Statistics Bootcamp using R

DAY 1 INTRODUCTION TO STATISTICS IN BUSINESS

1.3 DATA COLLECTION & SUMMARIZATION

GU Zhan (Sam)
Institute of Systems Science
National University of Singapore

issgz@nus.edu.sg

Agenda

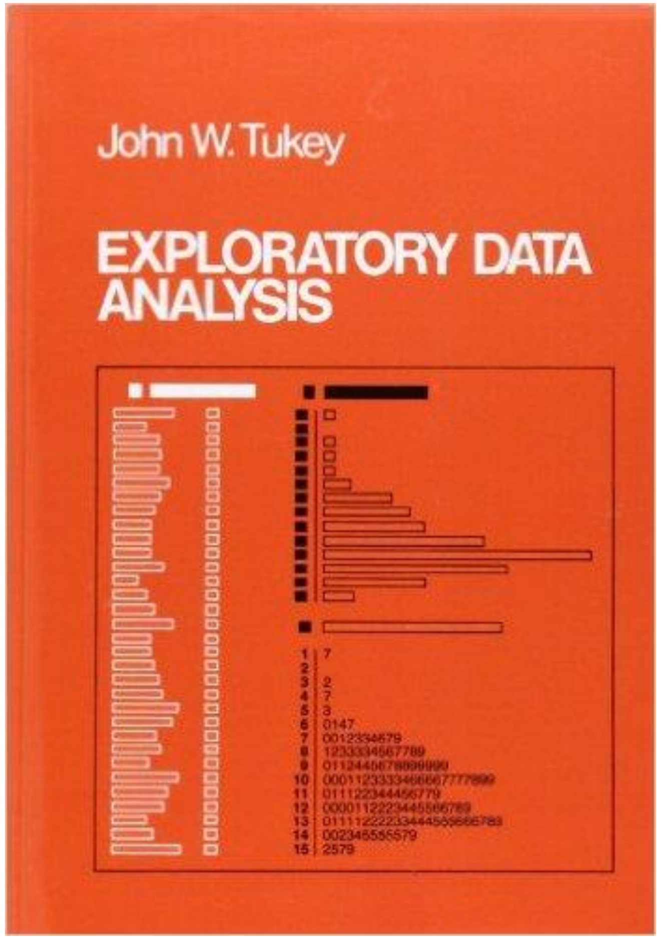
Day 1: Introduction to Statistics in Business

- Basic Vocabulary of Statistics & Data Types
- Introduction to R
- **Data Collection & Summarization**

Learning objectives

- Understand Data sources & Collection & Quality Issues
- Understand Basic Summarization & Visualization Methods
- Understand R data structures, data preparation & visualization using R

What is Exploratory Data Analysis?



- Consist of those preliminary investigate activities
- Undertaken to suggest or establish empirical models for subsequent confirmatory analysis
- Relies on Visual Analysis and examination of evidence.

Key concepts about EDA

Objectives : Discover *patterns*, spot *anomalies*, Frame Hypothesis, Check Assumptions

Exploratory : explore and identify possible underlying structure of a set of variables without imposing preconceived structure on outcome.

Type of activity:

- Central tendency
- Spread
- Distribution
- Trends
- Outliers
- Correlations

Confirmatory : statistical technique used to verify the structure/factor structure of variables. One would test perform hypothesis testing to confirm that observed structure/construct exist.

Type of activity:

- Hypothesis testing

Something about Data

- ***Data*** is a set of measurement made on a group of individuals

- ***Individuals*** are the objects described by a set of data.
  - Example : students, cars,…

- A ***variable*** is any characteristics of an individual that is of interest to the researcher. It takes on different values for different individuals
  - Example : age, gender, GPA,…

- ***Measurement*** is the value of recorded for each variable on an individual.
  - Example : Catherine, 25, Female, 4.0..

Data Quality

- All data is dirty! – it does not perfectly describe the features of the real world.
  - Data might be missing.
  - Data might be duplicated.
  - Data contains typographical or data-entry errors.
  - Deliberate incomplete/incorrect information entered.
  - Categorical variables might have too many values.
  - Numeric variables might have unusual distributions and outliers.
  - Meanings can change over time.
  - Data might be coded inconsistently.

