Machine Learning Diploma

Session3: Statistics & Probability

AMIT

Agenda

- → Intro to Statistics
- → Data
- → Measurements of data
- → Quartiles
- → Covariance and Correlation
- → What is probability
- → Independent & Dependent Events

1. Intro to Statistics

Intro to Statistics:

➔ Statistics is the mathematical science behind the problem “what can I know about a population if I’m unable to reach every member?”

➔ If we could measure the height of every resident of Australia, then we could make a statement about the average height of Australians at the time we took our measurement, This is where random sampling comes in.

➔ If we take a reasonably sized random sample of Australians and measure their heights, we can form a statistical inference about the population of Australia. Probability helps us know how sure we are of our conclusions!

Intro to Statistics:

- ➔ They are also the tools that provide the foundation for more advanced linear algebra operations and machine learning methods, such as the covariance matrix and principal component analysis respectively.
- ➔ It is important to have a strong grip on fundamental statistics in the context of linear algebra notation.
- ➔ We will introduce some fundamental concepts in statistics and then re-visit them in contest of data analysis.

2. Data

What is Data?

- Data = the collected observations we have about something.
- Data can be continuous: *"What is the stock price?"*
- Or categorical: *"What car has the best resale price?"*

Why Data Matters?

- Helps us understand things as they are: *"What relationships if any exist between two events?"*

  "Do people who eat an apple a day enjoy fewer doctor's visits than those who don't?"

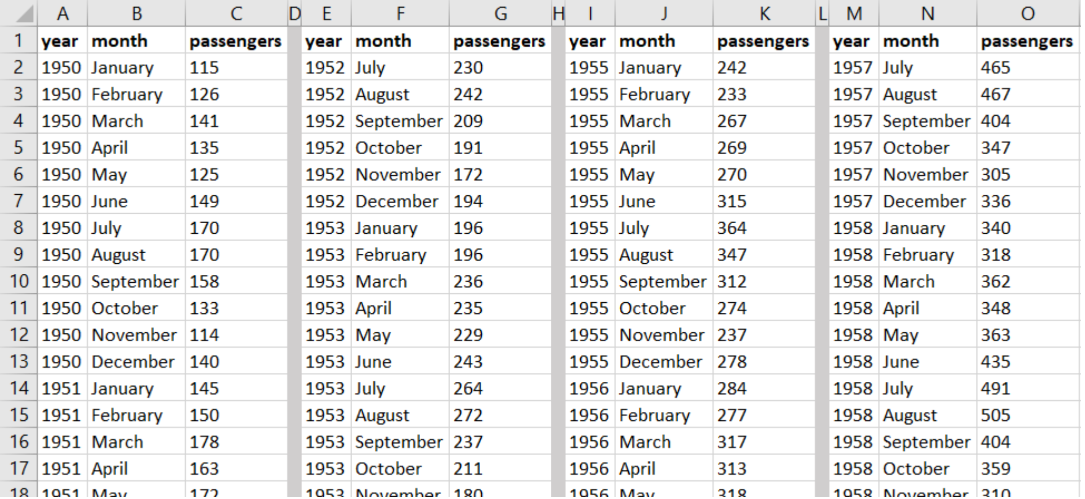
- Helps us predict future behavior to guide business decisions:

  "Based on a user's click history which ad is more likely to bring them to our site?"

Visualizing Data

- Compare a table:

Flights data



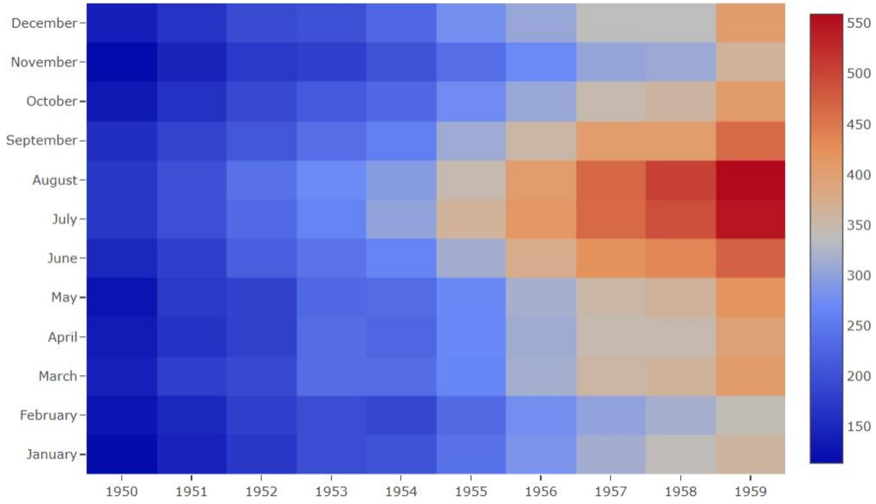
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	**year**	**month**	**passengers**		**year**	**month**	**passengers**		**year**	**month**	**passengers**		**year**	**month**	**passengers**
2	1950	January	115		1952	July	230		1955	January	242		1957	July	465
3	1950	February	126		1952	August	242		1955	February	233		1957	August	467
4	1950	March	141		1952	September	209		1955	March	267		1957	September	404
5	1950	April	135		1952	October	191		1955	April	269		1957	October	347
6	1950	May	125		1952	November	172		1955	May	270		1957	November	305
7	1950	June	149		1952	December	194		1955	June	315		1957	December	336
8	1950	July	170		1953	January	196		1955	July	364		1958	January	340
9	1950	August	170		1953	February	196		1955	August	347		1958	February	318
10	1950	September	158		1953	March	236		1955	September	312		1958	March	362
11	1950	October	133		1953	April	235		1955	October	274		1958	April	348
12	1950	November	114		1953	May	229		1955	November	237		1958	May	363
13	1950	December	140		1953	June	243		1955	December	278		1958	June	435
14	1951	January	145		1953	July	264		1956	January	284		1958	July	491
15	1951	February	150		1953	August	272		1956	February	277		1958	August	505
16	1951	March	178		1953	September	237		1956	March	317		1958	September	404
17	1951	April	163		1953	October	211		1956	April	313		1958	October	359
18	1951	May	172		1953	November	180		1956	May	318		1958	November	310

Not much can be gained by reading it.

Visualizing Data

- to a graph:

Flights data



The graph uncovers two distinct trends - an increase in passengers flying over the years and a greater number of passengers flying in the summer months.

Population vs. Sample

- Population = every member of a group
- Sample = a subset of members that time and resources allow you to measure

3. Measurements of Data

Measurements ofData

- “What was the average return?”

