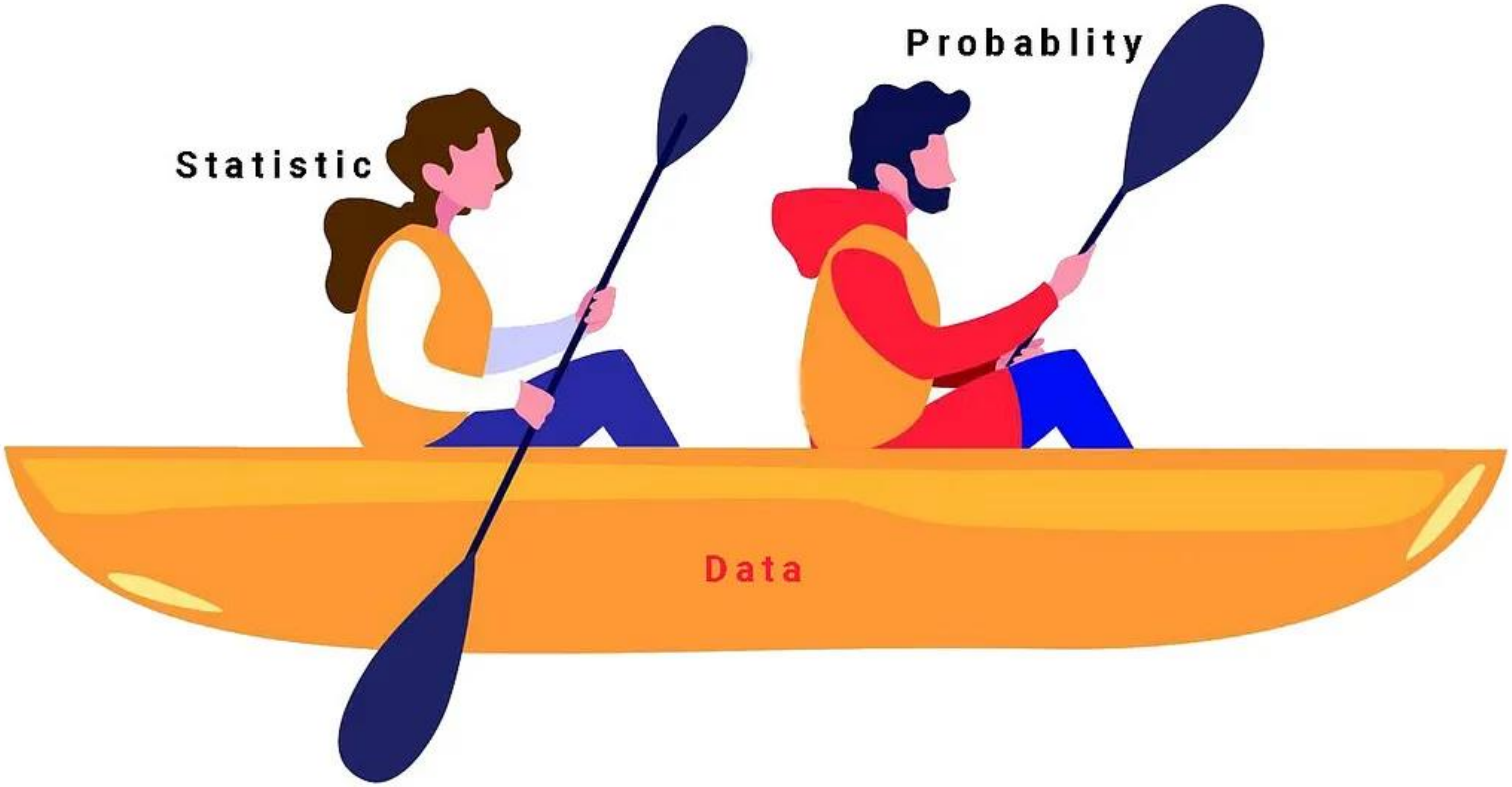
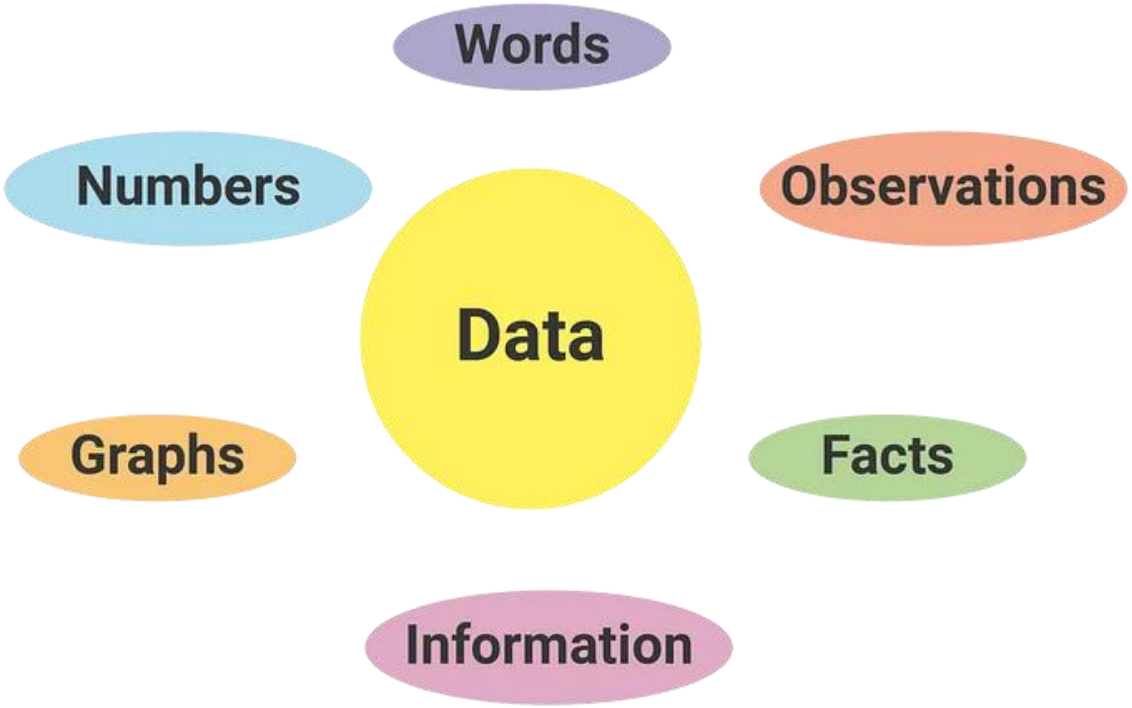