Data is always dirty

Time beyond 24 hours

777	17-Jun-17	2:00PM
778	17-Jun-17	25:00PM
779	17-Jun-17	2:50PM

Inconsistent AM/PM

222	28-Jun-17	7.30am
223	28-Jun-17	5.40PM
224	28-Jun-17	8.20pm

Incorrect values

632	24-May-17	4:30PM
633	24-May-17	5:40pjm
634	24-May-17	6:10PM

Incorrect date

112	13-Feb-17	11.40pm
113	13-Feb-17	12.10am
114	13-Feb-17	1.00am

Inconsistent date format

336	12-Jan-17	6:40PM
337	12-Jan-17	7:30PM
338	13/01/2017	8:00PM

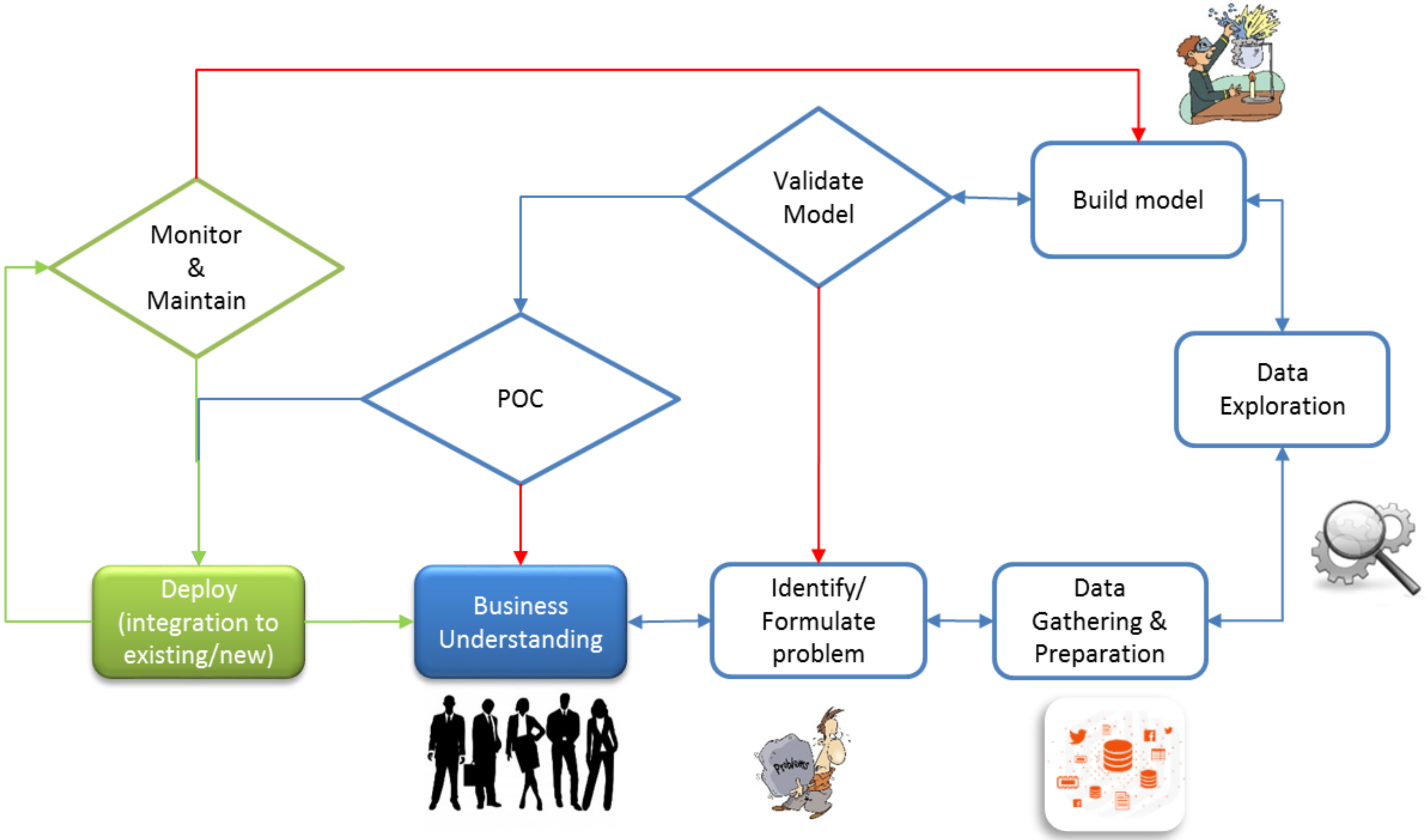
Sources of Data

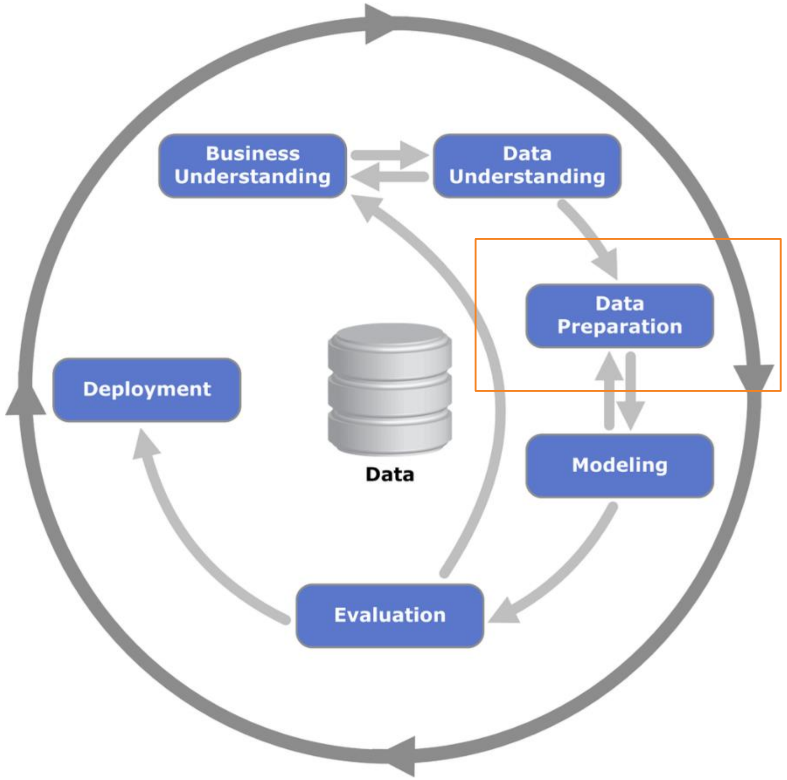
- **Primary Data:** Primary data is **data that is collected by a researcher from first-hand sources**, using methods like surveys, interviews, or experiments. It is collected with the research project in mind, directly from primary sources.
  - Data collected from a customer surveys
  - Data collected by Market Research companies to fulfil specific research requirement

- **Secondary Data:** Secondary data is **data gathered from studies, surveys, or experiments** that have been run by other people or for other research or generated from regular organizational activity
  - Census data
  - Data from a past transactions, operations
  - Data from printed sources – the competition, internet, market analysts

Data visualization for data exploration

Where does visualization exist in Analytics Project Life Cycle?





https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining

Describe the monthly salary level of this class

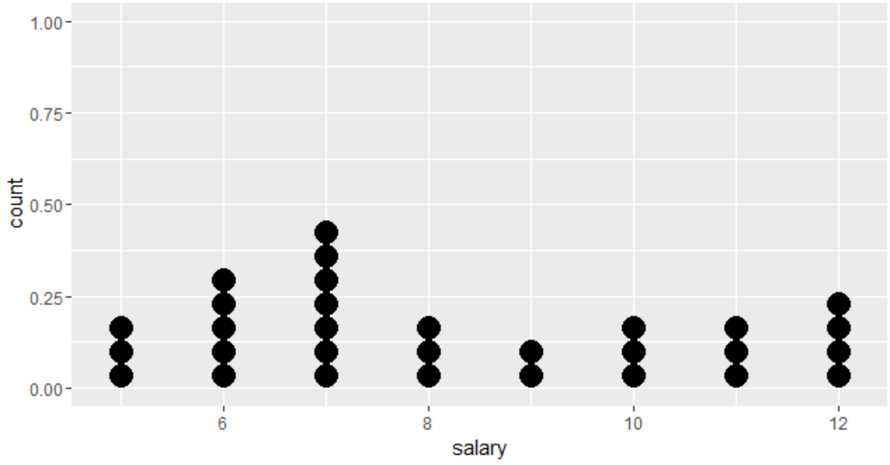
ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
$'000	6	5	8	7	11	12	6	6	7	7	10	6	7	11	10	9	5	6	12	8	12	7	7	12	10	5	7	9	11	8

Average :

ID	2	17	26	1	7	8	12	18	4	9	10	13	22	23	27	3	20	30	16	28	11	15	25	5	14	29	6	19	21	24
$'000	5	5	5	6	6	6	6	6	7	7	7	7	7	7	7	8	8	8	9	9	10	10	10	11	11	11	12	12	12	12

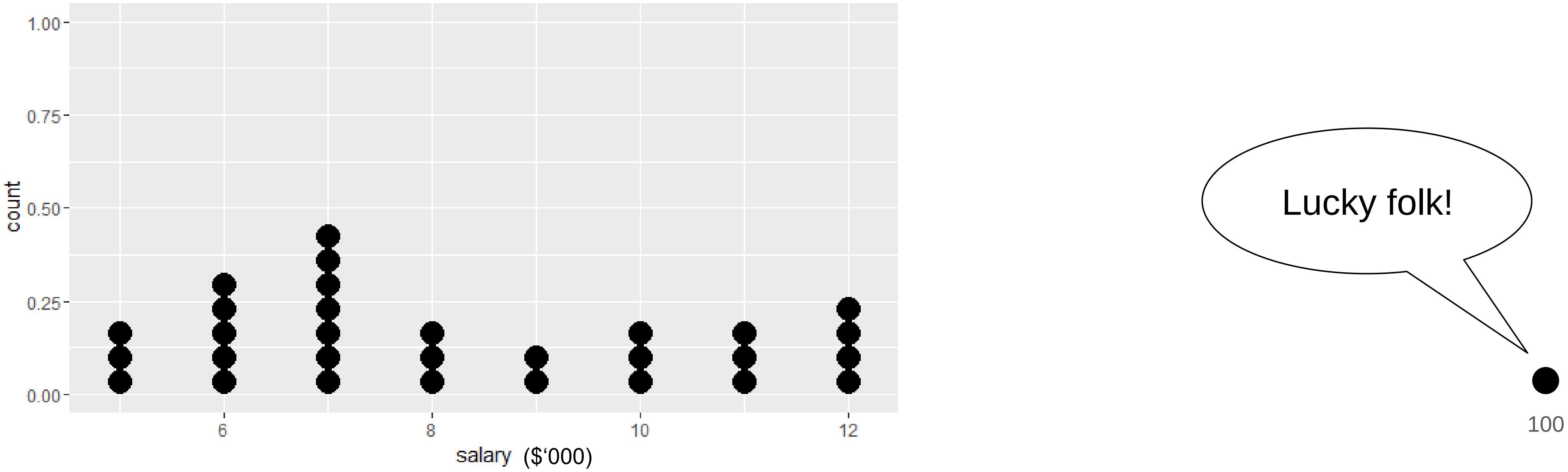
Mode :

Median :



Question

How does one lucky class participant with his/her salary suddenly increased to $100K impact the statistics results earlier?