  Measures ofCentral Tendency

- “How far from the average didindividual values stray?”

  Measures ofDispersion

INSTITUTE OF MANAGEMENT PROFESSIONALS

Measures of Central Tendency (mean, median, mode)

- Describe the “location” of the data
- Fail to describe the “shape” of the data

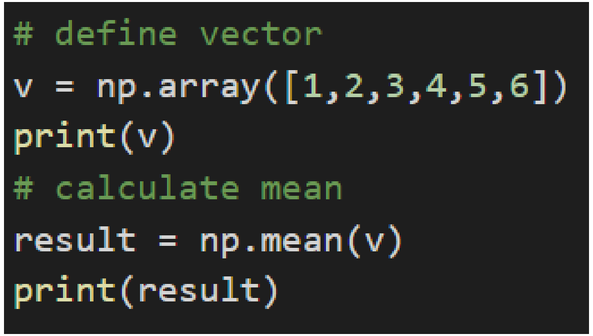
mean = “calculated average”

median = “middle value”

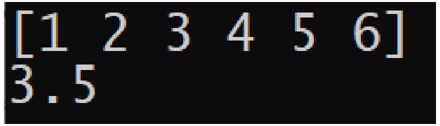
mode = “most occurring value”

Mean for Vectors:

➔ The arithmetic mean can be calculated for a vector or matrix in NumPy by using the mean() function.



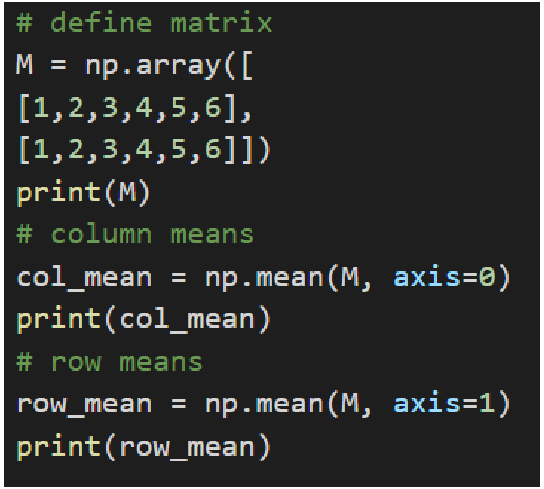
```
# define vector
v = np.array([1,2,3,4,5,6])
print(v)
# calculate mean
result = np.mean(v)
print(result)
```



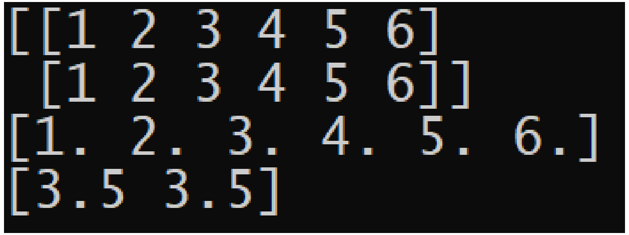
```
[1 2 3 4 5 6]
3.5
```

Mean for Matrices:

- ➔ The mean function can calculate the row or column means of a matrix by specifying the axis argument and the value 0 or 1 respectively.
- ➔ The example below defines a 2 × 6 matrix and calculates both column and row means:



```
# define matrix
M = np.array([
[1,2,3,4,5,6],
[1,2,3,4,5,6]])
print(M)
# column means
col_mean = np.mean(M, axis=0)
print(col_mean)
# row means
row_mean = np.mean(M, axis=1)
print(row_mean)
```



```
[[1 2 3 4 5 6]
 [1 2 3 4 5 6]]
[1. 2. 3. 4. 5. 6.]
[3.5 3.5]
```

Random Variable:

- ➔ A random variable is a variable whose value is unknown or a function that assigns values to each of an experiment's outcomes.

- ➔ Random variables are often designated by letters and can be classified as discrete, which are variables that have specific values, or continuous, which are variables that can have any values within a continuous range.

- ➔ In probability and statistics, random variables are used to quantify outcomes of a random occurrence.

Random Variable:

➔ A typical example of a random variable is the outcome of a coin toss.

➔ If random variable, Y, is the number of heads we get from tossing two coins, then Y could be 0, 1, or 2. This means that we could have no heads, one head, or both heads on a two-coin toss.

Expected Value and Mean:

➔ In probability, the average value of some random variable X is called the expected value or the expectation. $E[X]$

➔ It is calculated as the probability weighted sum of values that can be drawn.

$$E[X] = \sum x_1 \times p_1, x_2 \times p_2, x_3 \times p_3, \cdots, x_n \times p_n$$

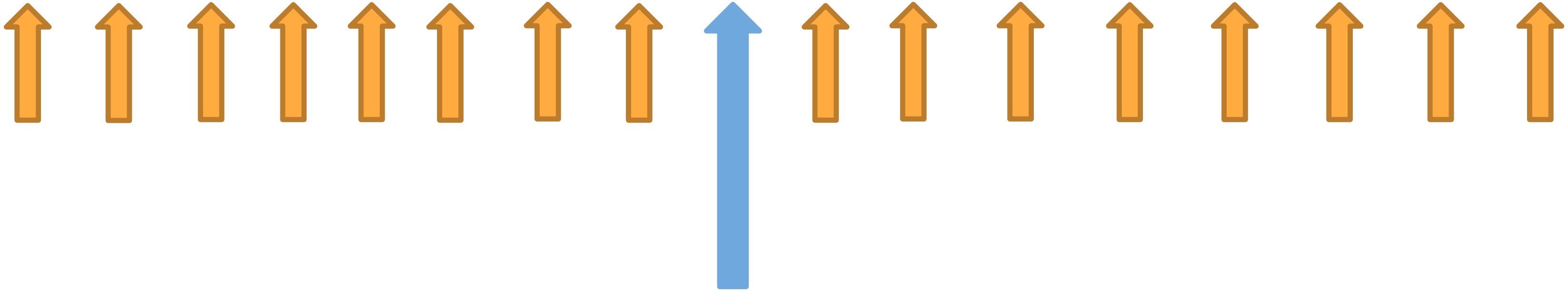
➔ Or in case that all probabilities are equal:

$$E[X] = \frac{1}{n} \times \sum x_1, x_2, x_3, \cdots, x_n$$

Where n is the number of values.