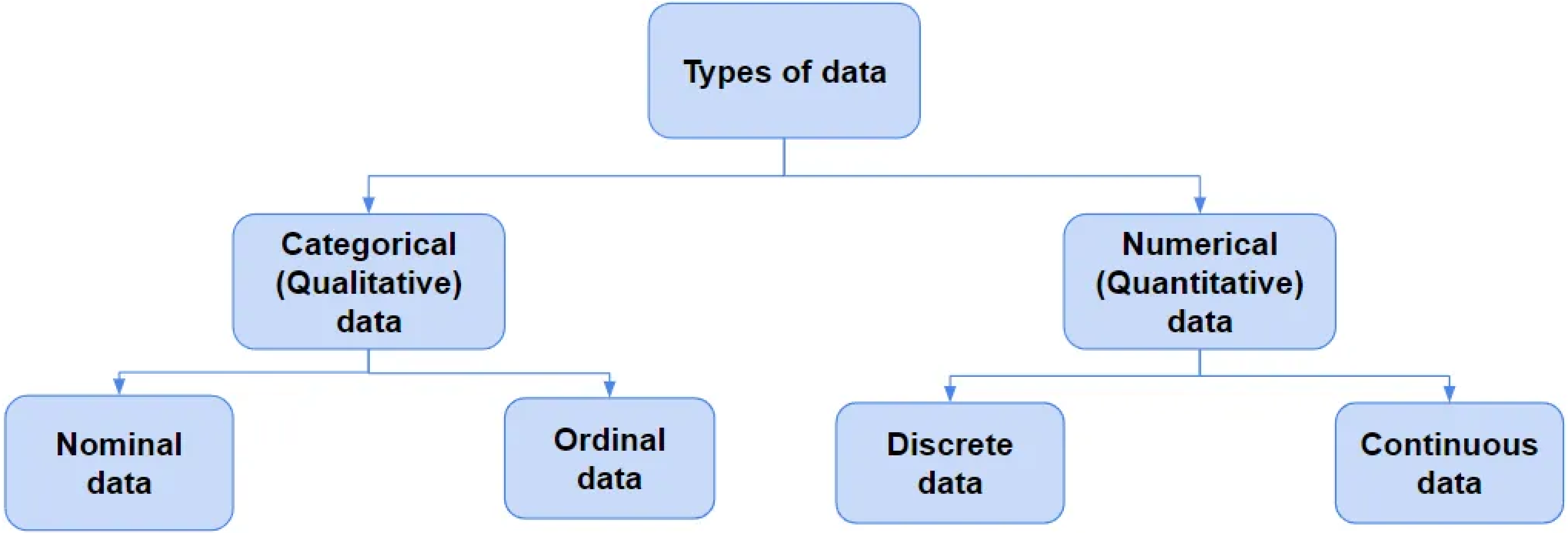
Balachandar K

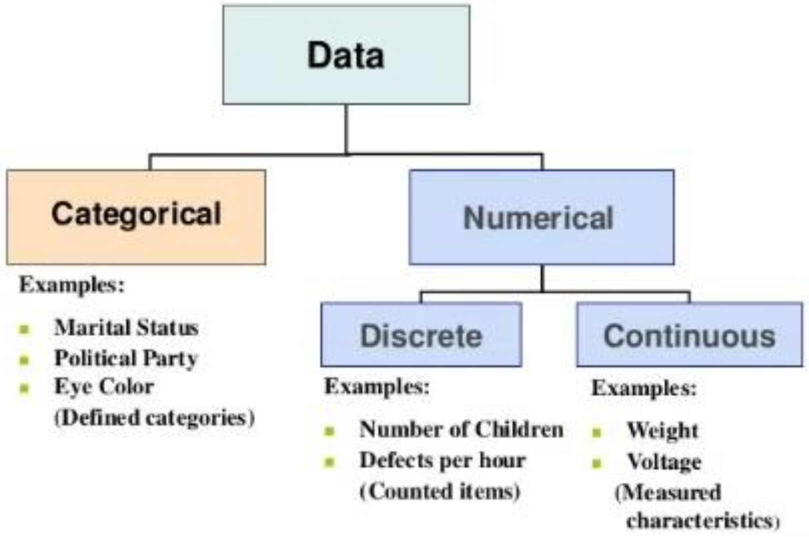
Important one….

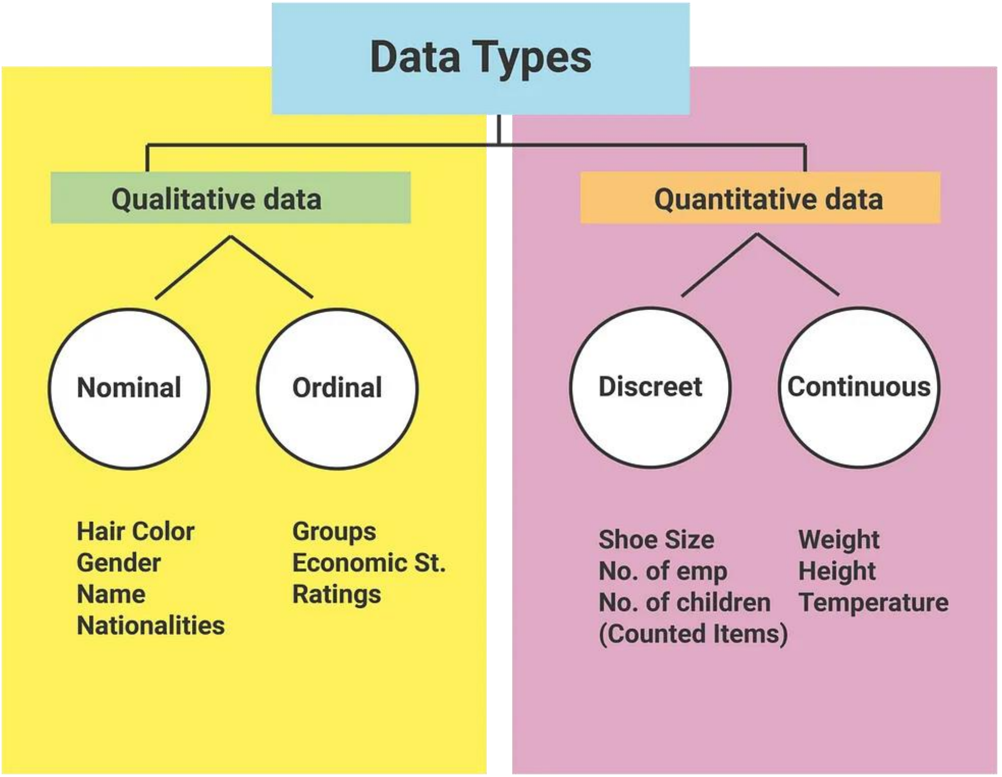
- *Data Types*
- *Probability & Bayes’ Theorem*
- *Measures of Central Tendency*
- *Skewness*
- *Kurtosis*
- *Measures of Dispersion*
- *Covariance*
- *Correlation*
- *Probability Distributions*
- *Hypothesis Testing*
- *Regression*