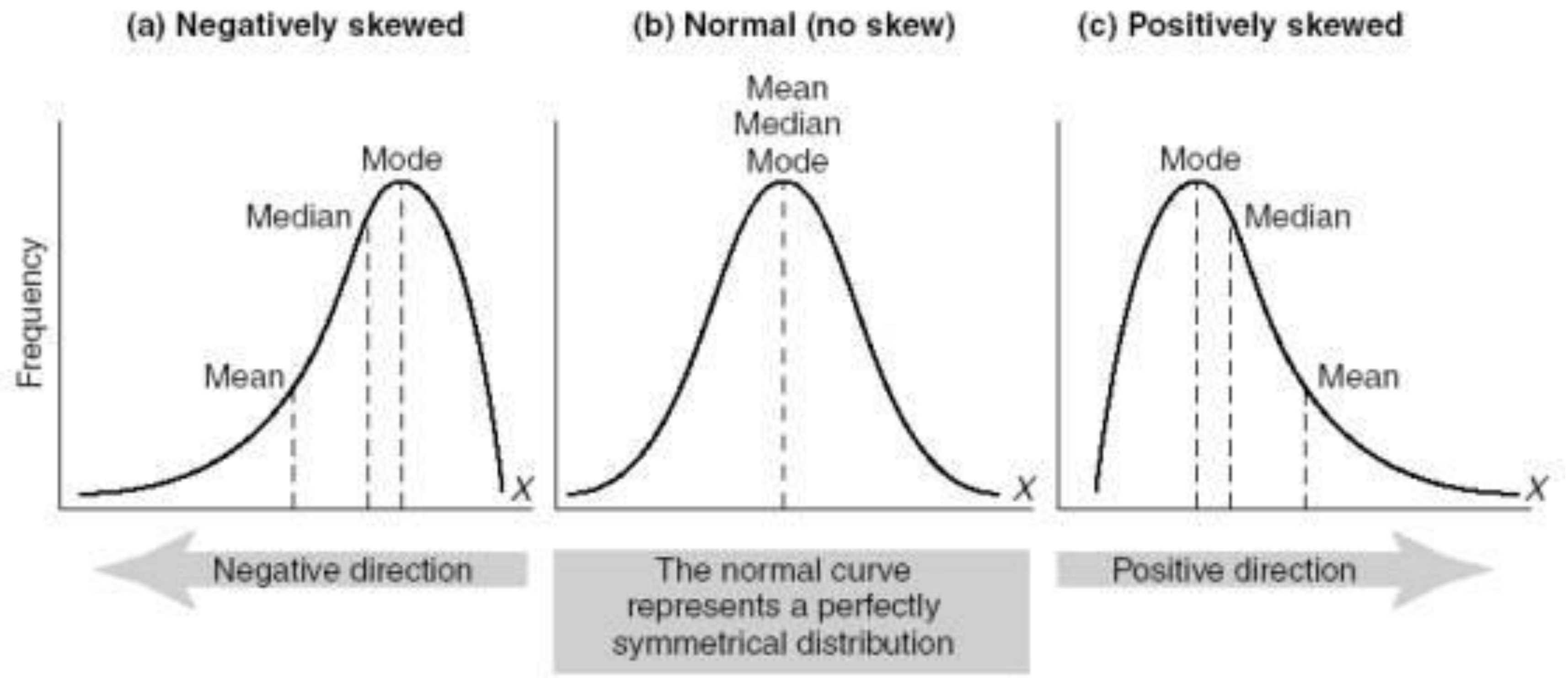


Measures of Central Tendency

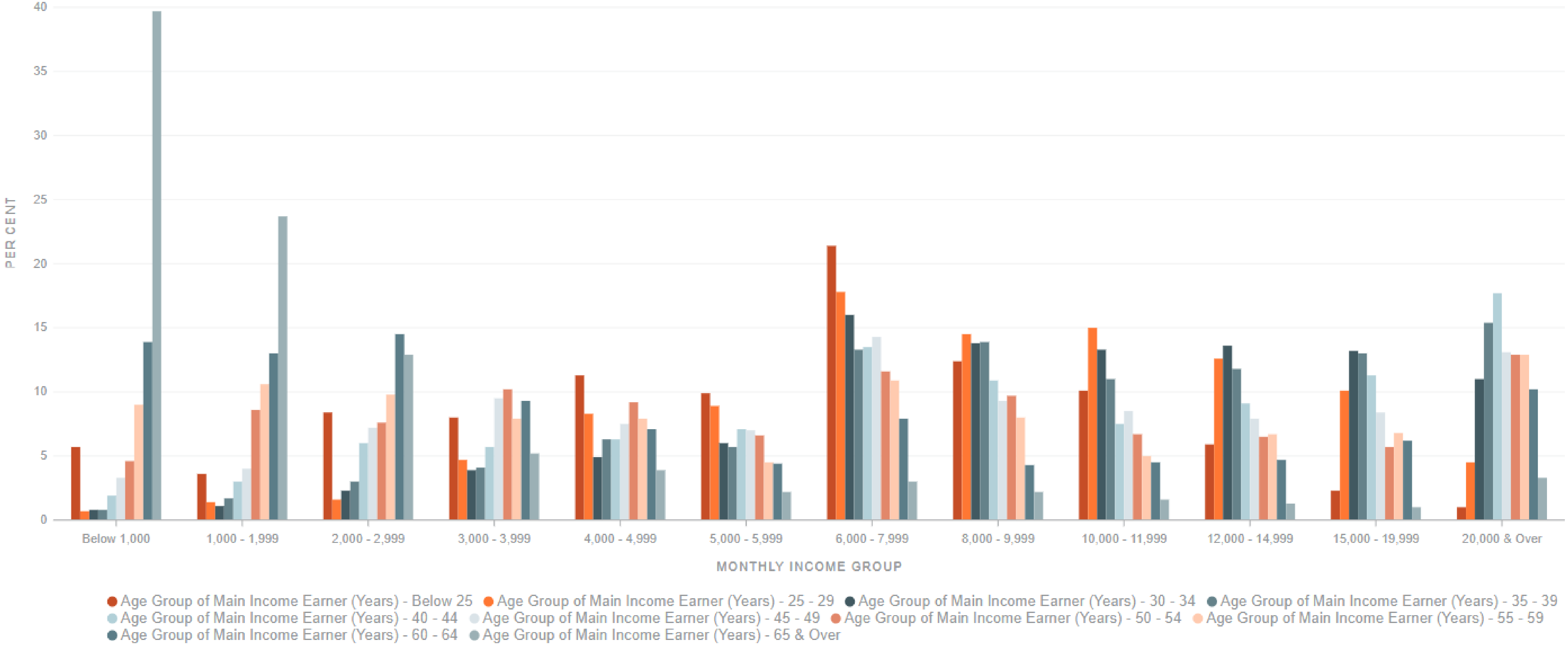
- Measures of central tendency provide descriptive information about the single numerical value that is considered to be the most **typical** of the values of a quantitative variable (subject to natural changes).

