independent variables; in that case, we run **ANOVA test**.
5. In all the above applications, we assumed that variables are numeric. However, When you want to compare two categorical variables, we run **Chi-square test**.

# $t$ -test statistic

- The  $t$  statistic allows researchers to use sample data to test hypotheses about an unknown population mean.
- The  $t$  statistic is mostly used when a researcher wants to determine whether or not a treatment intervention causes a significant change from a population or untreated mean.
- The goal for a hypothesis test is to evaluate the *significance* of the observed discrepancy between a sample mean and the population mean.
- Therefore, the  $t$  statistic requires that you use the sample data to compute an estimated standard error of  $M$ .
- A large value for  $t$  (a large ratio) indicates that the obtained difference between the data and the hypothesis is greater than would be expected if the treatment has no effect.

# Perform One-Sample t-test

The basic method of applying a t-test to compare two vectors of numeric data:

```
t.test(data.1, data.2)
```

```
> set.seed(100)    # Generating random data with normal distribution
```

```
> x <- round(rnorm(30, mean = 12, sd = 1), 0)
```

```
> t.test(x, mu=12) # Checking if mean is really 12 years:
```

p-value = 0.813, this value is greater than alpha value, and thus we must accept the null hypothesis. Here the null hypothesis was that the average life of the bulb is 12. And the alternative hypothesis was that it is not equal to 12.

95 percent confidence interval:

(11.74772, 12.31895) – The 95% CI also includes the twelve, and thus it is fine to state that the mean value is 12.

## #Create sample data

#Define Sample 1

```
smp2014 <- c(222, 823, 1092, 400, 948, 836)
```

#Define Sample 2

```
smp2019 <- c(910, 650, 700, 892, 229, 1051)
```

#Two sample T-test

```
t.test(smp2014, smp2019, var.equal=FALSE)
```

*#What is the p-value?*

#Run Welch's T-test of Equal Variance

```
t.test(smp2014, smp2019, var.equal=TRUE)
```

# Conditions for a One-Sample z-test

A one sample z test is one of the most basic types of hypothesis test.

1. Normality -normal population. Data roughly fits a [bell curve](#) shape.  
Central Limit Theorem ( $n \geq 30$ )

Graphing - Qplots/box plots/normal probability

2. Independence - population is greater than 10 times the sample size ( $N \geq 10n$ )

Null hypothesis for a 1 sample z-test

- The mean of a population is equal to the sample mean,  $\mu = \mu_0$

Alternative hypotheses for a 1 sample z-test

- The mean of a population is (not equal to/less than/greater than) the sample mean,  
 $\mu \neq \mu_0$  OR  $\mu < \mu_0$  OR  $\mu > \mu_0$

# Conditions for a Two-Sample z-test

Both (two) populations are normally distributed

Null hypothesis for a 2-sample z-test

- The difference between the means of 2 populations is zero (equal means)  
 $\mu_1 - \mu_2 = 0$  OR  $\mu_1 = \mu_2$

Alternative hypotheses for a 2-sample z-test

- Population 1 mean is (not equal to/greater than/less than) the population 2 mean