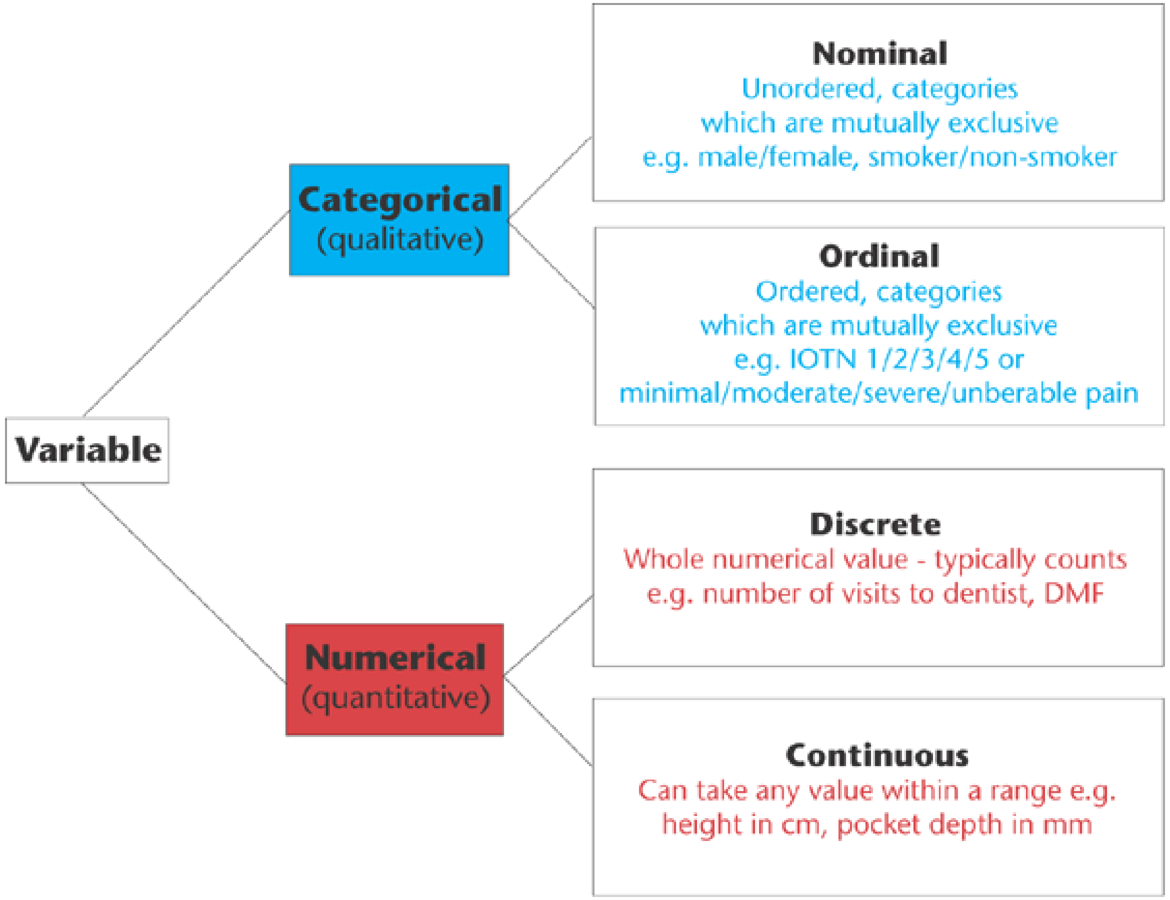
What is data ?

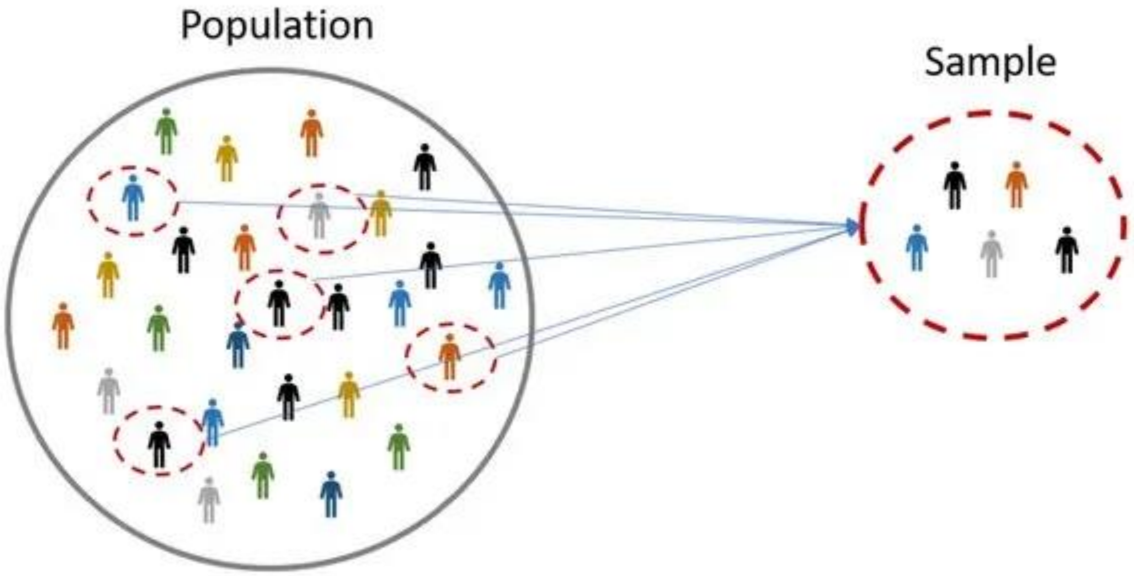

Words
Numbers
Observations
Data
Graphs
Facts
Information


Types of data
Categorical (Qualitative) data
Numerical (Quantitative) data
Nominal data
Ordinal data
Discrete data
Continuous data


Data
Categorical
Numerical
Examples:
Marital Status
Political Party
Eye Color
(Defined categories)
Discrete
Continuous
Examples:
Number of Children
Defects per hour
(Counted items)
Examples:
Weight
Voltage
(Measured characteristics)


Data Types
Qualitative data
Quantitative data
Nominal
Ordinal
Discreet
Continuous
Hair Color
Gender
Name
Nationalities
Groups
Economic St.
Ratings
Shoe Size
No. of emp
No. of children
(Counted Items)
Weight
Height
Temperature


Variable
Categorical
(qualitative)
Nominal
Unordered, categories
which are mutually exclusive
e.g. male/female, smoker/non-smoker
Ordinal
Ordered, categories
which are mutually exclusive
e.g. IOTN 1/2/3/4/5 or
minimal/moderate/severe/unberable pain
Numerical
(quantitative)
Discrete
Whole numerical value - typically counts
e.g. number of visits to dentist, DMF
Continuous
Can take any value within a range e.g.
height in cm, pocket depth in mm


Population
Sample

Measure of central tendency

Mean

Where you add up all the numbers and then divide by the amount of numbers you added.

Example :
13, 18, 13, 14, 13, 16, 14, 21, 13.

(13 + 18 + 13 + 14 + 13 + 16 + 14 + 21 + 13) ÷ 9 = 15

Median

It is the "middle" value in the list of numbers.

Example :
13, 18, 13, 14, 13, 16, 14, 21, 13

Arrange the number
13, 13, 13, 13, 14, 14, 16, 18, 21

There are nine numbers in the list.
(9 + 1) ÷ 2 = 10 ÷ 2 = 5th number.

13, 13, 13, 13, 14, 14, 16, 18, 21
The median is 14.

Mode

The number or numbers that occur most often in a set of numbers.

Example :
13, 18, 13, 14, 13, 16, 14, 21, 13

The number that is repeated more often is 13.