Median – *odd number of values*

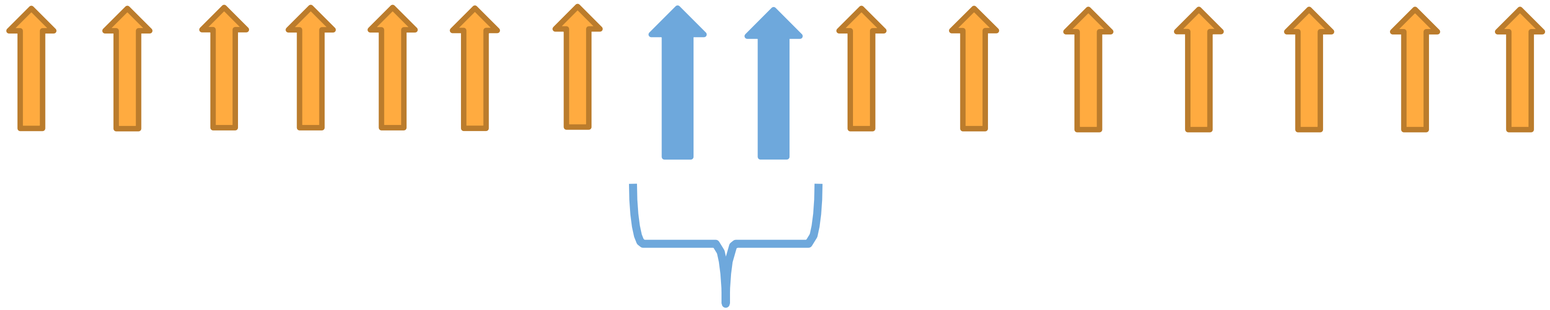
9 10 10 11 13 15 16 19 19 21 23 28 30 33 34 36 44



= 19

Median - *even number of values*

10 10 11 13 15 16 19 19 21 23 28 30 33 34 36 44



$$\frac{19 + 21}{2} = 20$$

Mean vs. Median

- The mean can be influenced by *outliers*.
- The mean of {2,3,2,3,2,12} is 4
- The median is 2.5
- The median is much closer to most of the values in the series!

Mode

10 10 1113 15 16 16 16 21 23 28 30 33 34 36 44

= 16

Measures of Dispersion [Variance]:

➔ In probability, the variance of some random variable X is a measure of how much values in the distribution vary on average with respect to the mean. $Var[X]$

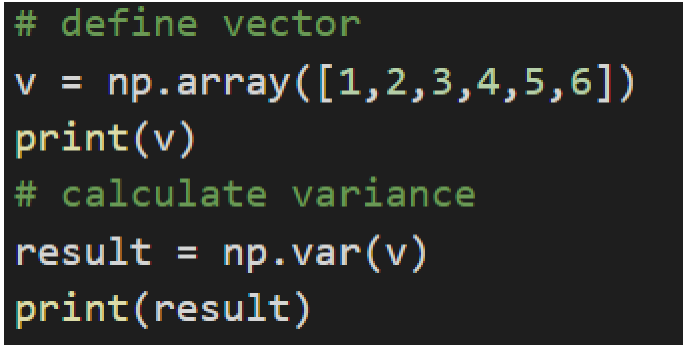
➔ Variance is calculated as the average squared difference of each value in the distribution from the expected value. Or the expected squared difference from the expected value.

$$Var[X] = E[(X - E[X])^2]$$

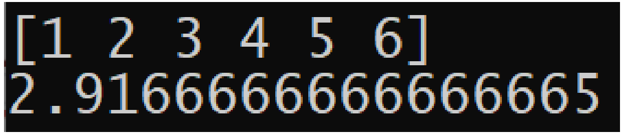
➔ “The average of the squared differences from the Mean.”

Variance for Vectors:

➔ By default, the var() function calculates the population variance (The whole scope of observation).



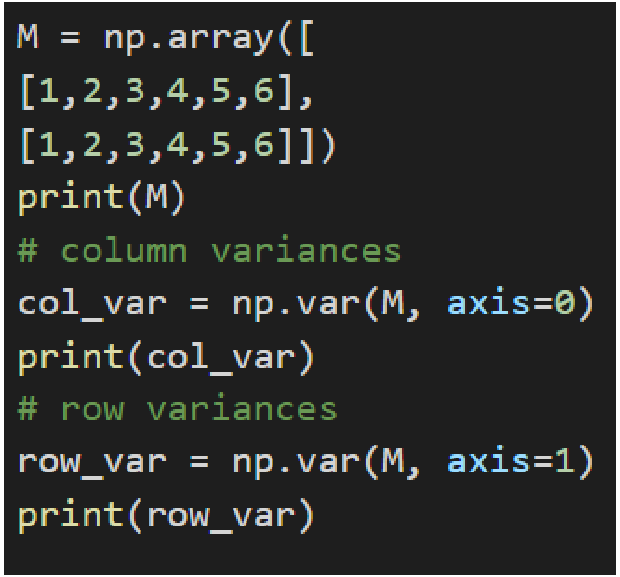
```
# define vector
v = np.array([1,2,3,4,5,6])
print(v)
# calculate variance
result = np.var(v)
print(result)
```



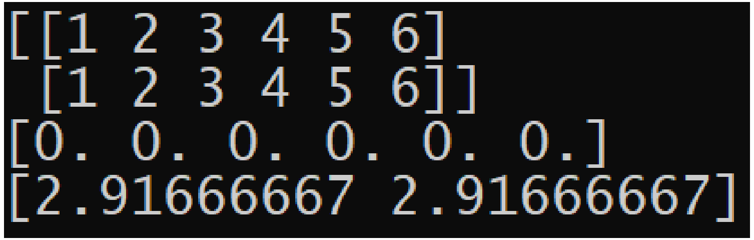
```
[1 2 3 4 5 6]
2.9166666666666665
```

Variance for Matrices:

➔ By default, the var() function calculates the population variance (The whole scope of observation).



```
M = np.array([
[1,2,3,4,5,6],
[1,2,3,4,5,6]])
print(M)
# column variances
col_var = np.var(M, axis=0)
print(col_var)
# row variances
row_var = np.var(M, axis=1)
print(row_var)
```



```
[[1 2 3 4 5 6]
 [1 2 3 4 5 6]]
[0. 0. 0. 0. 0. 0.]
[2.91666667 2.91666667]
```

Standard Deviation:

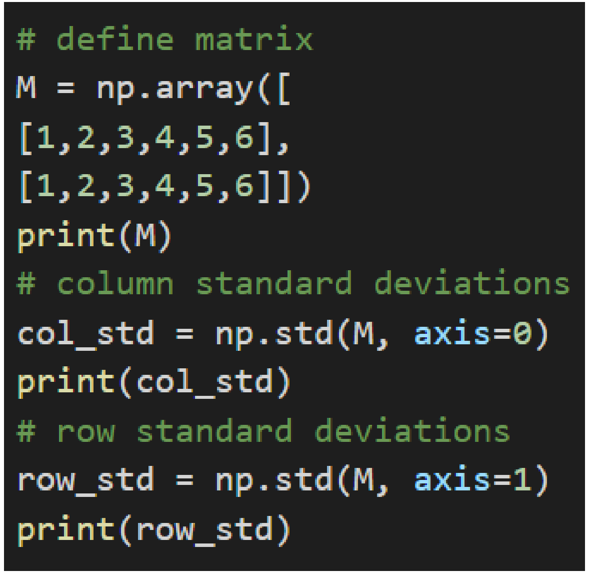
- ➔ The standard deviation is calculated as the square root of the variance and is denoted as lowercase s

$$s = \sqrt{\sigma^2}$$

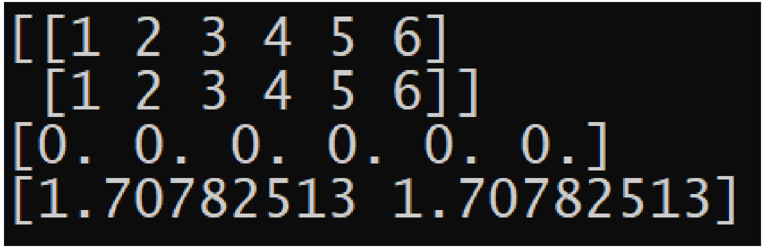
- ➔ Using the Standard Deviation we have a "standard" way of knowing what is normal among the values.

Standard Deviation:

➔ NumPy also provides a function for calculating the standard deviation directly via the std() function



```
# define matrix
M = np.array([
[1,2,3,4,5,6],
[1,2,3,4,5,6]])
print(M)
# column standard deviations
col_std = np.std(M, axis=0)
print(col_std)
# row standard deviations
row_std = np.std(M, axis=1)
print(row_std)
```



```
[[1 2 3 4 5 6]
 [1 2 3 4 5 6]]
[0. 0. 0. 0. 0. 0.]
[1.70782513 1.70782513]
```

Example:

➔ You and your friends have just measured the heights of your dogs:
The heights (at the shoulders) are: 600mm, 470mm, 170mm, 430mm and 300mm.

➔ Mean = (600 + 470 + 170 + 430 + 300) /5
= 1970 / 5 = 394

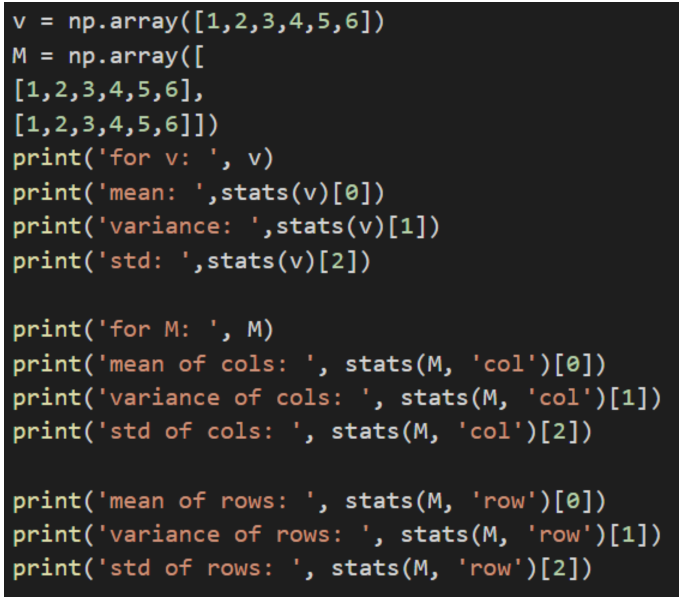
➔ Variance = $((600-394)^2+(470-394)^2+(170-394)^2+(430-394)^2+(300-394)^2)/ 5$
= 21704

➔ Std = $\sqrt{21704}$ = 147.32… = 147 mm

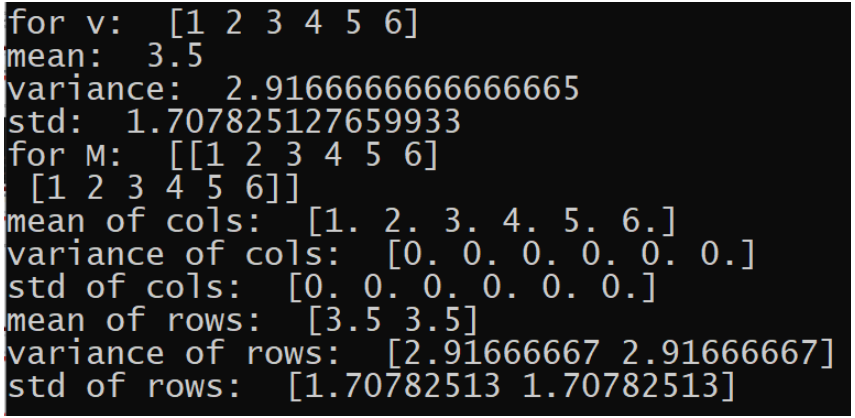
➔ Now we can say that the average height of dogs is 394 mm and any dog within 247 and 541 has normal height.

Quiz:

➔ Write a function stats() that takes any sequence of numbers and give back its mean, variance and std. Look at the example input and example output:



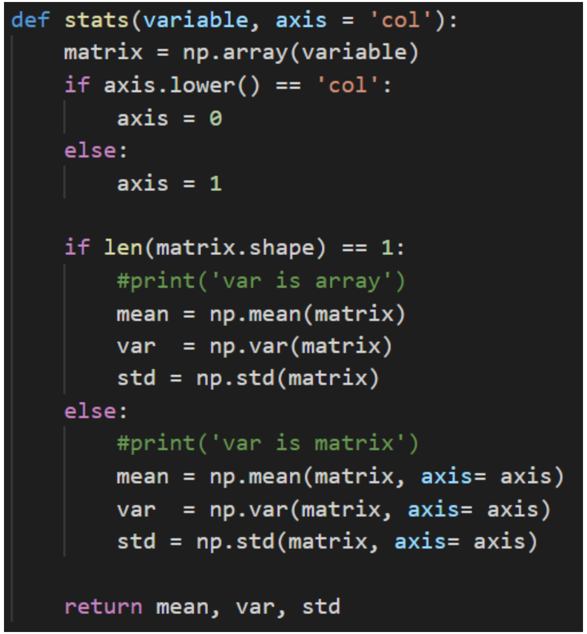
```
v = np.array([1,2,3,4,5,6])
M = np.array([
[1,2,3,4,5,6],
[1,2,3,4,5,6]])
print('for v: ', v)
print('mean: ',stats(v)[0])
print('variance: ',stats(v)[1])
print('std: ',stats(v)[2])

print('for M: ', M)
print('mean of cols: ', stats(M, 'col')[0])
print('variance of cols: ', stats(M, 'col')[1])
print('std of cols: ', stats(M, 'col')[2])

print('mean of rows: ', stats(M, 'row')[0])
print('variance of rows: ', stats(M, 'row')[1])
print('std of rows: ', stats(M, 'row')[2])
```



```
for v:   [1 2 3 4 5 6]
mean:   3.5
variance:   2.9166666666666665
std:   1.707825127659933
for M:   [[1 2 3 4 5 6]
 [1 2 3 4 5 6]]
mean of cols:   [1. 2. 3. 4. 5. 6.]
variance of cols:   [0. 0. 0. 0. 0. 0.]
std of cols:   [0. 0. 0. 0. 0. 0.]
mean of rows:   [3.5 3.5]
variance of rows:   [2.91666667 2.91666667]
std of rows:   [1.70782513 1.70782513]
```

Quiz (Solution):

➔ Write a function stats() that takes any sequence of numbers and give back its mean, variance and std.



```python
def stats(variable, axis = 'col'):
    matrix = np.array(variable)
    if axis.lower() == 'col':
        axis = 0
    else:
        axis = 1

    if len(matrix.shape) == 1:
        #print('var is array')
        mean = np.mean(matrix)
        var  = np.var(matrix)
        std = np.std(matrix)
    else:
        #print('var is matrix')
        mean = np.mean(matrix, axis= axis)
        var  = np.var(matrix, axis= axis)
        std = np.std(matrix, axis= axis)

    return mean, var, std
```

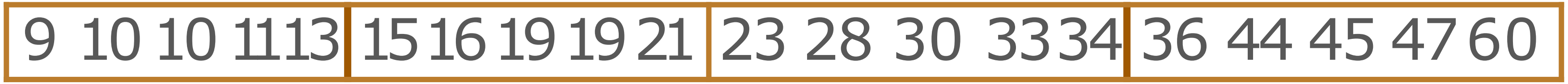
4.Quartiles

Quartiles and IQR

- Another way to describe data is through quartiles and the interquartile range (IQR)
- Has the advantage that every data point is considered, not aggregated!

Quartiles andIQR

- Consider the following series of 20 values:



9 10 10 1113 | 15 16 19 19 21 | 23 28 30 3334 | 36 44 45 47 60

1st quartile

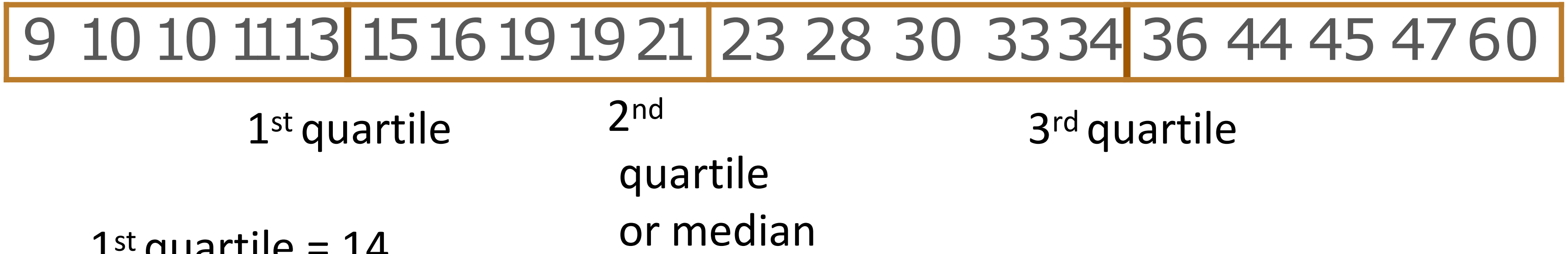
2nd quartile or median

3rd quartile

1. Divide the series
2. Divide each subseries
3. These become quartiles

Quartiles and IQR

- Consider the following series of 20 values:



9 10 10 11 13	15 16 19 19 21	23 28 30 33 34	36 44 45 47 60

1st quartile

2nd quartile or median

3rd quartile

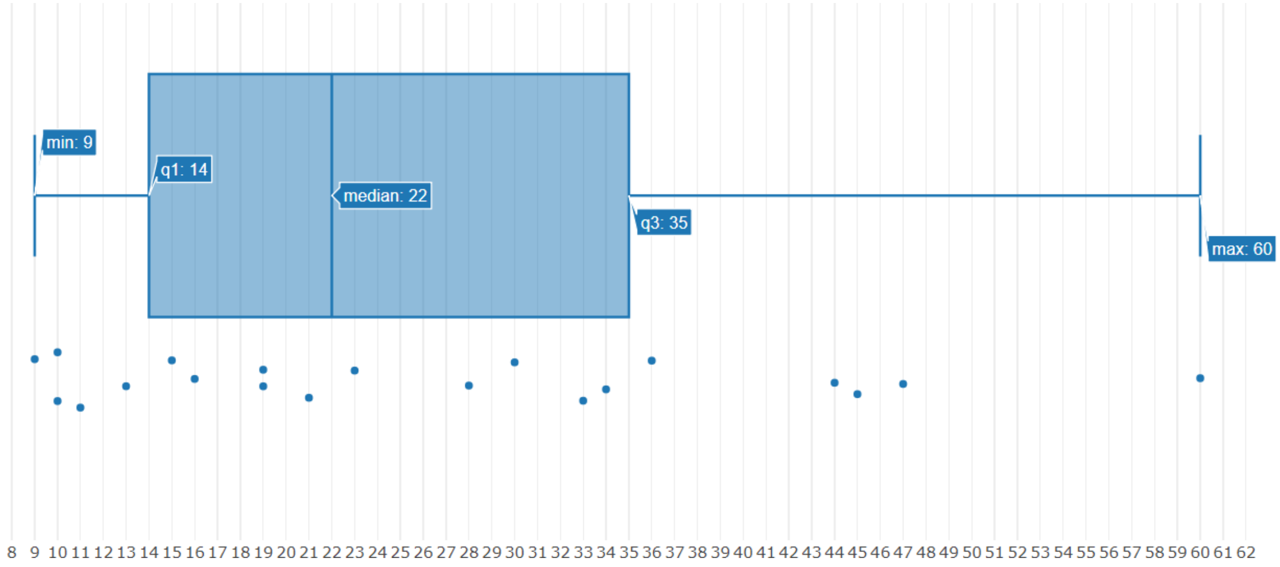
1st quartile = 14

2nd quartile = 22

3rd quartile = 35

Plot the Quartiles

9 10 10 11 13	15 16 19 19 21	23 28 30 33 34	36 44 45 47 60

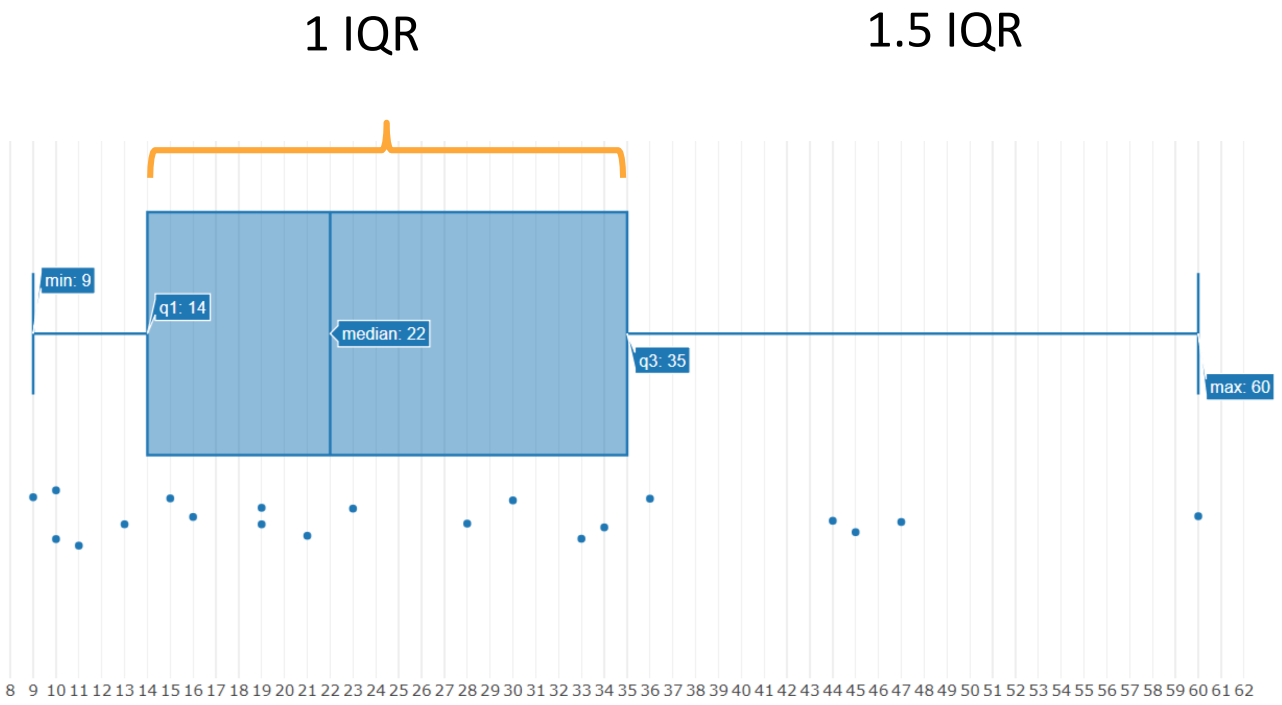


Quartile ranges are seldom the same size!

Fences & Outliers

- What is considered an “outlier”?
- A common practice is to set a “fence” that is 1.5 times the width of the IQR
- Anything outside the fence is an outlier
- This is determined by the *data*, not an arbitrary percentage!

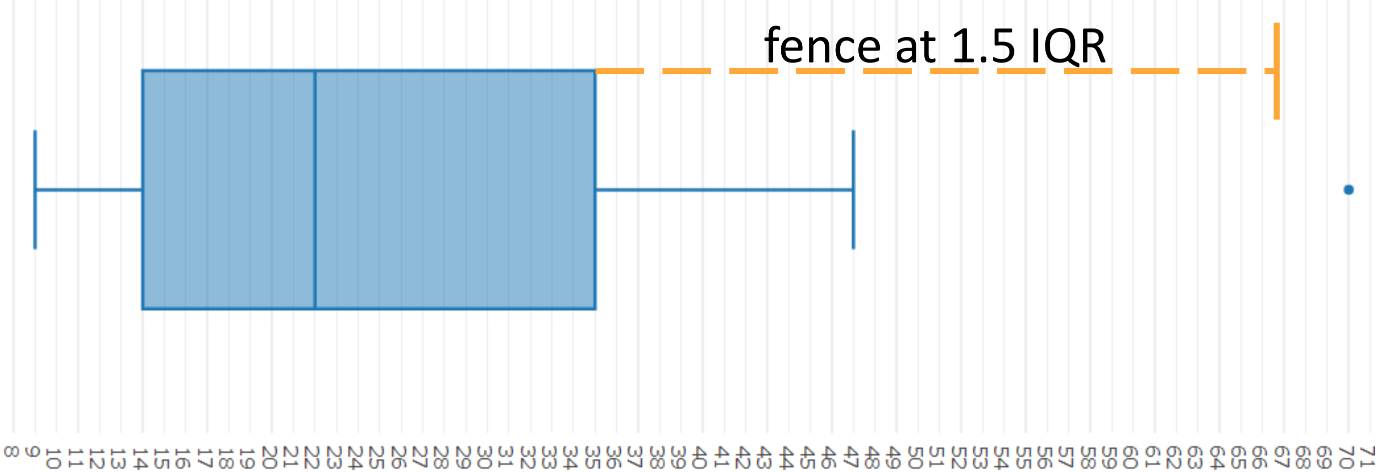
Fences & Outliers



In this set, 60 is *not* an outlier, but 70 would be

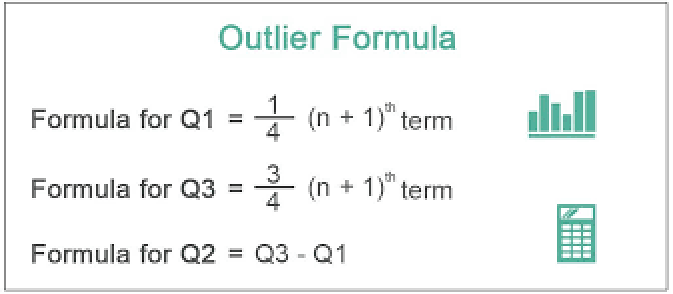
Fences & Outliers

9 10 10 11 13	15 16 19 19 21	23 28 30 33 34	36 44 45 47 70


fence at 1.5 IQR

- When drawing box plots, the whiskers are brought inward to the outermost values inside the fence.