- Three common measures of central tendency:
  - Mean : the arithmetic average
  - Median : the center point in a set of numbers
  - Mode : the most frequently occurring number

Comparisons between Mean, Median, and Mode

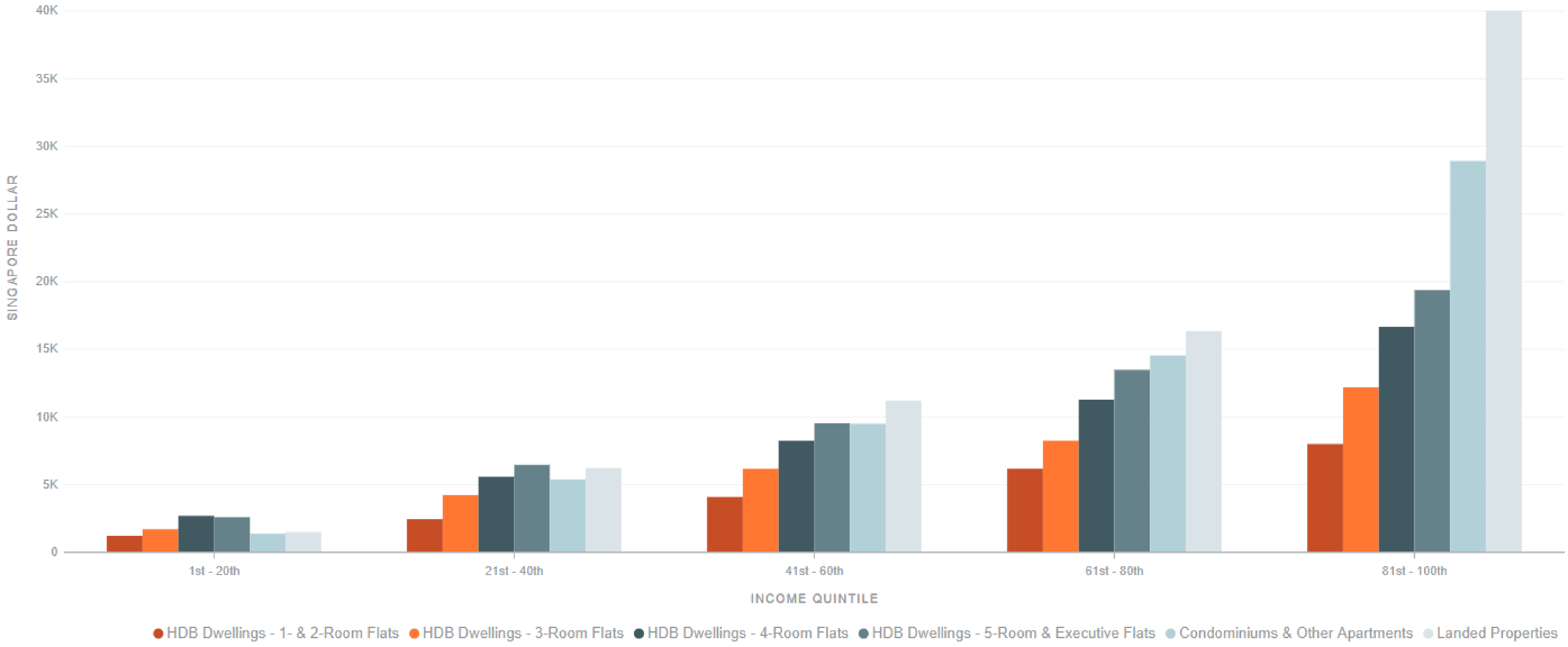


Households by Monthly Household Income and Age Group of Main Income Earner



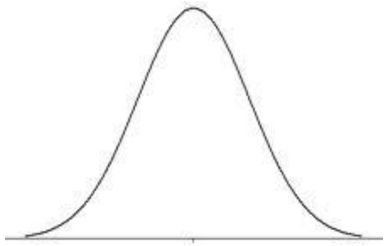
Source https://data.gov.sg/dataset/households-by-monthly-household-income-and-age-group-of-main-income-earner

Average Monthly Household Income by Income Quintile and Type of Dwelling

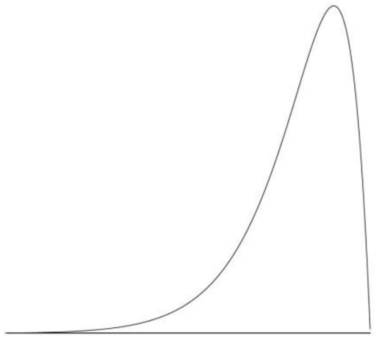


Source https://data.gov.sg/dataset/average-monthly-household-income-by-income-quintile-and-type-of-dwelling-quinquennial

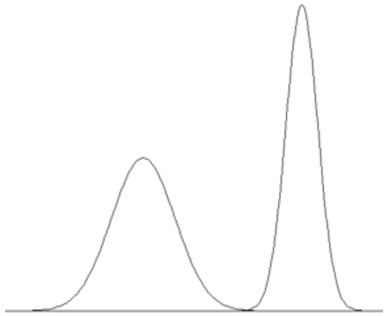
So… Which Statistic to Use?



Mean
The data is fairly symmetric.

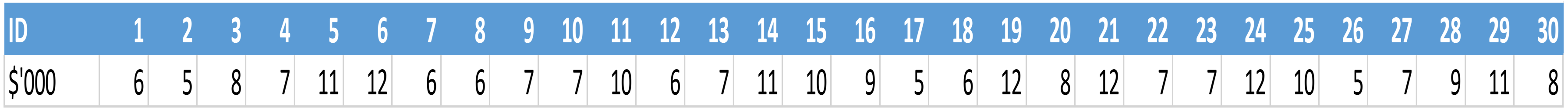


Median
The data is skewed.



Mode
The data shows two or more clusters.
The data is categorical.

Formal Definitions: Mean, Median, Mode (for n observation)



ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
$'000	6	5	8	7	11	12	6	6	7	7	10	6	7	11	10	9	5	6	12	8	12	7	7	12	10	5	7	9	11	8

$$Average = \bar{y} = \frac{\sum_{i=1}^{n}(y_i)}{n}$$

Median

$$if\ n\ is\ odd:\ Median = y_{\frac{n+1}{2}}$$

$$if\ n\ is\ even:\ Median = \frac{\left(y_{\frac{n}{2}} + y_{\frac{n}{2}+1}\right)}{2}$$

Mode :

Value of y_i which has the highest frequency.

Summary: Measuring Central Tendency

Average	How to calculate	When to use it
Mean	Add all the numbers in a data set together, and then divide by how many there are.	The data is fairly symmetric and shows just the one trend.
Median	Line up all the values in ascending order. If there are an odd number of values, the median is the one in the middle. If there are an even number of values, add the two middle ones together, and divide by two.	When the data is skewed because of outliers.
Mode	Choose the value(s) with the highest frequency. If the data is showing two clusters of data, report a mode for each group.	When you’re working with categorical data. When the data shows two or more clusters.

Measures of Variability

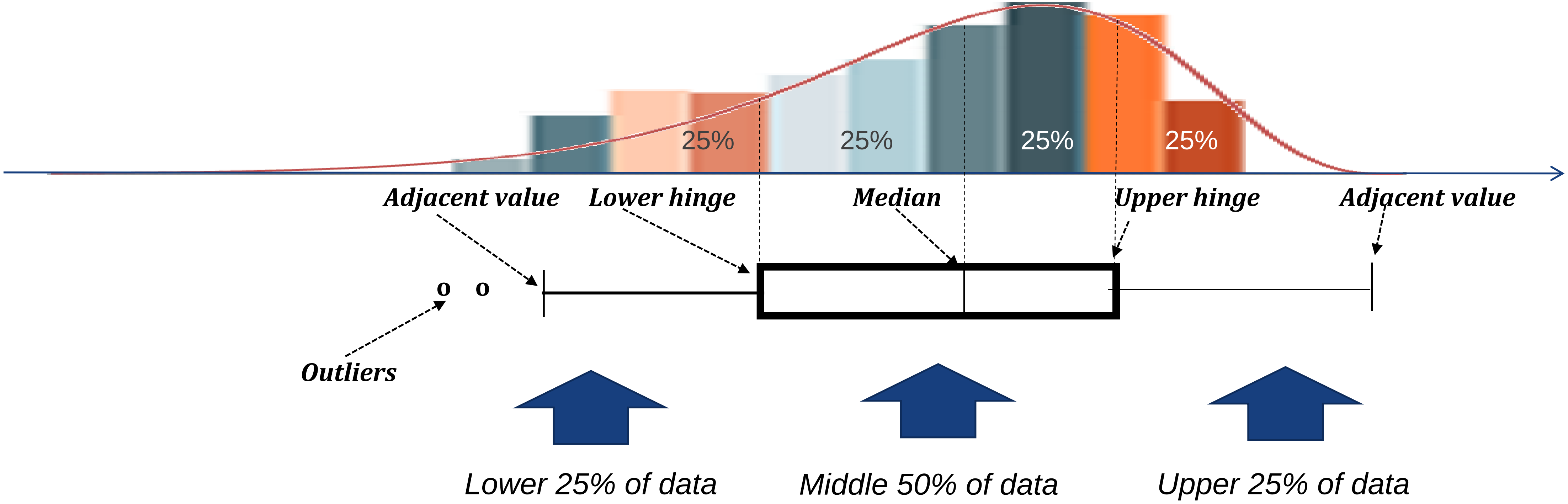
- Measures of variability describe the spread or dispersion of a set of data. They tell you how different your numbers tend to be for a sample/population (a group of individuals/data points).