Range

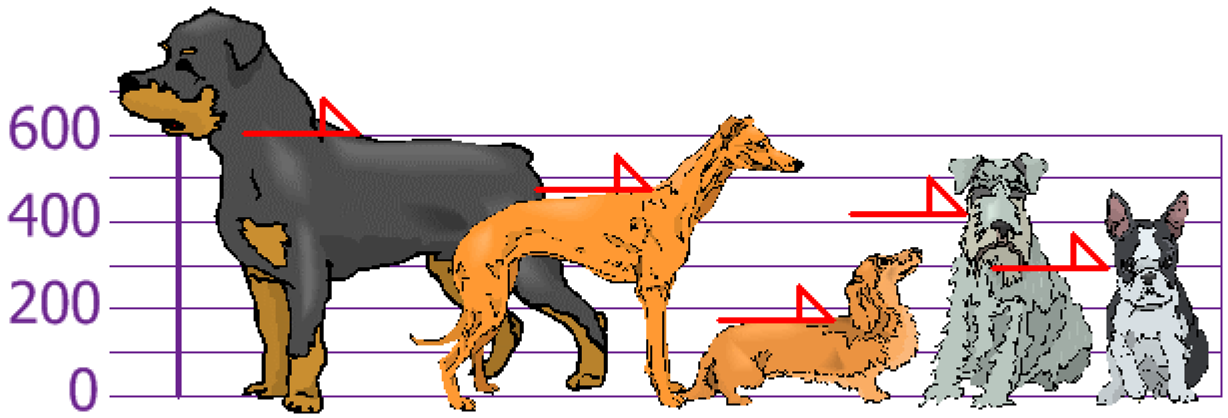
The difference between the highest and the lowest numbers in a set of numbers.

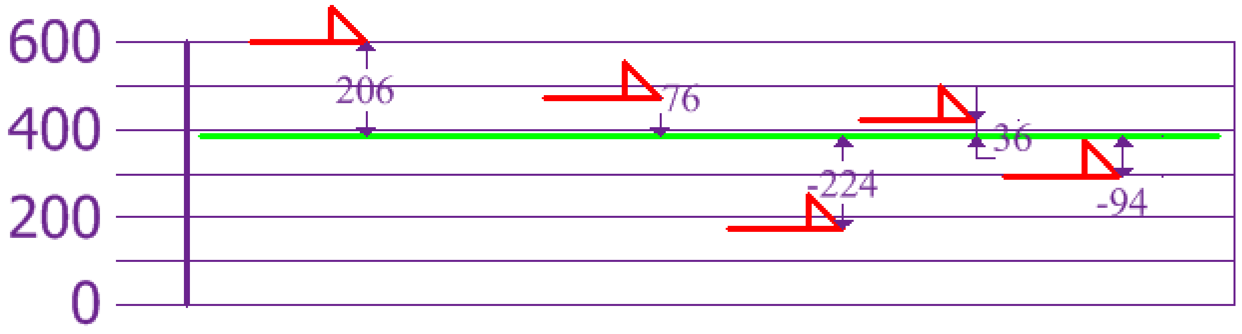
Example :
13, 18, 13, 14, 13, 16, 14, 21, 13

The largest value in the list is 21, and the smallest is 13,

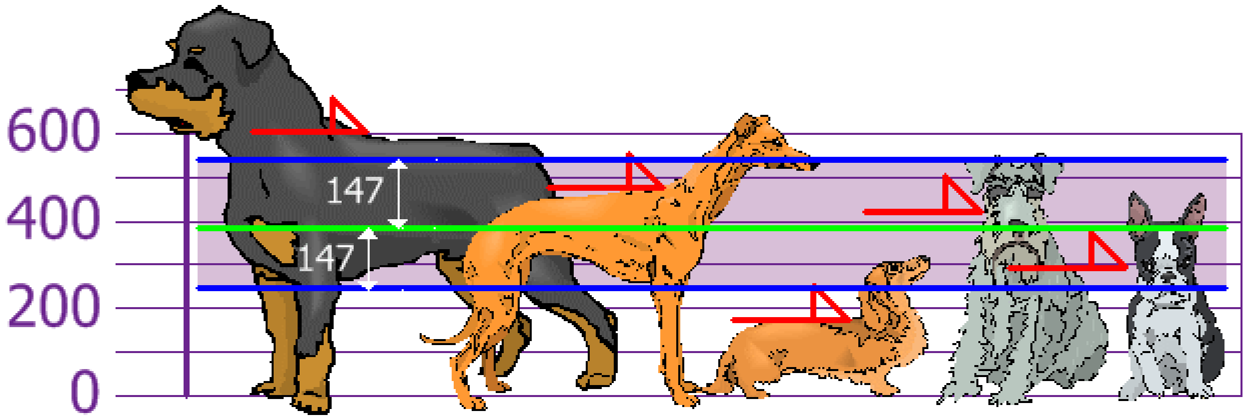
so the range is 21 – 13 = 8.

Height & Average Height (mean)





Variance & SD



Variance $\sigma^2 = (206^2 + 76^2 + (-224)^2 + 36^2 + (-94)^2) / 5$

$\Rightarrow (42436 + 5776 + 50176 + 1296 + 8836) / 5 \Rightarrow 108520 / 5 \Rightarrow 21704$

So the Variance is 21,704

Standard Deviation $\sigma = \sqrt{21704} = 147.32... = 147$

Statistics methods….

```
statistics.mean([1,2,3,4,4])
```

```
2.8
```

```
statistics.mode([1,1,2,3,3,3,3,4])
```

```
3
```

```
statistics.mode(["red","blue","red","red"])
```

```
'red'
```

```
statistics.median([1,3,5])
```

```
3
```

