




correlation analysis

Data Science



What is correlation analysis?

- ❑ **Correlation means an association**, It is a measure of the extent to which two variables are related.
 - ❑ **Correlation analysis is a statistical method used to measure the strength of the linear relationship between two variables and compute their association.**
 - ❑ **Simply put - correlation analysis calculates the level of change in one variable due to the change in the other.**
 - ❑ A high correlation means a strong relationship between the two variables, while a low correlation means that the variables are weakly related.
- 



Cont....


- use correlation analysis to analyze quantitative data collected through research methods like surveys and live polls.
- They try to identify the relationship, patterns, significant connections, and trends between two variables or datasets.
- There is a positive correlation between two variables when an increase in one variable leads to the increase in the other.
- On the other hand, a negative correlation means that when one variable increases, the other decreases and vice-versa.

How does the `corr` method work in Python?

- The `corr` method computes the correlation coefficient between every pair of numerically-valued columns in a DataFrame. The Pearson correlation coefficient measures the linear relationship between two variables, ranging from -1 to +1, where:
 - +1 indicates a perfect positive linear relationship,
 - -1 indicates a perfect negative linear relationship,
 - 0 indicates no linear relationship.

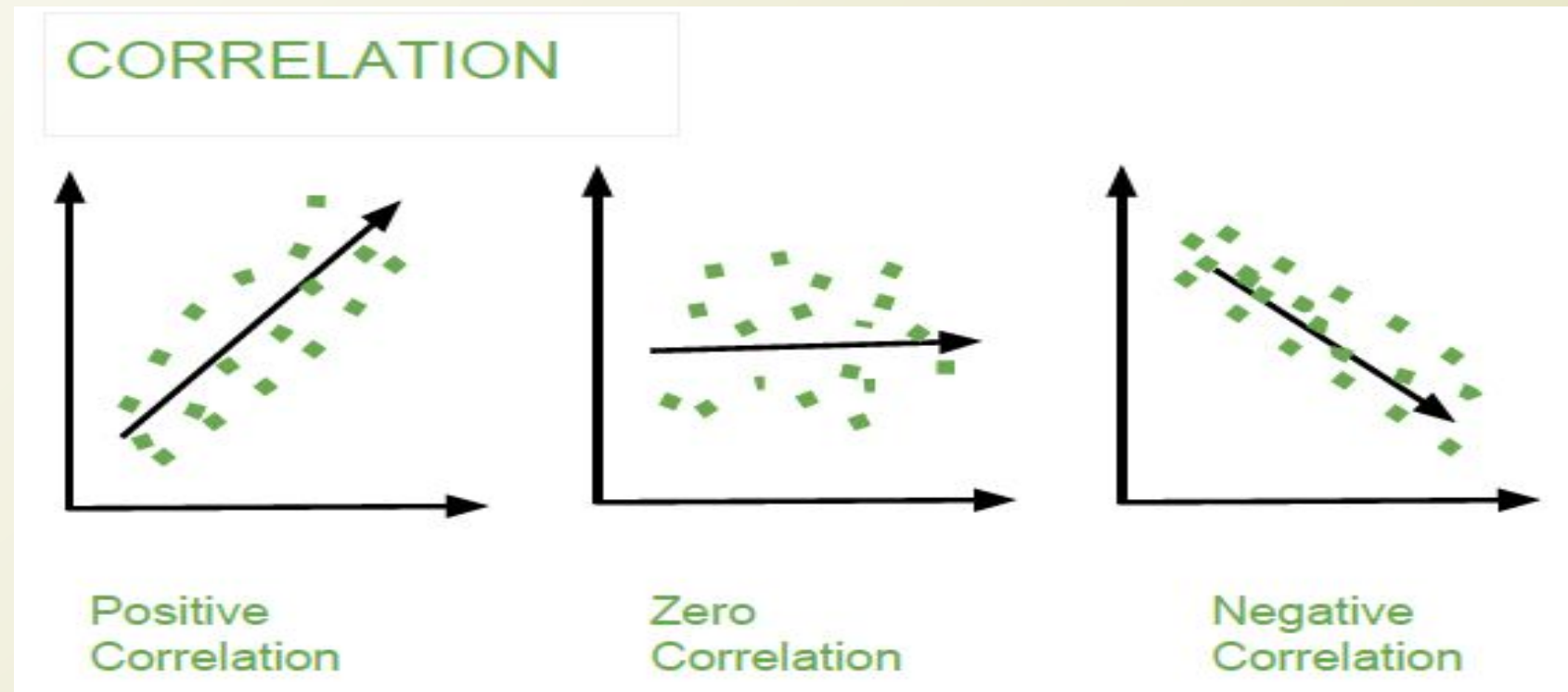


Correlation Coefficient

- The correlation coefficient is the unit of measurement used to calculate the intensity in the linear relationship between the variables involved in a correlation analysis.
 - it is represented with the symbol r and is usually a value without units which is located between 1 and -1.
- 