Outlier Formula



Outlier Formula

Formula for Q1 = $\frac{1}{4}$ $(n + 1)^{th}$ term

Formula for Q3 = $\frac{3}{4}$ $(n + 1)^{th}$ term

Formula for Q2 = Q3 - Q1

Outliers Formula

Lower Outlier = Q1 – (1.5 x IQR)

Higher Outlier = Q3 + (1.5 x IQR)

5. Covariance and Correlation

Covariance:

➔ Covariance is a quantitative measure of the degree to which the deviation of one variable (X) from its mean is related to the deviation of another variable (Y) from its mean.

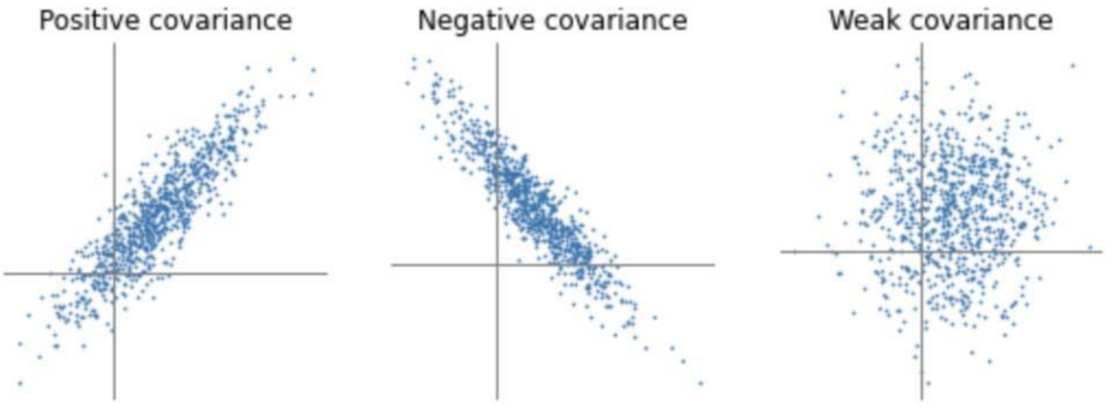
➔ In probability, covariance is the measure of the joint probability for two random variables. It describes how the two variables change together. It is denoted as the function cov(X, Y), where X and Y are the two random variables being considered.

➔ Covariance is calculated as expected value or average of the product of the differences of each random variable from their expected values, where E[X] is the expected value for X and E[Y] is the expected value of y.

$$cov(X,Y) = \frac{1}{n} \times \sum (x - E[X]) \times (y - E[Y])$$

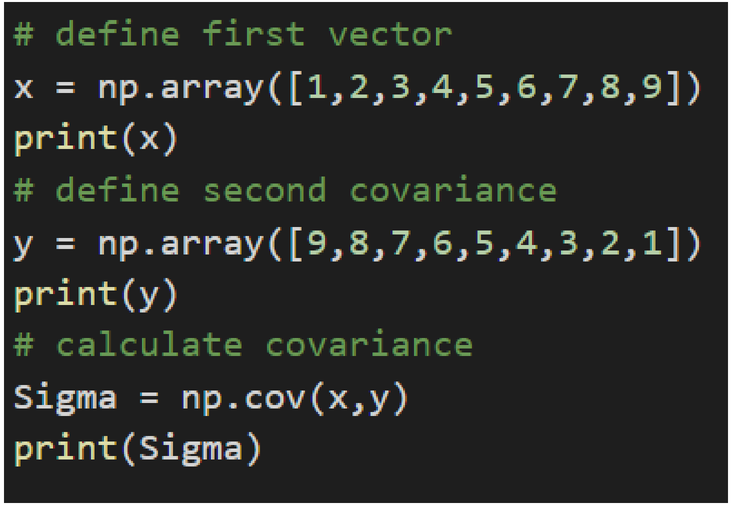
Covariance:

- ➔ The sign of the covariance can be interpreted as whether the two variables increase together (positive) or decrease together (negative).
- ➔ A covariance value of zero indicates that both variables are completely independent.

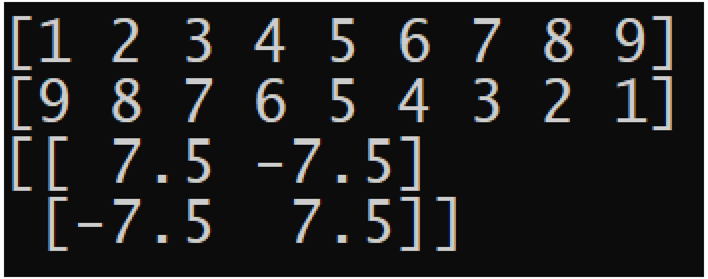


Covariance:

➔ The example below defines two vectors of equal length with one increasing and one decreasing. We would expect the covariance between these variables to be negative.



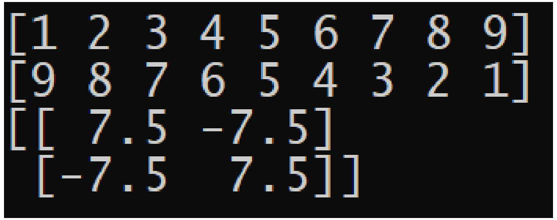
```
# define first vector
x = np.array([1,2,3,4,5,6,7,8,9])
print(x)
# define second covariance
y = np.array([9,8,7,6,5,4,3,2,1])
print(y)
# calculate covariance
Sigma = np.cov(x,y)
print(Sigma)
```



```
[1 2 3 4 5 6 7 8 9]
[9 8 7 6 5 4 3 2 1]
[[ 7.5 -7.5]
 [-7.5  7.5]]
```

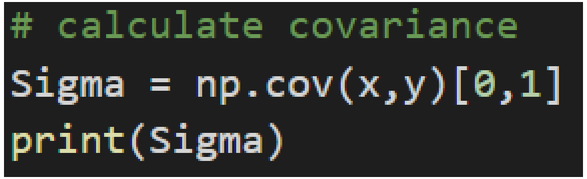
Covariance:

➔ The output is the covariance matrix. For example the first row and first column is X with X - with itself (7.5) which is the variance of this variable.