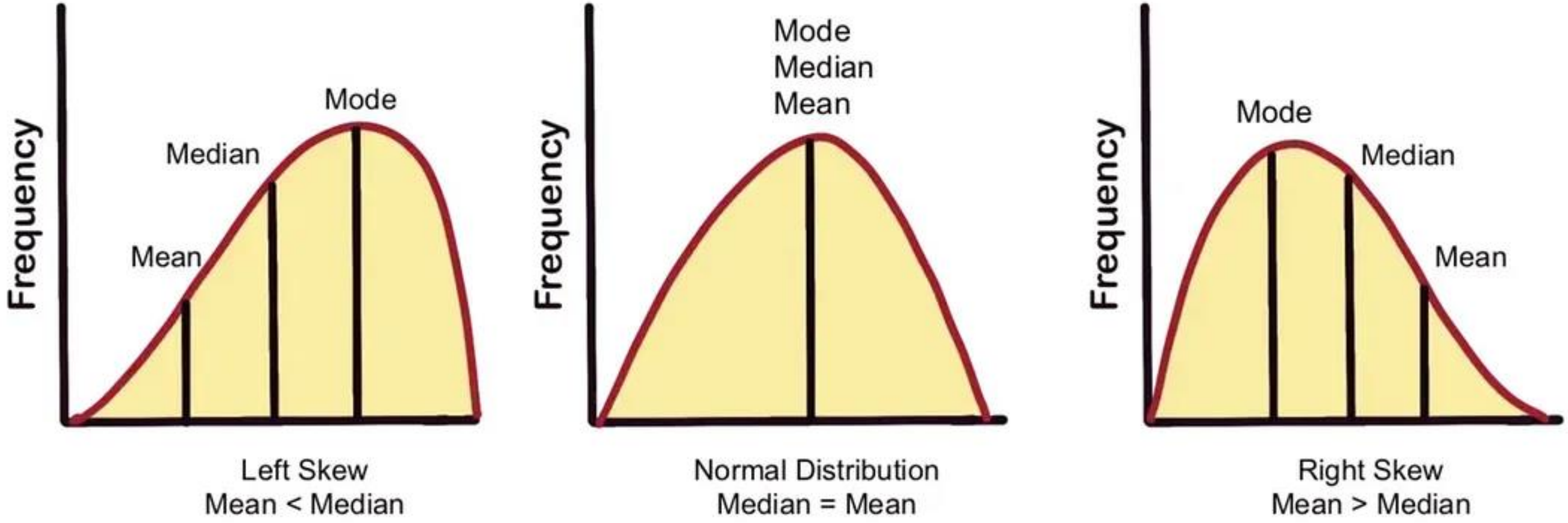
```
statistics.median([1,3,5,7])
```

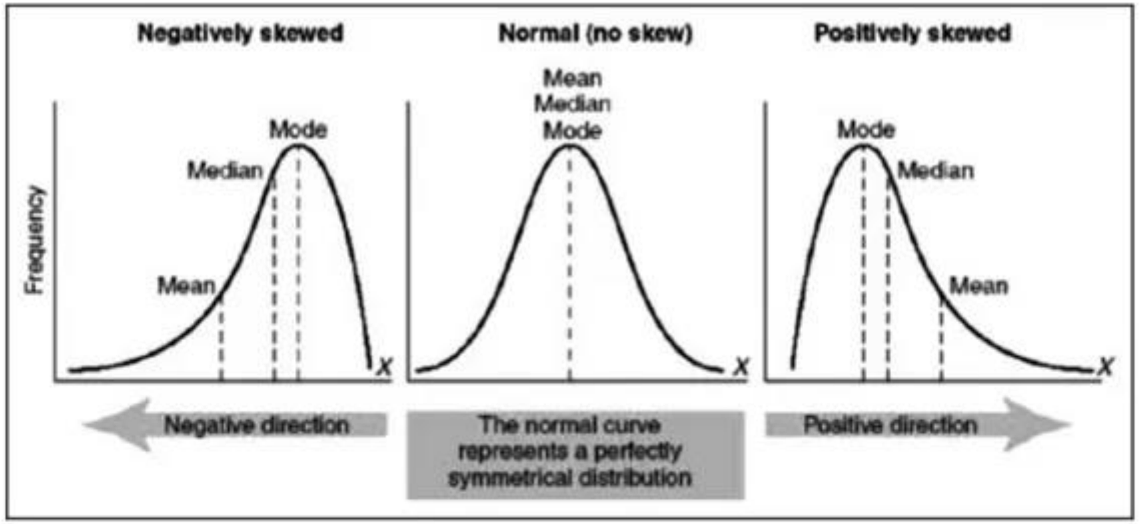
```
4.0
```

Measures of central tendency

Type of Variable	Best measure of central tendency
Nominal	Mode
Ordinal	Median
Interval/Ratio (not skewed)	Mean
Interval/Ratio (skewed)	Median

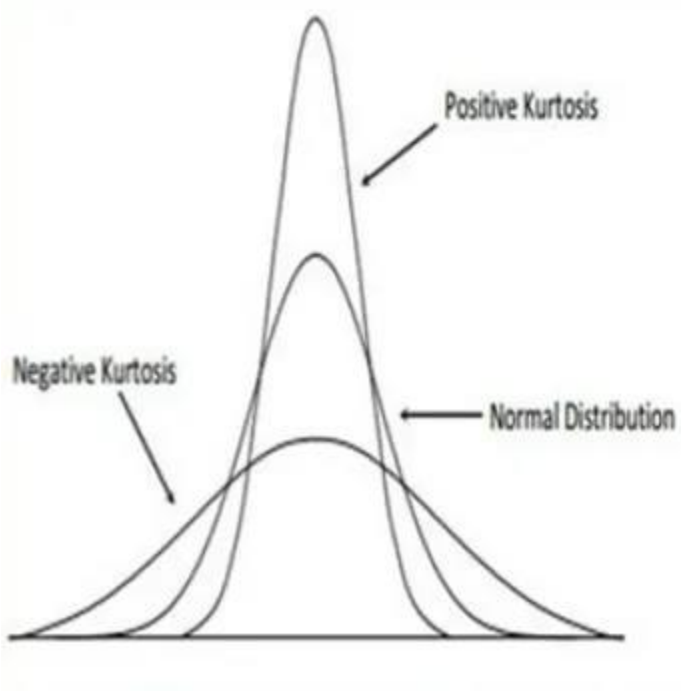
Frequency Distribution




Negatively skewed
Normal (no skew)
Positively skewed
Mean
Median
Mode
Frequency
Mode
Median
Mean
X
X
Mode
Median
Mean
X
Negative direction
The normal curve represents a perfectly symmetrical distribution
Positive direction

- Positive Skewness: It occurs when the Mean > Median < Mode. The tail is skewed to the right in this case, i.e outliers are skewed to the right.
- Negative Skewness: It occurs when the Mean < Median < Mode. The tail is skewed to the left, i.e the outliers are skewed to left

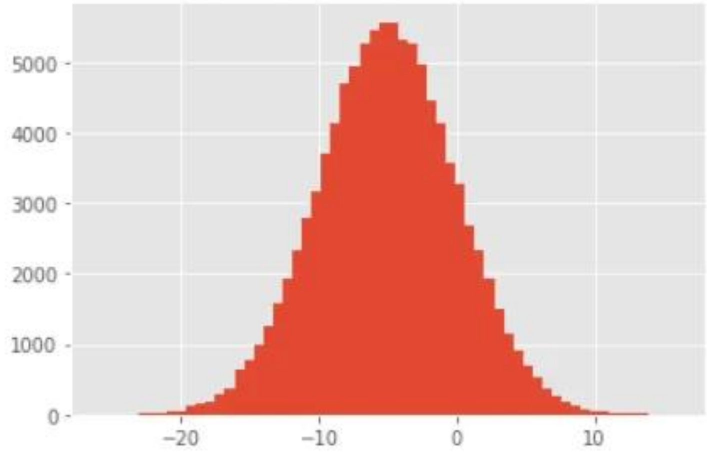
Kurtosis



Positive Kurtosis
Negative Kurtosis
Normal Distribution

```
import numpy as np
from scipy.stats import kurtosis
from scipy.stats import skew
import matplotlib.pyplot as plt
plt.style.use('ggplot')
data = np.random.normal(-5, 5, 100000)
plt.hist(data, bins=60)
print("skew : ",skew(data))
print("kurt : ",kurtosis(data))
```

```
skew :  -0.005168246772588942
kurt :  0.010248125871068048
```



Measures of Dispersion

```
data=[1,2,3,4,5,7,9]
#sample variance
statistics.variance(data)
```

```
7.9523809523809526
```

```
#population variance
statistics.pvariance(data)
```

```
6.816326530612245
```

```
#square root of sample variance
statistics.stdev(data)
```

```
2.819996622760558
```

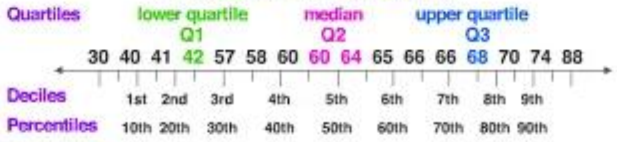
```
#square root of population variance
statistics.pstdev(data)
```

```
2.610809554642438
```

quartiles, deciles, percentiles

These are three common measures used in statistics to divide an ordered data set into equal parts.