- Some common measures of variability
  - Range : the difference between the largest value of a data set and the smallest value of a set.
  - Variance : the average of the squared deviations about the arithmetic mean for a set of numbers.
  - Standard Deviation : the square root of the variance

Statistics is
a study of variation(changes)

The Box & Whisker plot (boxplot)

- A Box plot is a graphical display that indicates the behaviour of measurements from a data sample
  - Indicates how “tightly spread” a sample may be
  - Indicates what values may be unusual - “outliers”
  - Allows to compare different data sets
  - Can be used with very small samples
  - Values of “Hinges” & “Whiskers” are calculated from data set



Describing the Variability

ID	2	17	26	1	7	8	12	18	4	9	10	13	22	23	27	3	20	30	16	28	11	15	25	5	14	29	6	19	21	24
$'000	5	5	5	6	6	6	6	6	7	7	7	7	7	7	7	8	8	8	9	9	10	10	10	11	11	11	12	12	12	12

$$Maximum \ = \max(yi)$$

$$Minumun \ = \min(yi)$$

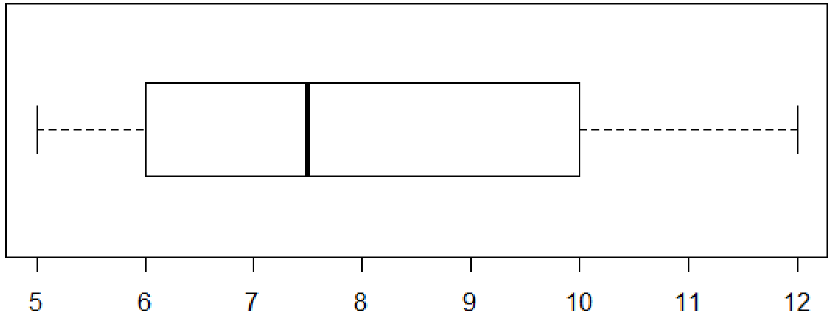
Range = maximum - minimum

$$Average = \ \bar{y} = 8.23333$$

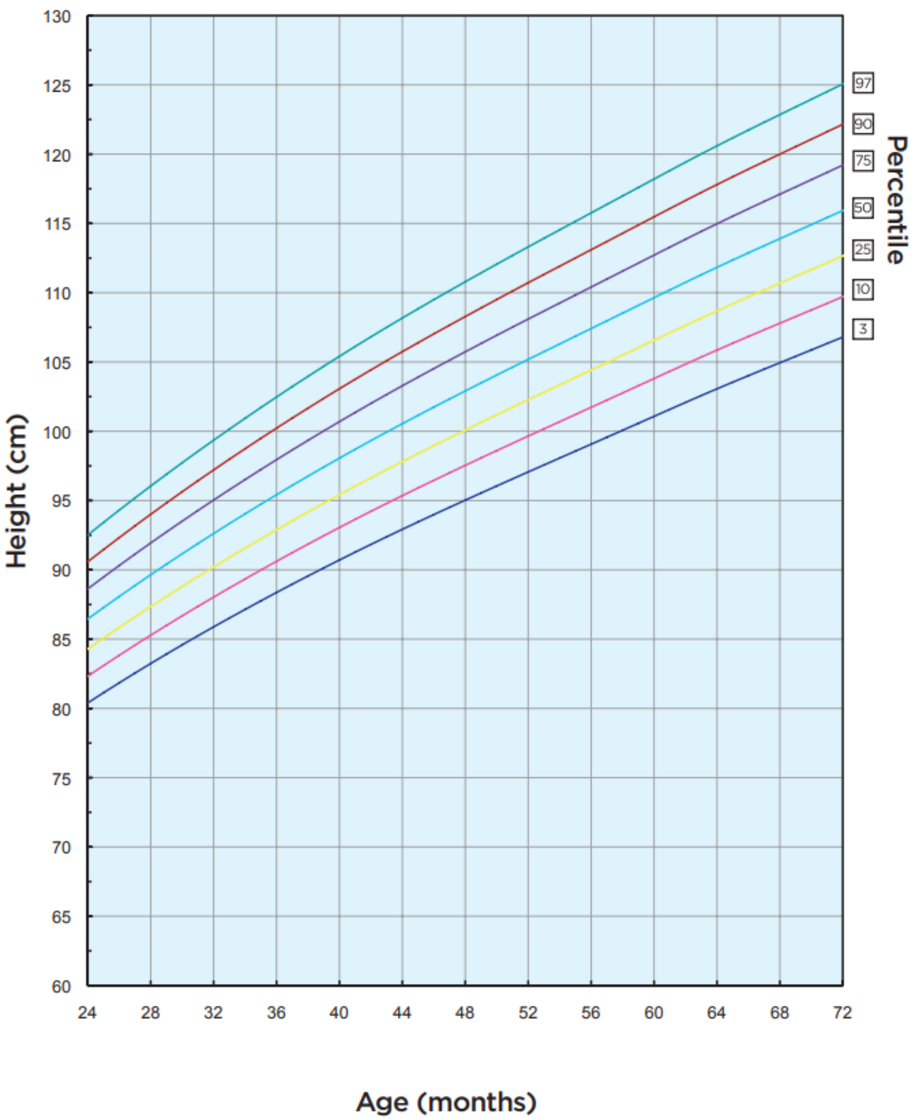
$$\text{Sample } Variance = s^2 = \frac{\sum_{i=1}^{n}(y_i - \ \bar{y})2}{n-1}$$