# ANOVA: Analysis of Variance

Here, Null Hypothesis:  $\mu_1 = \mu_2 = \mu_3$

and, Alternative:  $\mu_1 \neq \mu_2 \neq \mu_3$  or  $\mu_1 = \mu_2 \neq \mu_3$  or  $\mu_1 \neq \mu_2 = \mu_3$

```
> result <- aov(Sepal.Length ~ Species, data = iris) #Running anova
```

```
> summary(result) # Checking the result
```

```
> TukeyHSD(result) # pass the model output
```

```
Df Sum Sq Mean Sq F value    Pr(>F)
Species      2  63.21  31.606  119.3 <0.00000000000000002 ***
Residuals  147   38.96   0.265
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Confusion Matrix

		Prediction Result	
		Attrition	No Attrition
Actual State	Attrition	True Positive	False Negative
	No Attrition	False Positive	True Negative

**Precision = True Positive / (True Positive + False Positive)**

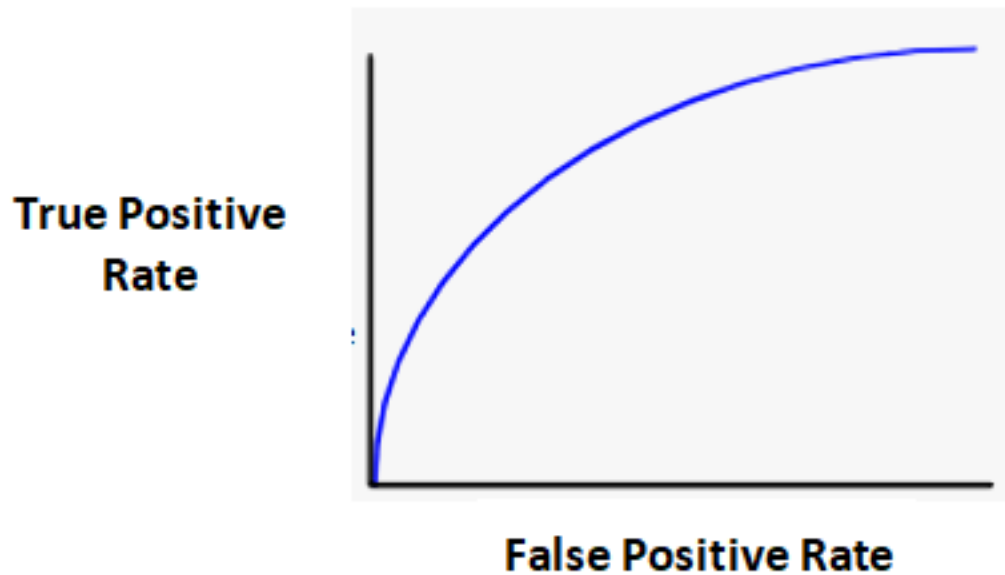
- True Positive : Number of customers who actually attrited whom we correctly predicted as attritors.
- False Positive : Number of customers who actually did not attrite whom we incorrectly predicted as attritors.

**Recall = True Positive / (True Positive + False Negative)**

- True Positive : Number of customers who actually attrited whom we correctly predicted as attritors.
- False Negative: Number of customers who actually attrited whom we incorrectly predicted as non-attritors.



# Precision Recall Curve: popular model performance metrics to evaluate binary classification model



Advantage of using AUPRC over ROC

Cutoff	0.9	0.75	0.6	0.5	0.4
Recall (X)	0.14	0.41	0.69	0.88	0.90
Precision (Y)	0.90	0.85	0.82	0.81	0.50

```
#Area under the precision recall area
recall=c(0.14, 0.41, 0.69, 0.88, 0.90)
precision=c(0.90, 0.85, 0.82, 0.81, 0.50)
i = 2:length(recall)
recall = recall[i] - recall[i-1]
precision = precision[i] + precision[i-1]
(AUPRC = sum(recall * precision)/2)
```

# Were you right ? ...

		The Truth (Based on Entire Population)	
		Nothing Is There ( $H_0$ Is True)	Something Is There ( $H_0$ Is False)
Your Conclusion (Based on Your Sample)	I Don't See Anything (Nonsignificant)	Right!	Wrong (Type II Error)
	I See Something (Significant)	Wrong (Type I Error)	Right!

# Which of the two errors is more serious? Type I or Type II ?

- Since *Type I is the more serious error* (usually), rationale is stick to the status quo or default assumption, at least you're not making things *worse*. But it depends...
- Note: alpha is not a Type I error. Alpha, ' $\alpha$ ', is the *probability of committing* a Type I error. (i.e.  $\alpha = .05$ ; 5% is the level of reasonable doubt that you are willing to accept when statistical tests are used to analyze the data after the study is completed.). Likewise beta, ' $\beta$ ', is the *probability of committing* a Type II error. A Type II error relates to the concept of "power,"; having enough power depends on whether the sample size is sufficiently large to detect a difference when it exists.
- Although type I and type II errors can never be avoided entirely, the you can reduce the likelihood of occurrence by increasing the sample size (the larger the sample, the lesser is the likelihood that it will differ substantially from the population).
- False-positive and false-negative results can also occur because of bias (observer, instrument, recall, etc.). (Errors due to bias, however, are not referred to as type I and type II errors.) Such errors are troublesome, since they may be difficult to detect and cannot usually be quantified.



**Conclusions** are sentence answers which include whether there is enough evidence or not (based on the decision), the level of significance, and whether the original claim is supported or rejected.

**Conclusions** are based on the original claim, which may be the null or alternative hypotheses. The decisions are always based on the null hypothesis

Decision	Original Claim	
	$H_0$ "REJECT"	$H_1$ "SUPPORT"
Reject $H_0$ "SUFFICIENT"	There is <b>sufficient</b> evidence at the alpha level of significance to <b>reject</b> the claim that (insert original claim here)	There is <b>sufficient</b> evidence at the alpha level of significance to <b>support</b> the claim that (insert original claim here)
Fail to reject $H_0$ "INSUFFICIENT"	There is <b>insufficient</b> evidence at the alpha level of significance to <b>reject</b> the claim that (insert original claim here)	There is <b>insufficient</b> evidence at the alpha level of significance to <b>support</b> the claim that (insert original claim here)

# Chi-square

The function used for performing chi-Square test is `chisq.test()`. The syntax is `chisq.test(data)`.