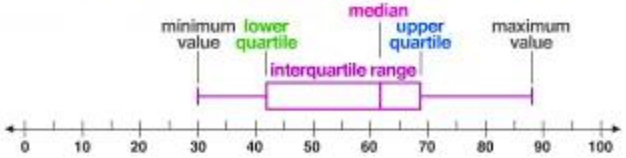
Quartiles = 4 equal parts, Deciles = 10 equal parts, Percentiles = 100 equal parts.

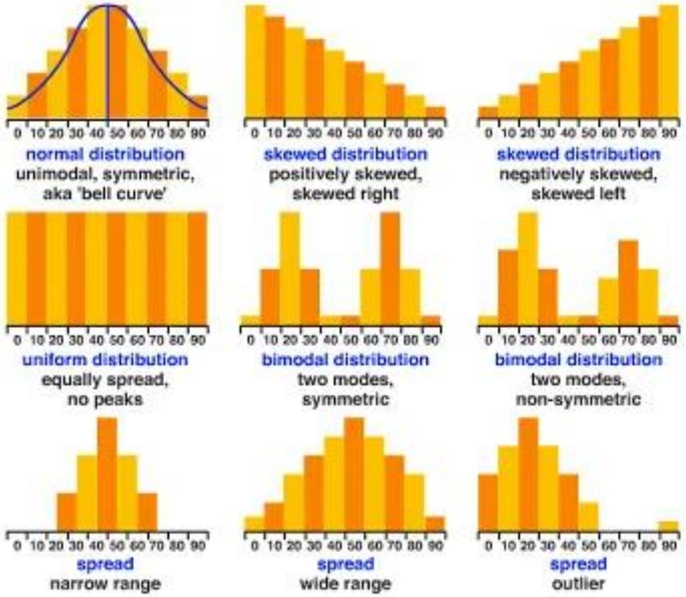


five number summary

The five number summary gives the minimum value, lower quartile, median, upper quartile, and the maximum value.

This is often represented in a box or box-and-whisker plot.

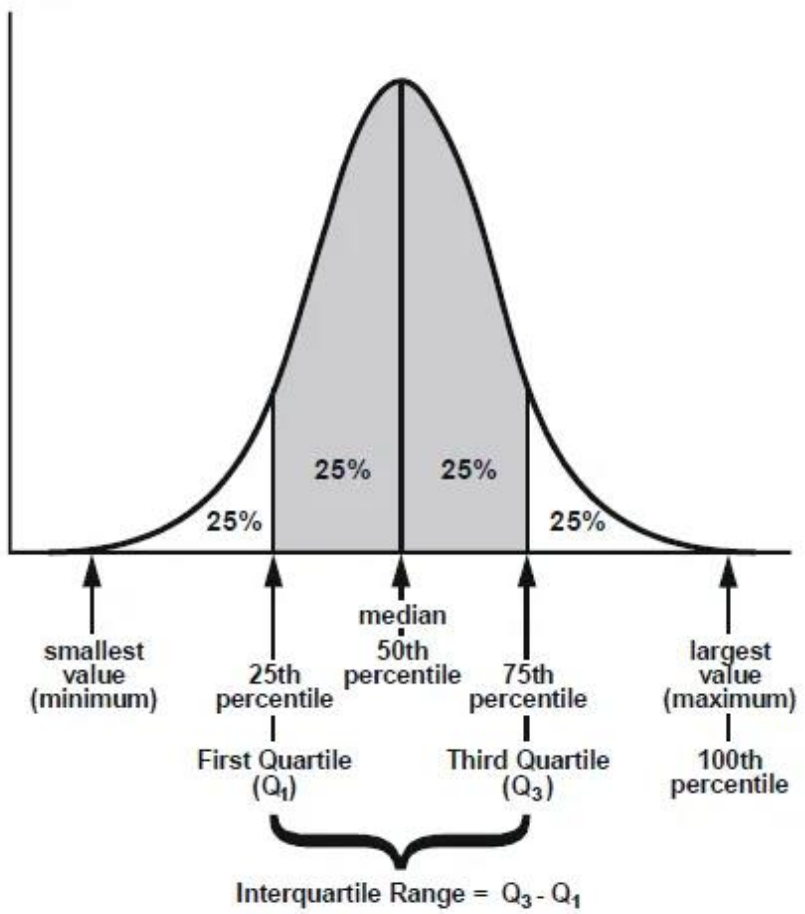



0 10 20 30 40 50 60 70 80 90
normal distribution
unimodal, symmetric,
aka 'bell curve'
0 10 20 30 40 50 60 70 80 90
skewed distribution
positively skewed,
skewed right
0 10 20 30 40 50 60 70 80 90
skewed distribution
negatively skewed,
skewed left
0 10 20 30 40 50 60 70 80 90
uniform distribution
equally spread,
no peaks
0 10 20 30 40 50 60 70 80 90
bimodal distribution
two modes,
symmetric
0 10 20 30 40 50 60 70 80 90
bimodal distribution
two modes,
non-symmetric
0 10 20 30 40 50 60 70 80 90
spread
narrow range
0 10 20 30 40 50 60 70 80 90
spread
wide range
0 10 20 30 40 50 60 70 80 90
spread
outlier

- import numpy as np
- arr1= np.array([[12,43,56],[78,88,95],[79,89,43], [101,34,67]])
- arr2 = np.array([5,6,7,12,34,67,89])
- #Mean function
- print("Mean:", np.mean(arr2))
- #Median function
- print("Median:",np.median(arr2))
- #Standard Deviation Function
- print("Standard Deviation:", np.std(arr2))
- #Variance Function
- print("Variance:",np.var(arr2))
- #Average Function
- print("Average:",np.average(arr2))
- #Percentile Function
- print("Percentile:",np.percentile(arr2,5,0))
- #Minimum Function
- print("Minimum element:",np.amin(arr))
- #Maximum Function
- print("Maximum element:",np.amax(arr))

- Mean: 31.428571428571427
- Median: 12.0
- Standard Deviation: 31.409084867994768
- Variance: 986.530612244898
- Average: 31.428571428571427
- Percentile: 5.3
- Minimum element: 12
- Maximum element: 101

Measure of Spread

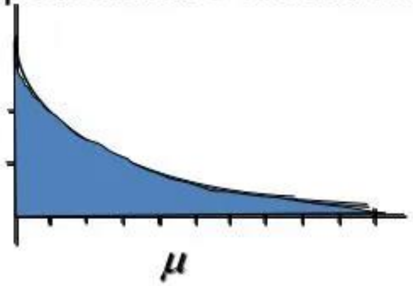


```
arr = [31, 35, 45, 49, 59, 69, 74, 79, 80, 81, 89, 94, 96, 99, 101, 104, 112,
117,119,127,134]

# First quartile (Q1)
Q1 = np.median(arr[:12])

# Third quartile (Q3)
Q3 = np.median(arr[12:])

# Interquartile range (IQR)
IQR = Q3 - Q1

print(IQR)
```