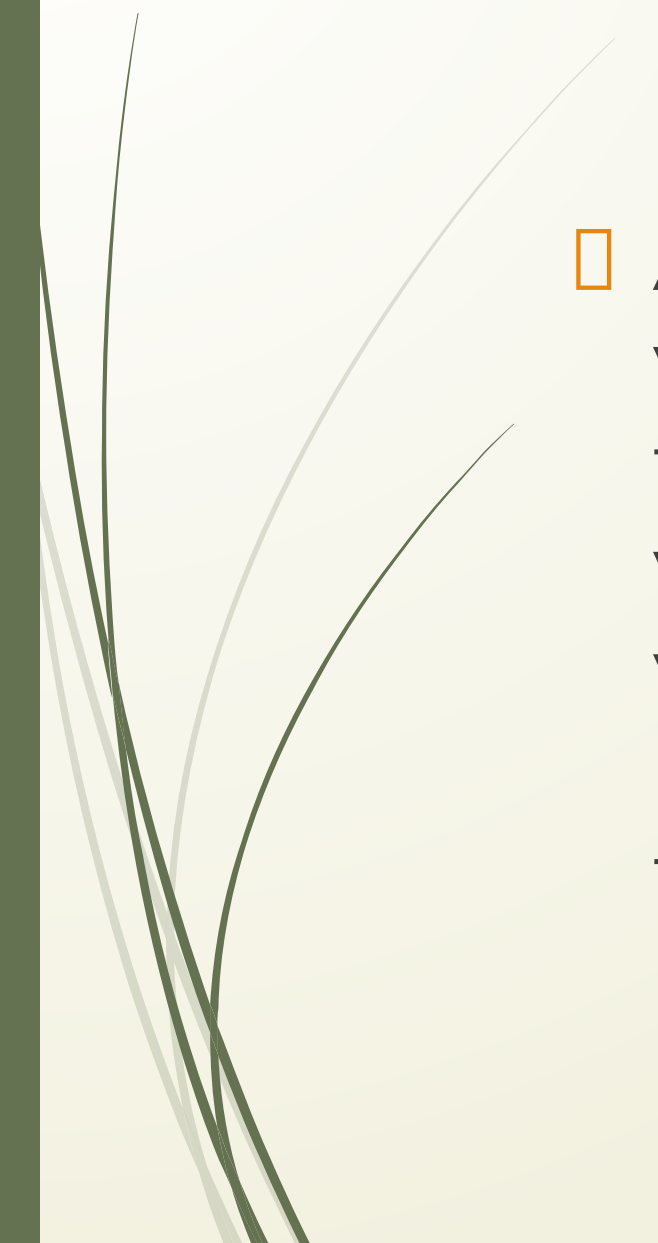
types of correlation:

- There are different types of correlation:
- Positive Correlation:
- Neutral Correlation(**Weak/Zero correlation**)
- Negative Correlation