```
> install.library("MASS") #install package
```

```
> library ("MASS")      #load library
```

```
# Create a data frame from the main data set.
```

```
> car.data <- data.frame(Cars93$AirBags, Cars93$Type)
```

```
# Create a table with the needed variables.
```

```
> car.data = table(Cars93$AirBags, Cars93$Type)
```

```
> print(car.data)
```

```
# Perform the Chi-Square test.
```

```
> print(chisq.test(car.data))
```

# Analysis of Variance (ANOVA)

#One Way ANOVA (Completely Randomized Design)

```
> fit <- aov(y ~ A, data=mydataframe)
```

#Randomized Block Design (B is the blocking factor)

```
> fit <- aov(y ~ A + B, data=mydataframe)
```

#Two Way Factorial Design

```
> fit <- aov(y ~ A + B + A:B, data=mydataframe)
```

```
> fit <- aov(y ~ A*B, data=mydataframe) # same thing
```

#Analysis of Covariance

```
> fit <- aov(y ~ A + x, data=mydataframe)
```

#Diagnostic **plots** provide checks for heteroscedasticity, normality, and influential observations.

	Outcome variable						
Input Variable		Nominal	Categorical (>2 Categories)	Ordinal	Quantitative Discrete	Quantitative Non-Normal	Quantitative Normal
	Nominal	$\chi^2$ or Fisher's	$\chi^2$	$\chi^2$ -trend or Mann-Whitney	Mann-Whitney	Mann-Whitney or log-rank <sup>a</sup>	Student's <i>t</i> test
	Categorical (2>categories)	$\chi^2$	$\chi^2$	Kruskal-Wallis <sup>b</sup>	Kruskal-Wallis <sup>b</sup>	Kruskal-Wallis <sup>b</sup>	Analysis of variance <sup>c</sup>
	Ordinal (Ordered categories)	$\chi^2$ -trend or Mann-Whitney	e	Spearman rank	Spearman rank	Spearman rank	Spearman rank or linear regression <sup>d</sup>