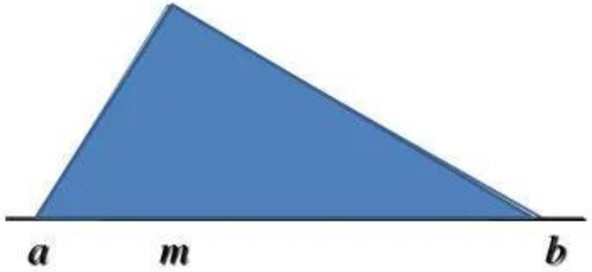
Continuous Probability Distributions

Uniform Distribution

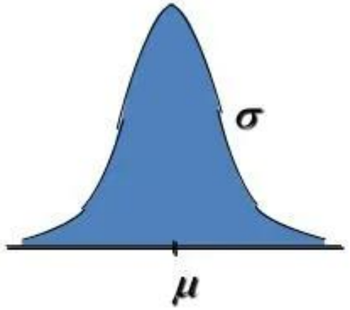
Exponential Distribution



Triangular Distribution



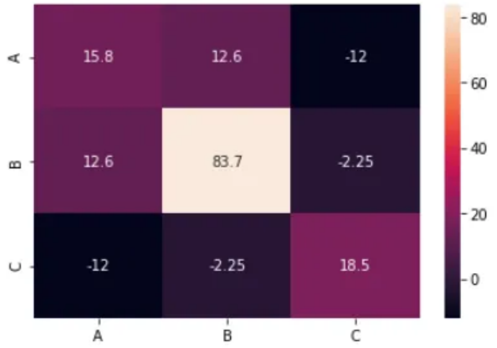
Normal Distribution



Covariance

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

data = {'A': [55,47,52,45,49],
        'B': [48,41,36,38,23],
        'C': [20,25,27,31,22]
        }

df = pd.DataFrame(data,columns=['A','B','C'])

covmtrx = pd.DataFrame.cov(df)
print (covmtrx)
sns.heatmap(covmtrx, annot=True, fmt='g')
plt.show()
```

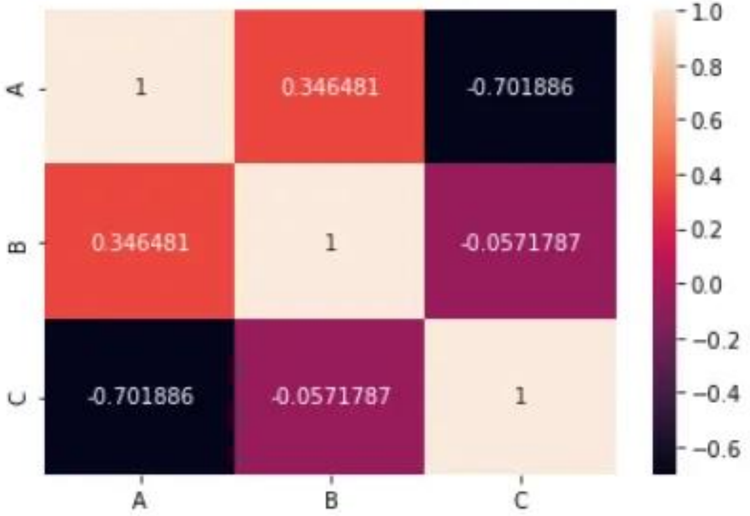
```
      A      B      C
A  15.8  12.60 -12.00
B  12.6  83.70  -2.25
C -12.0  -2.25  18.50
```



Correlation