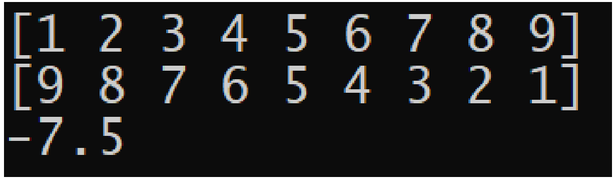


```
[1 2 3 4 5 6 7 8 9]
[9 8 7 6 5 4 3 2 1]
[[ 7.5 -7.5]
 [-7.5  7.5]]
```

➔ To find the covariance between two arrays. Access it by [0,1]



```
# calculate covariance
Sigma = np.cov(x,y)[0,1]
print(Sigma)
```



```
[1 2 3 4 5 6 7 8 9]
[9 8 7 6 5 4 3 2 1]
-7.5
```

Correlation:

➔ We’ve established that covariance indicates the extent to which two random variables increase or decrease in tandem with each other. Correlation tells us both the strength and the direction of this relationship.

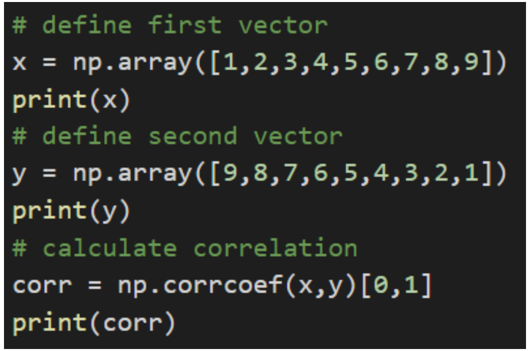
➔ The covariance can be normalized to a score between -1 and 1 to make the magnitude interpretable by dividing it by the standard deviation of X and Y.

➔ The result is called the correlation of the variables, also called the Pearson correlation coefficient, named for the developer of the method.

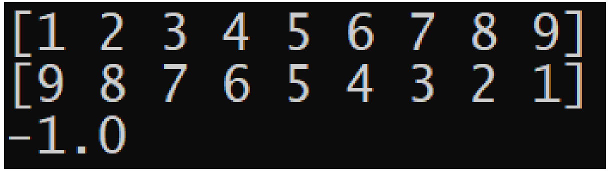
$$r = \frac{cov(X,Y)}{s_X \times s_Y}$$

Correlation:

- ➔ NumPy provides the corrcoef() function for calculating the correlation between two variables directly.
- ➔ Like cov(), it returns a matrix, in this case a correlation matrix.
- ➔ As with the results from cov() we can access just the correlation of interest from the [0,1] value from the returned squared matrix.
- ➔ We can see that the vectors are maximally negatively correlated.



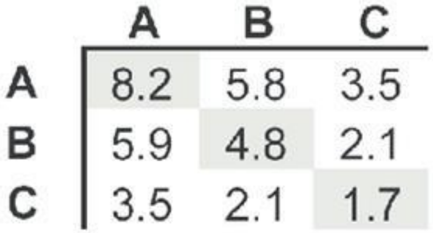
```python
# define first vector
x = np.array([1,2,3,4,5,6,7,8,9])
print(x)
# define second vector
y = np.array([9,8,7,6,5,4,3,2,1])
print(y)
# calculate correlation
corr = np.corrcoef(x,y)[0,1]
print(corr)
```



```
[1 2 3 4 5 6 7 8 9]
[9 8 7 6 5 4 3 2 1]
-1.0
```

Covariance Matrix:

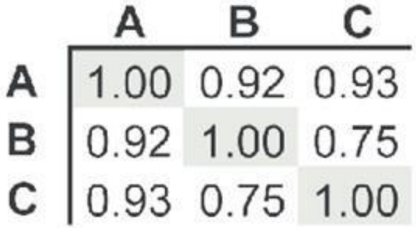
- ➔ The covariance matrix is a square and symmetric matrix that describes the covariance between two or more random variables.
- ➔ The diagonal of the covariance matrix are the variances of each of the random variables, as such it is often called the variance-covariance matrix.



	A	B	C
A	8.2	5.8	3.5
B	5.9	4.8	2.1
C	3.5	2.1	1.7

Correlation Matrix:

➔ A correlation matrix is simply a table which displays the correlation coefficients for different variables. The matrix depicts the correlation between all the possible pairs of values in a table.



	A	B	C
A	1.00	0.92	0.93
B	0.92	1.00	0.75
C	0.93	0.75	1.00

6. What is probability

What is Probability?

- Probability is a value between 0 and1 that a certain event will occur
- For example, the probability that a fair coin will come up heads is 0.5
- Mathematically we write:

$$P(E_{heads}) = 0.5$$

What is Probability?

- In the above "heads" example, the act of flipping a coin is called a trial.
- Over very many trials, a fair coin should come up "heads" half of the time.

Trials Have No Memory!

- If a fair coin comes up tails 5 times in a row, the chance it will come up heads is *still* 0.5
- You can’t think of a series of independent events as needing to “catch up” to the expected probability.
- Each trial is independent of all others.

Experiments and Sample Space

- Each trial of flipping a coin can be called an experiment
- Each mutually exclusive outcome is called a simple event
- The sample space is the sum of every possible simple event

Experiments and Sample Space

- Consider rolling a six-sided die
- One roll is an experiment
- The simple events are:

$E_1 = 1$ $E_2 = 2$ $E_3 = 3$

$E_4 = 4$ $E_5 = 5$ $E_6 = 6$

- Therefore, the sample space is:

$$S = \{E_1 , E_2 , E_3 , E_4, E_5, E_6\}$$

Experiments and Sample Space

- The probability that a fair die will roll a six: The simple event is:

  E_6=6 (one event)

  Total sample space:

  S = {E_1 , E_2 , E_3 , E_4, E_5, E_6} (six possible outcomes)

  The probability:

  P(Roll Six) = 1/6

7. Independent & Dependent Events

Independent Events

- An independent series of events occur when the outcome of one event has no effect on the outcome of another.
- An example is flipping a fair coin twice, The chance of getting heads on the second toss is independent of the result of the first toss.

1st Toss	2nd Toss
H	H
H	T
T	H
T	T

Dependent Events

- A dependent event occurs when the outcome of a first event does affect the probability of a second event.
- A common example is to draw colored marbles from a bag *without replacement*.

Any Questions?

THANK YOU!

AMIT