	Quantitative Discrete	Logistic regression	e	e	Spearman rank	Spearman rank	Spearman rank or linear regression <sup>d</sup>
	Quantitative non-Normal	Logistic regression	e	e	e	Plot data and Pearson or Spearman rank	Plot data and Pearson or Spearman rank and linear regression
	Quantitative Normal	Logistic regression	e	e	e	Linear regression <sup>d</sup>	Pearson and linear regression

# Correlations

A simplified format is `cor(x, use=, method= )`

Pearson correlation – Pearson correlation is used when we want to assess the degree of association between two quantitative variables.

- `cor(x, method = "pearson")`

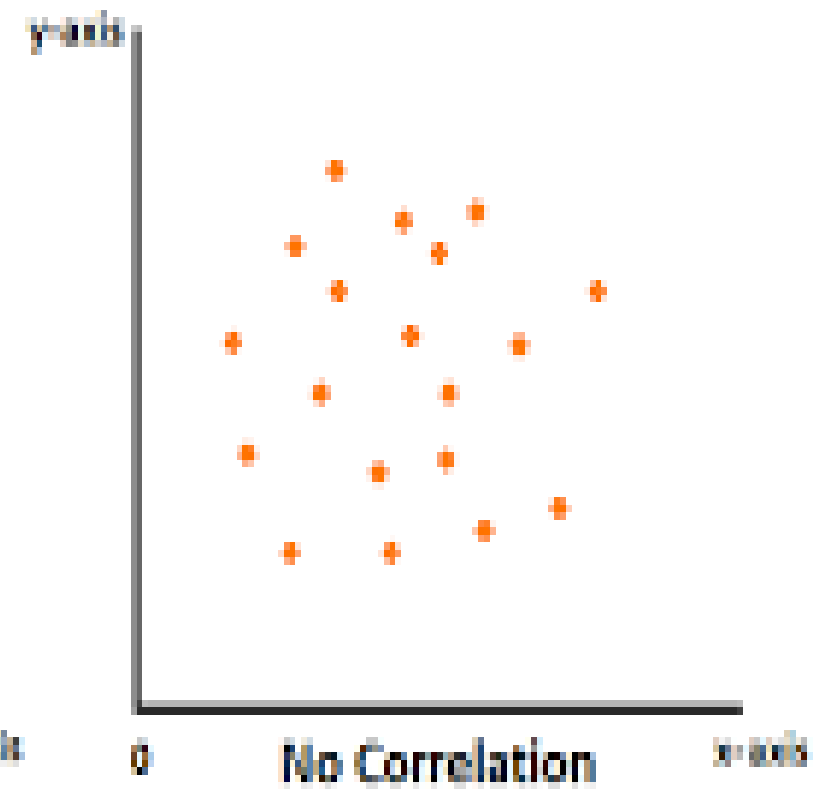
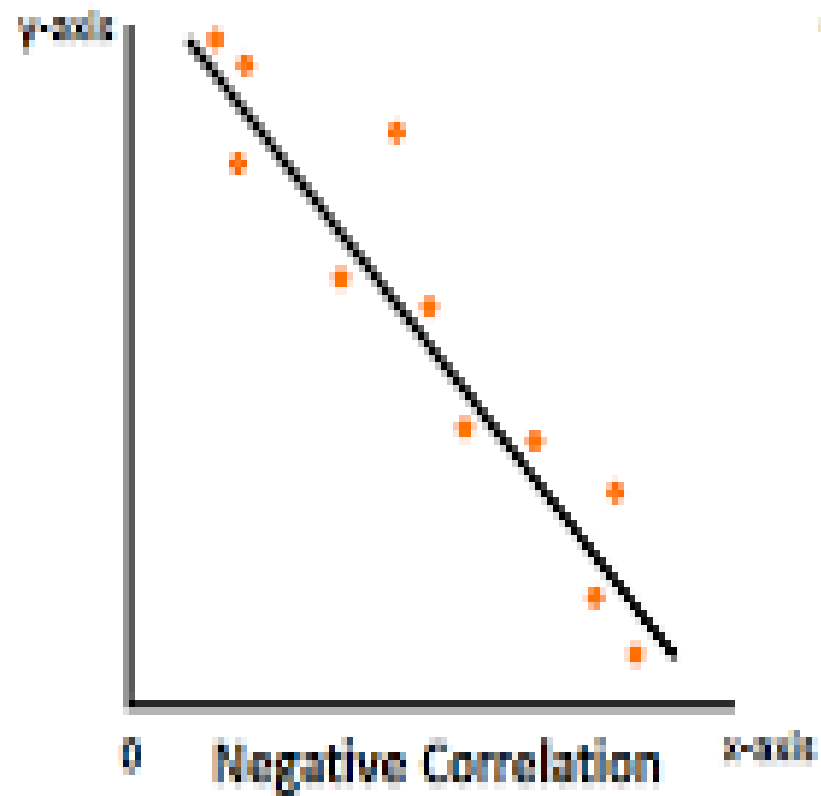
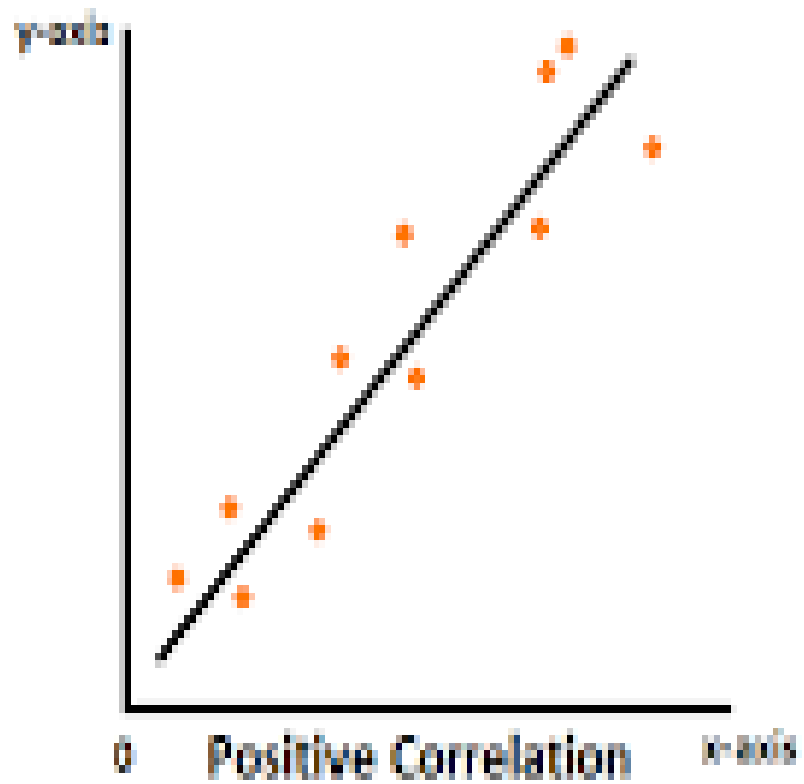
Spearman correlation – Use spearman correlation when you want to assess the degree of association between rank-ordered variables.

- `cor(x, method = "spearman")`

Kendall's correlation – Kendall's correlation can also be used to assess the degree of association between rank-ordered variables. However, it is a non-parametric measure.

- `cor(x, method = "kendall")`



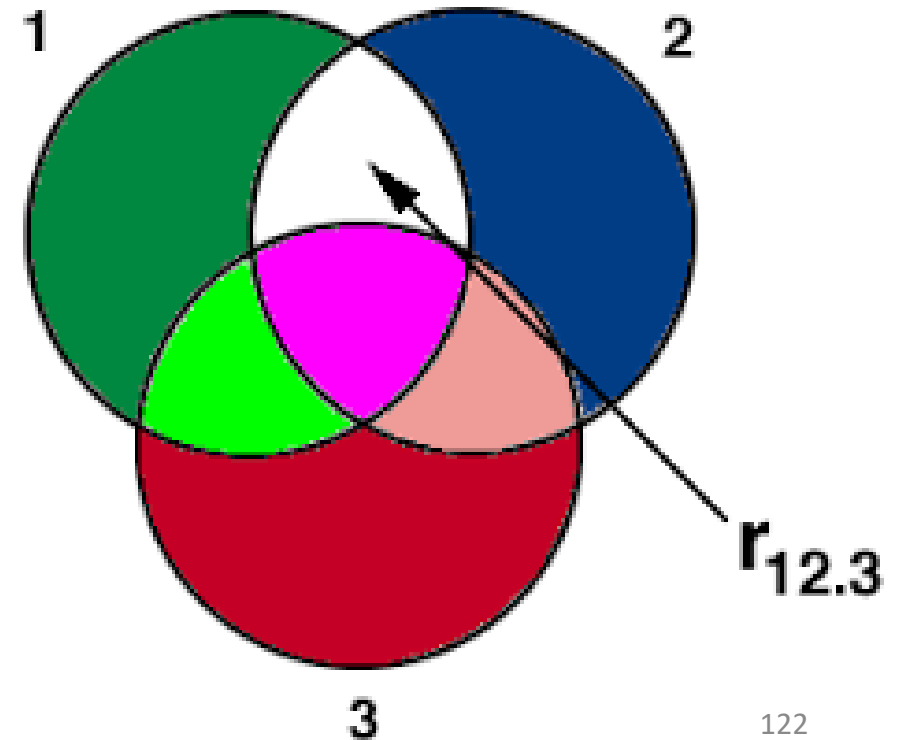


# Partial Correlation

- Partial Correlation measures relationship between two variables (X,Y) while eliminating influence of a third variable (Z).
- Called “correlation” but it is actually regression. It requires estimation of variances.

$$r_{12.3} = \frac{r_{12} - r_{13}r_{23}}{\sqrt{1 - r_{13}^2} \sqrt{1 - r_{23}^2}}$$