```
corrmtrx = pd.DataFrame.corr(df)
print (corrmtrx)
sns.heatmap(corrmtrx, annot=True, fmt='g')
plt.show()
```

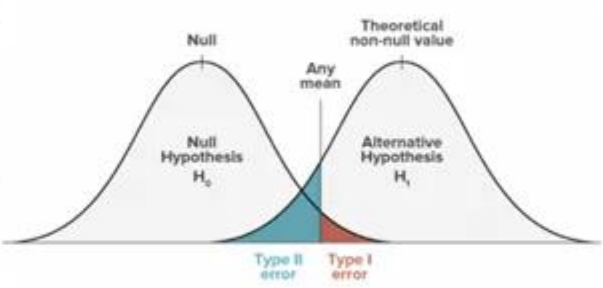
```
          A         B         C
A  1.000000  0.346481 -0.701886
B  0.346481  1.000000 -0.057179
C -0.701886 -0.057179  1.000000
```



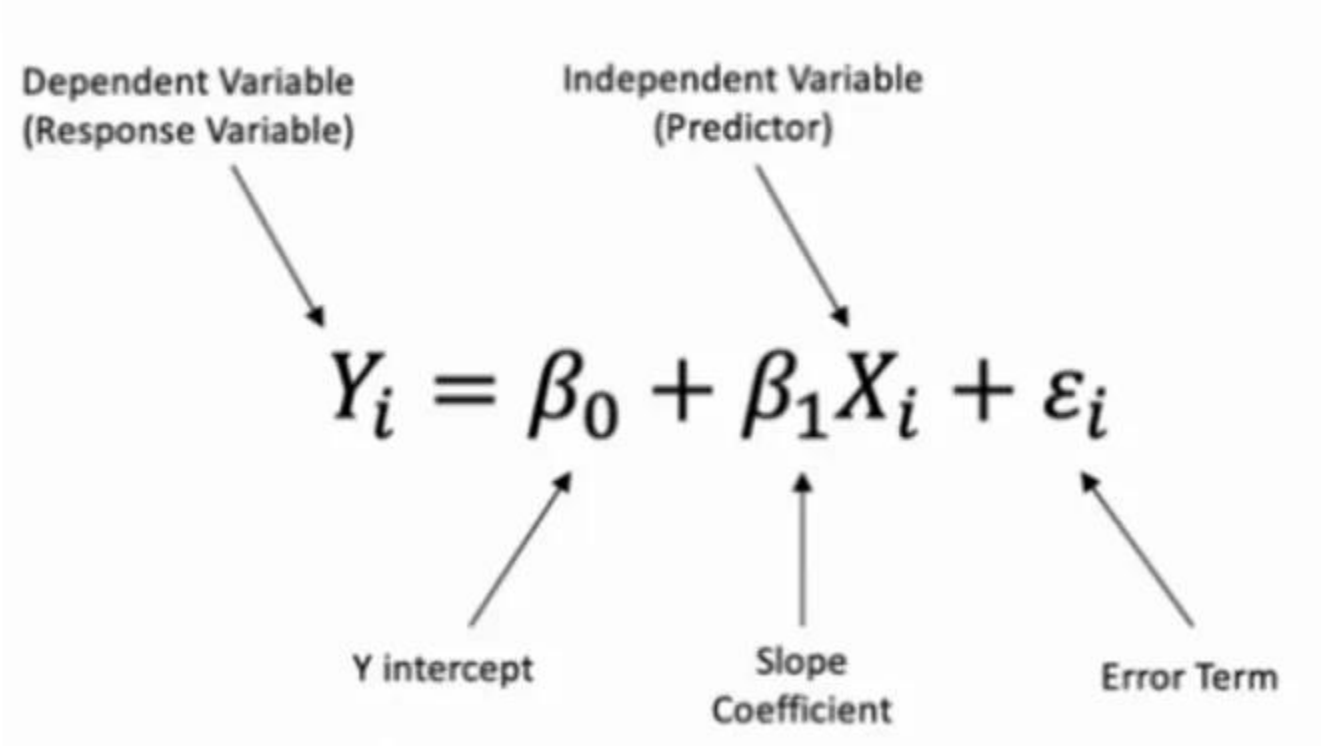
Covariance	Correlation
Measure of how much two random variables vary together	Indicates how strongly two variables are related
Involves the relationship between two variables or datasets	Involves relationship between multiple variables as well
Lies between $-\infty$ and $+\infty$	Lies between -1 and +1
Measure of correlation	Scaled version of covariance
Provides direction of relationship	Provides direction as well as strength of relationship
Dependent on scale of variable	Independent of scale of variable
Has dimensions	Dimensionless

Type 1 & 2

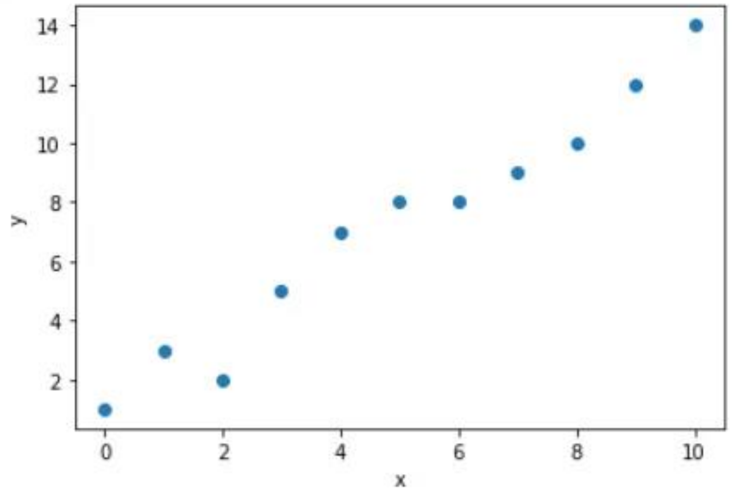
		Conclusion	
		Accept the Null	Reject the Null
The True State of the Nature	H_0 is True	Correct	False Positive Type I Error α
	H_0 is False	False Negative Type II Error β	Correct

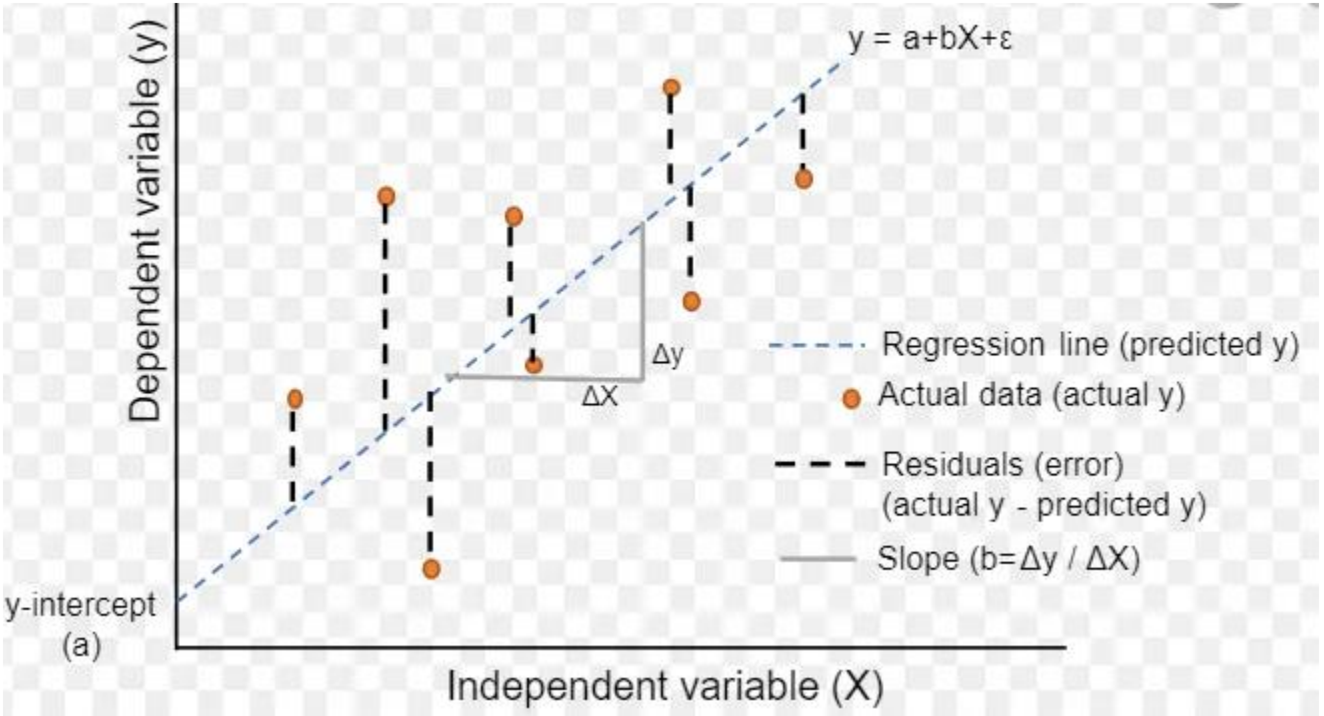


Linear Regression



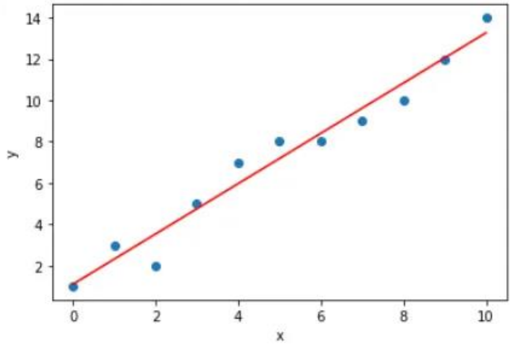
```
import numpy as np
import matplotlib.pyplot as plt
#sample dataset
x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12,14])
#scatter plot of dataset
plt.scatter(x,y)
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```




y = a+bX+ε
Dependent variable (y)
Δy
ΔX
Regression line (predicted y)
Actual data (actual y)
Residuals (error)
(actual y - predicted y)
Slope (b=Δy / ΔX)
y-intercept
(a)
Independent variable (X)

```
n = np.size(x)
mx = np.mean(x)
my = np.mean(y)
# cross-deviation and deviation about x
SSxy = np.sum(y*x) - n*my*mx
SSxx = np.sum(x*x) - n*mx*mx
# estimating regression coefficients
b1 = SSxy/SSxx
b0 = my - b1*mx
print("Estimated coefficients:\nb_0 = {} \
    \nb_1 = {}".format(b0, b1))
plt.scatter(x, y)
y_pred = b0 + b1*x
# plotting the regression line
plt.plot(x, y_pred, color = "r")
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

```
Estimated coefficients:
b_0 = 1.09090909090909
b_1 = 1.2181818181818183
```



Multiple Regression

Dependent Variable
(Response Variable)

Independent Variables
(Predictors)

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \varepsilon$$

Y intercept

Slope
Coefficient

Error Term