Sample Standard deviation = s

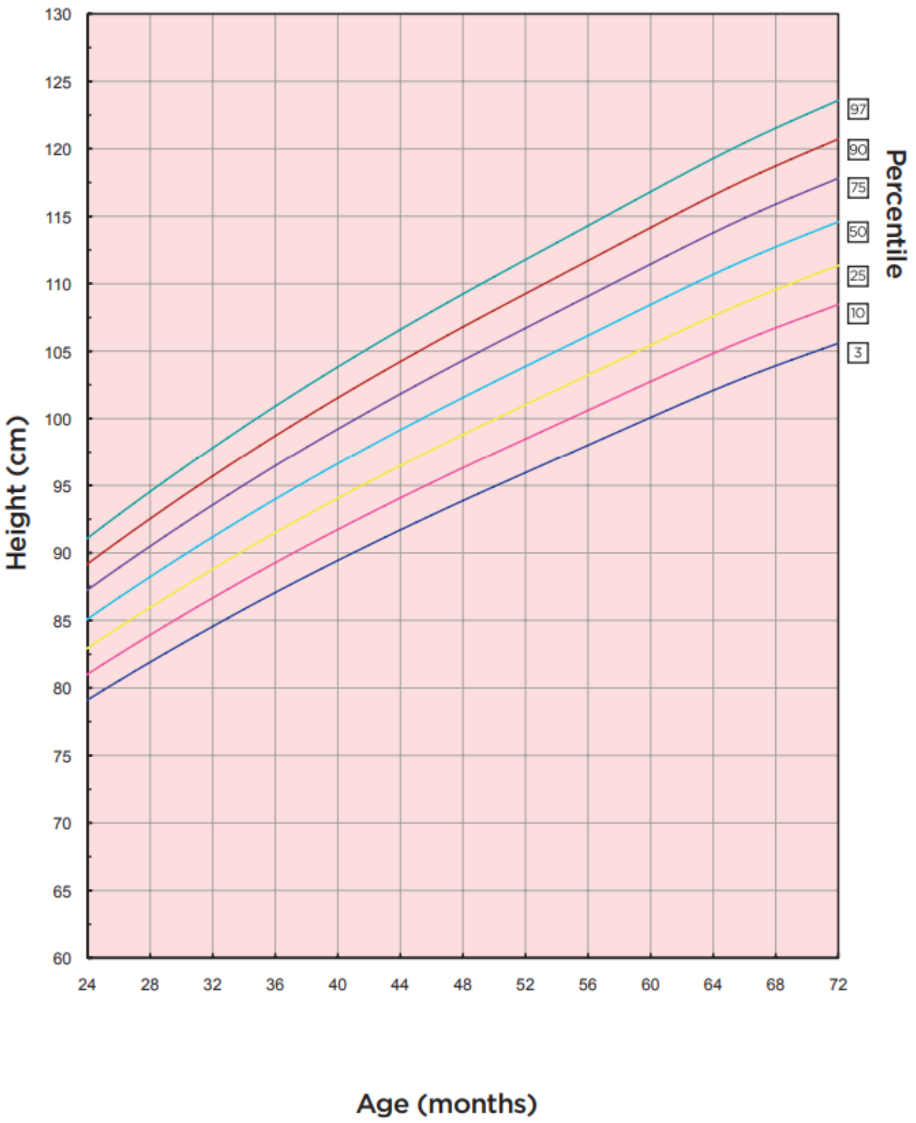
$$= \sqrt[2]{variance}$$



PERCENTILES OF HEIGHT-FOR-AGE BOYS AGED 24 TO 72 MONTHS



PERCENTILES OF HEIGHT-FOR-AGE GIRLS AGED 24 TO 72 MONTHS



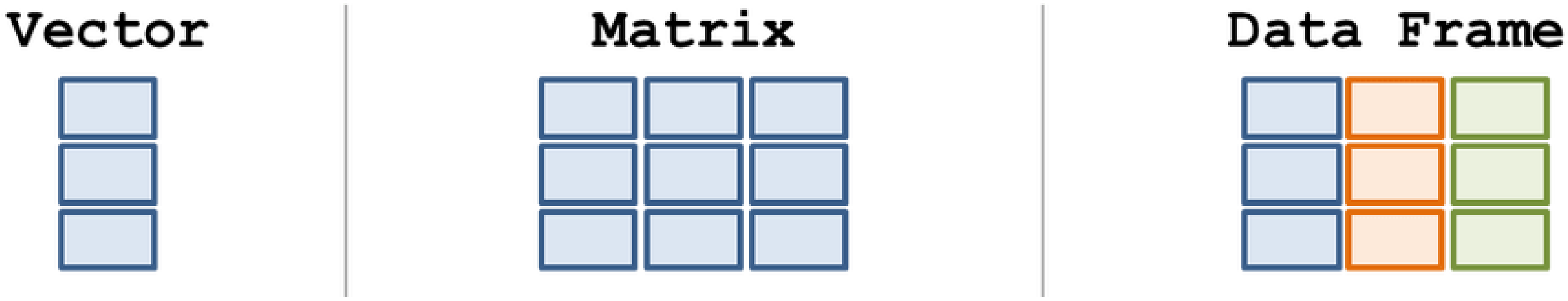
Analytics dedicated for those:

Source https://www.kiasuparents.com/kiasu/

Source https://www.healthhub.sg/sites/assets/Assets/Programs/screening/201810/pdf/health-booklet-2014.pdf

Data Collection & Summarization using R

R data structures



column
vector

V1 V2 V3
same data type

V3 V4 V5
different data type

Source: http://slow-data.com/r-training-basics/

Vector

- A vector is a list of values having the same value type

- Create a vector using `c()` function

```
     # Create a numeric vector
     > Bill =c(105, 111, 103, 122,
107, 119)


     # Create a character vector
     > Telco = c('Singtel',
'Starhub', 'M1', 'MyRepublic')
```

- We can retrieve values in a vector using square bracket `[]`

```
# Tip: R index starts from 1, not 0
     > Bill[2]

     [1] 111
     > Bill[c(1,3)]

     [1] 105 103
     > Telco[2:4]
     [1] "Starhub" "M1" "MyRepublic“

     # Try to identify differences
   between below two vectors:
     > NDay = c(9, 'Aug', 1965)
     > Nday = c('D'=9,'M'='Aug','Y'=
   1965)
```

Operations on vector

- Operations on a vector work element-wise, i.e. they operate on each element:

```
  # Create a numeric vector to
store workshop marks
  > Marks = c(52, 37, 41, 32,
31)


  # Operations on 'Marks' and
store it on 'MarksAdj'
  > MarksAdj = Marks + 20
  > MarksAdj
  [1] 72 57 61 52 51
```

```
  > MarksAdj > 60
  [1]  TRUE FALSE  TRUE FALSE
FALSE
```

Matrix

- Matrix can be created using **matrix()** function

```
  # Create a matrix
  > matrix(1:9, nrow=3, ncol=3)
       [,1] [,2] [,3]
  [1,]    1    4    7
  [2,]    2    5    8
  [3,]    3    6    9


  # Can only specify one 
dimension
  > matrix(c(2, 5, 7, 4), 
nrow=2)
       [,1] [,2]
  [1,]    2    7
  [2,]    5    4
```

- Operators on matrix also work element-wise:

```
> mx = matrix(1:9, nrow=3)
> my = mx + 1
> mx + my
     [,1] [,2] [,3]
[1,]    3    9   15
[2,]    5   11   17
[3,]    7   13   19
```

List

- List can be created using **list()** function

- It is similar to vector, but a list can contains various type of data

```
    # Not encouraged - data type coerced to
string
    > NDay = c(9, 'Aug', 1965)
    > NDay
    [1] "9"    "Aug"   "1965"
    > NDay[1]
    [1] "9"

    # Use list for mixed data types
    > NDay=list('D'=9,'M'='Aug','Y'= 1965)
    > NDay=list('D'=9,'M'='Aug','Y'= 1965)
    > class(NDay)
    [1] "list"
```

- You can access the elements in two ways:

```
> NDay
$D
[1] 9
$M
[1] "Aug"
$Y
[1] 1965
# Using square bracket
> NDay[1]
$D
[1] 9
# Using a $ followed by label
> NDay$M
[1] "Aug"
```

- You can check the number of elements in a list by

```
> length(NDay)
[1] 3
```