```
# Partial correlation  
library(ppcor)  
with(mydata, pcor.test(Height,Weight,Age))
```



# Semi-partial Correlation

- Semi-partial correlation measures the strength of linear relationship between variables X1 and X2 holding X3 constant for just X1 or just X2. It is also called part correlation.

$$r_{1(2.3)} = \frac{r_{12} - r_{13}r_{23}}{\sqrt{1 - r_{23}^2}} \text{ and } r_{2(1.3)} = \frac{r_{12} - r_{13}r_{23}}{\sqrt{1 - r_{13}^2}}$$

```
#Semipartial Correlation Coefficient
```

```
with(mydata, spcor.test(Height,Weight,Age))
```

```
#Semi partial correlation - Age constant for Weight only
```

# Testing for Normality

- Plotting returns in R `plot(x$s) or ggplot`
- Kolmogorov-Smirnov test in R `ks.test(x$s, "pnorm", mean=mean(x$r), sd=sd(x$r))`
- Shapiro-Wilk test in R `shapiro.test(x$columnname)`
- Jarque-Bera test in R `install.packages("tseries")  
library(tseries)  
jarque.bera.test(x$columnname)`

`#import the data`

`View(rtns)`

`#data wrangle and select a column from a dataframe (use select())`

## Data analysis flowchart