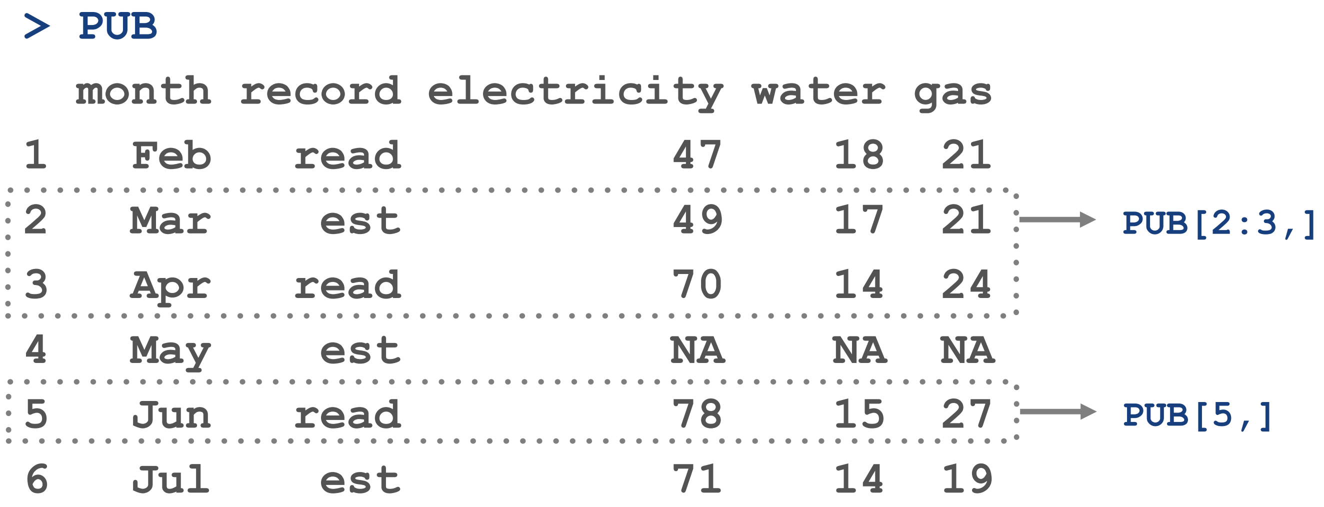
Data frame

- A data frame is used to store data table
- It is a list of vectors with equal length
- Use `data.frame()` to create a data frame

```
 # Create the vectors
 > month = c('Feb','Mar','Apr','May','Jun','Jul')
 > record = c('read','est','read','est','read','est')
 > electricity = c(47,49,70,NA,78,71)
 > water = c(18,17,14,NA,15,14)

 > gas = c(21,21,24,NA,27,19)
 # Combine all the vectors into a data frame
 > PUB = data.frame(month,record,electricity,water,gas,
stringsAsFactors=FALSE)
```

Accessing elements (rows)

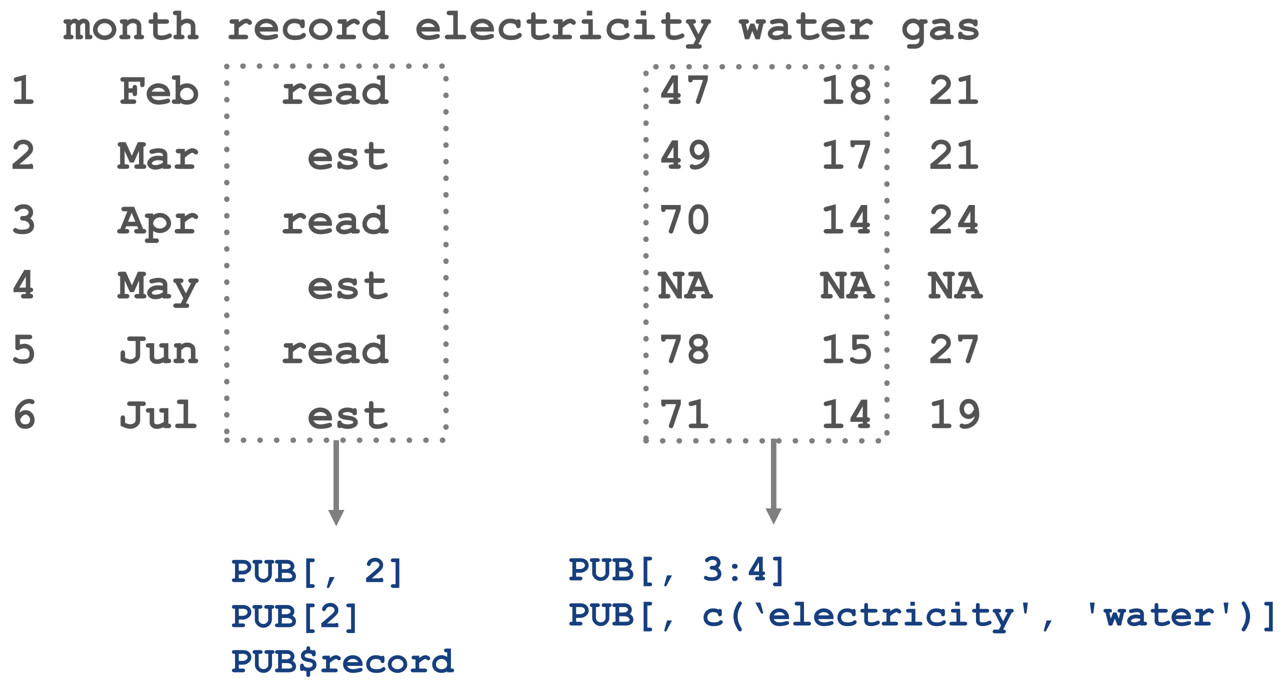


```
> PUB
  month record electricity water gas
1   Feb   read          47    18  21
2   Mar    est          49    17  21
3   Apr   read          70    14  24
4   May    est          NA    NA  NA
5   Jun   read          78    15  27
6   Jul    est          71    14  19
```

PUB[2:3,]

PUB[5,]

Accessing elements (columns)



```
  month record electricity water gas
1   Feb   read          47    18  21
2   Mar    est          49    17  21
3   Apr   read          70    14  24
4   May    est          NA    NA  NA
5   Jun   read          78    15  27
6   Jul    est          71    14  19
```

```
PUB[, 2]
PUB[2]
PUB$record
```

```
PUB[, 3:4]
PUB[, c(‘electricity', 'water')]
```

Accessing elements (single item)

```
  month record electricity water gas
1   Feb   read          47    18  21
2   Mar    est          49    17  21
3   Apr   read          70    14  24
4   May    est          NA    NA  NA
5   Jun   read          78    15  27
6   Jul    est          71    14  19
```

```
PUB[2, 3],
PUB$electricity[2]
```

Factors

- A special data type in R for categorical values
- Efficient storage for characters; advantages for working with modelling and graphing functions
- Use `factor()` to encode a vector as a factor

```
# Check PUB’s record vector
> PUB$record

[1] "read" "est"  "read" "est"  "read" "est"
# Encode ‘record’ as a factor
> PUB$record = factor(PUB$record)
> PUB$record
[1] read est  read est  read est
Levels: est read
```

Data Preparation

- In reality, often there is a need to perform further data preparation
- Recode values into category
- Handle missing values
- Create new variables
- Create subset

Recode values

- Make a note on water usage
- If water price < 15, marked as ‘low’; if water price ≥ 15, marked as ‘high’
- Recode values in category

```
# Check water vector
> PUB$water

[1] 18 17 14 NA 15 14

> PUB$water >= 15

[1]  TRUE  TRUE FALSE    NA  TRUE FALSE

# Recode water usage (based on price) into category
> PUB$water_use[PUB$water>=15] = 'high'
> PUB
 month record electricity water gas water_use
1   Feb   read          47    18  21      high
2   Mar    est          49    17  21      high
3   Apr   read          70    14  24      <NA>
4   May    est          NA    NA  NA      <NA>
5   Jun   read          78    15  27      high
6   Jul    est          71    14  19      <NA>
```

Recode values

- Make a note on water usage
- If water price < 15, marked as ‘low’; if water price ≥ 15, marked as ‘high’
- Recode values in category

```
# Check water vector
> PUB$water

[1] 18 17 14 NA 15 14

> PUB$water < 15

[1] FALSE FALSE  TRUE    NA FALSE  TRUE

# Recode water usage (based on price) into category
> PUB$water_use[PUB$water<15] = 'low'
> PUB
  month record electricity water gas water_use
1   Feb   read          47    18  21      high
2   Mar    est          49    17  21      high
3   Apr   read          70    14  24       low
4   May    est          NA    NA  NA      <NA>
5   Jun   read          78    15  27      high
6   Jul    est          71    14  19       low
```

Missing values

- Incomplete values in data collection are very common
- Need to find ways to handle such situations
- Few approaches to use

```
# Calculate the mean without handling missing values
> mean(PUB$gas)

[1] NA
# Calculate the mean with NA removed
> mean(PUB$gas, na.rm=TRUE)

[1] 22.4
# Remove rows where NA presents
> PUB_NoNA = na.omit(PUB)


> PUB_NoNA
  month record electricity water gas water_use
1   Feb   read          47    18  21      high
2   Mar    est          49    17  21      high
3   Apr   read          70    14  24       low
5   Jun   read          78    15  27      high
6   Jul    est          71    14  19       low
```

Create new variables

- Add the price of electricity, water and gas into a total

```
# Calculate the total PUB price
> attach(PUB)
> PUB$total = electricity + water + gas
> detach(PUB)
> PUB
  month record electricity water gas water_use total
1   Feb   read          47    18  21      high    86
2   Mar    est          49    17  21      high    87
3   Apr   read          70    14  24       low   108
4   May    est          NA    NA  NA      <NA>    NA
5   Jun   read          78    15  27      high   120
6   Jul    est          71    14  19       low   104
```

Create new subset

- Focus on the months where readings were actually taken
- Want to look at only electricity
- Create a subset for that

```
# Create a subset
> electricity_read = subset(PUB,PUB$record == 'read',
select=c('electricity'))
```

```
> electricity_read
  electricity
1          47
3          70
5          78
```

Exporting data from R

- Use write.table() to output data frame to a text file or csv file

- Use save() to save data frame into rdata file

```
  # Write data frame ‘PUB’ into ‘PUB.csv’
  > write.table(PUB, 'PUB.csv', sep=',', row.names =
FALSE)
```

```
  # Save data frame ‘PUB’ into rdata file
  > save(PUB, file='PUB.Rda')
```

```
  # To load the rdata file, just use:
  > load('PUB.Rda')
```

Importing data to R

- Instead of manually entering data, you may want to import data from csv file

- You could also use the GUI in RStudio to import data

```
# Import ‘PUB.csv’
> nPUB = read.table('PUB.csv', header=TRUE, sep=',')
```

```
> nPUB
  month record electricity water gas water_use total
1   Feb   read          47    18  21      high    86
2   Mar    est          49    17  21      high    87
3   Apr   read          70    14  24       low   108
4   May    est          NA    NA  NA      <NA>    NA
5   Jun   read          78    15  27      high   120
6   Jul    est          71    14  19       low   104
```

Useful R functions

Function	Description
c()	Combine values into a vector
data.frame()	Create a data frame
factor()	Encode a vector as factor
read.table()	Reads a file in table format and creates a data frame from it
is.na()	Indicate which elements are missing
na.omit()	Remove observations with missing values
subset	Select variables and observations
[]	Operators acting on vectors and data frames to extract or replace parts.
str()	See the structure of dataset
dim()	Show the dimension of dataset
head()	View first six rows
tail()	View last six rows

Useful R functions

Function	Description
`rm()`	Remove variable/object from environment
`rm(list=ls())`	Remove all objects from environment
`within(df,rm(x,y))`	Remove vector ‘x’ and ‘y’ from dataframe ‘df’

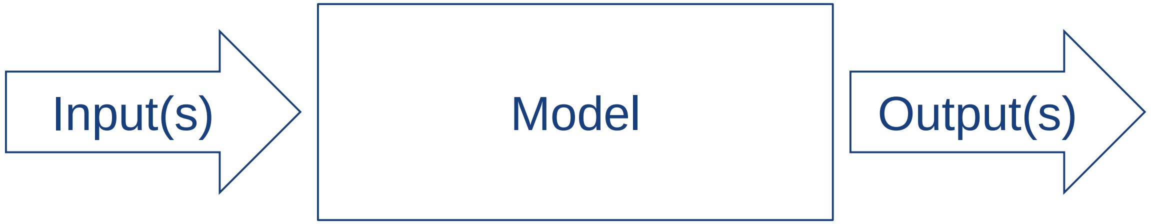
End of Lecture Notes

What’s a (statistical/machine-learning) “model”?

What’s a “model”?

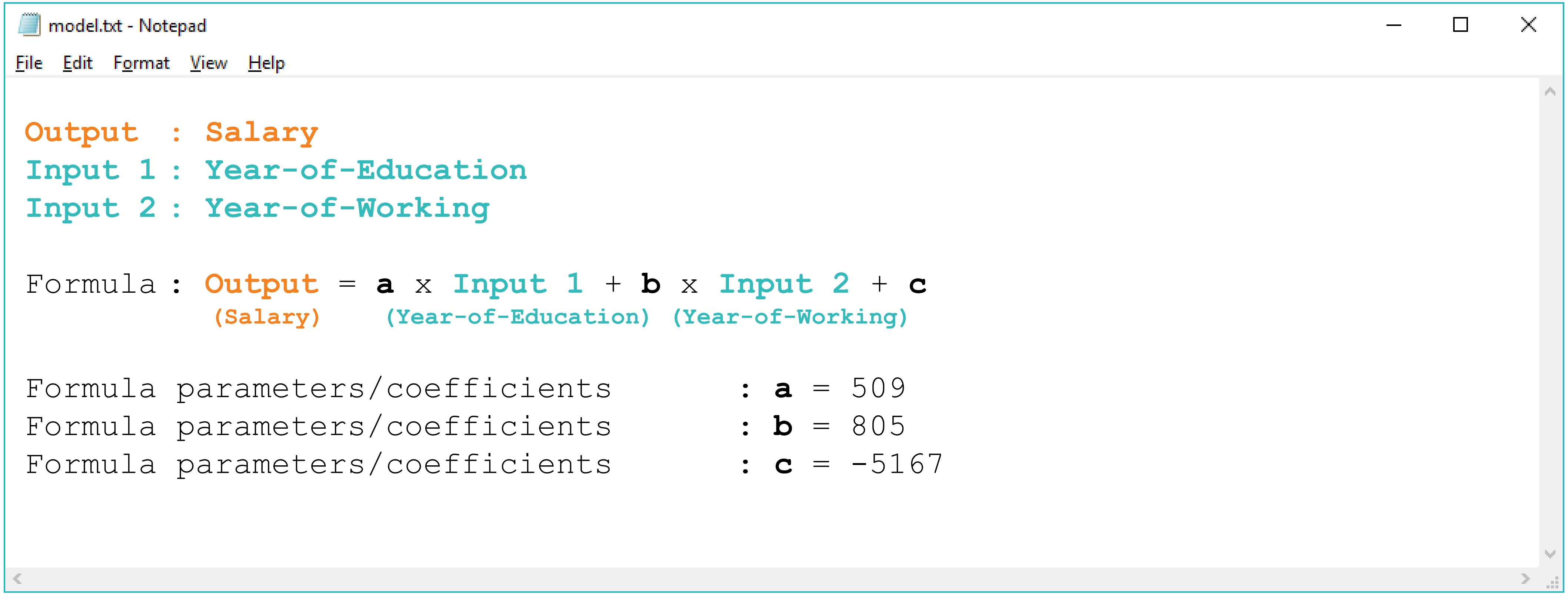
A **model** is a piece of ***knowledge*** (our understanding of the ***world/domain***), which can be (re)used to generate/predict **outcome results** based on **input observations**.

Technically, it’s a function (white or black box), which maps input(s) to output(s)



What’s a “model”?

A model could be considered just as a tangible text file stored in computer/server, e.g. model.txt



```
Output   : Salary
Input 1  : Year-of-Education
Input 2  : Year-of-Working

Formula : Output = a x Input 1 + b x Input 2 + c
          (Salary)   (Year-of-Education) (Year-of-Working)

Formula parameters/coefficients       : a = 509
Formula parameters/coefficients       : b = 805
Formula parameters/coefficients       : c = -5167
```

Salary Data