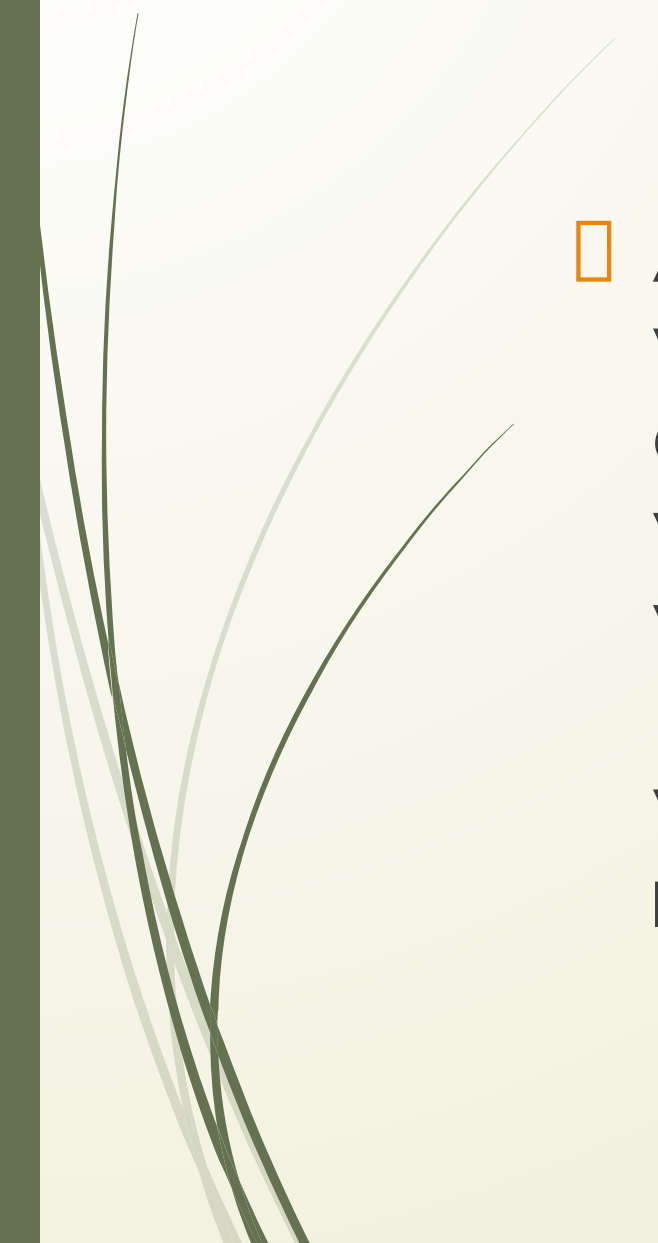


Positive correlation:

- A positive correlation between two variables means both the variables move in the same direction. An increase in one variable leads to an increase in the other variable and vice versa.
For example, spending more time on a treadmill burns more calories.
- 




Negative correlation:

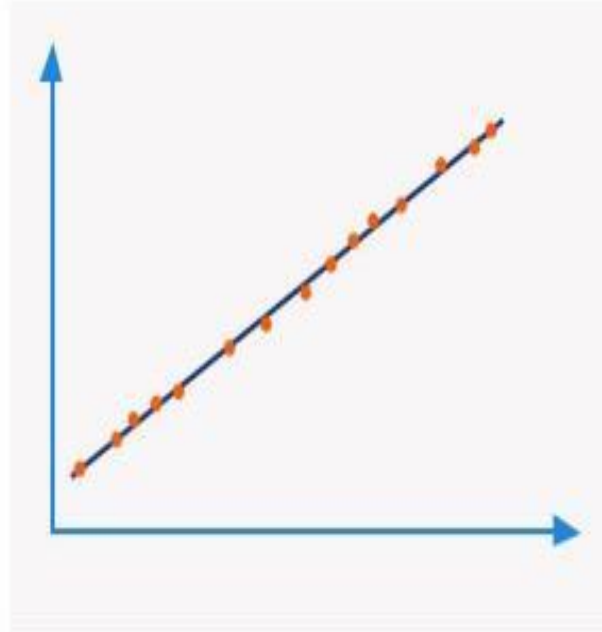
- A negative correlation between two variables means that the variables move in opposite directions. An increase in one variable leads to a decrease in the other variable and vice versa.
For example, increasing the speed of a vehicle decreases the time you take to reach your destination.
- 



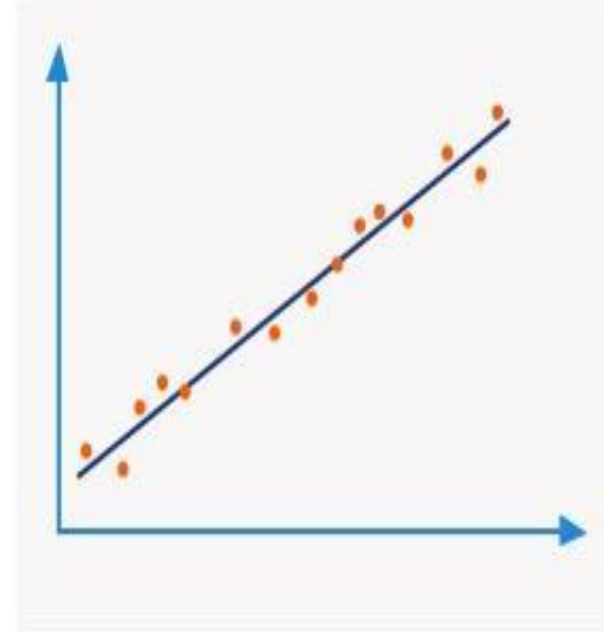
Weak/Zero correlation :

- No correlation exists when one variable does not affect the other. For example, there is no correlation between the number of years of school a person has attended and the letters in his/her name.
- 

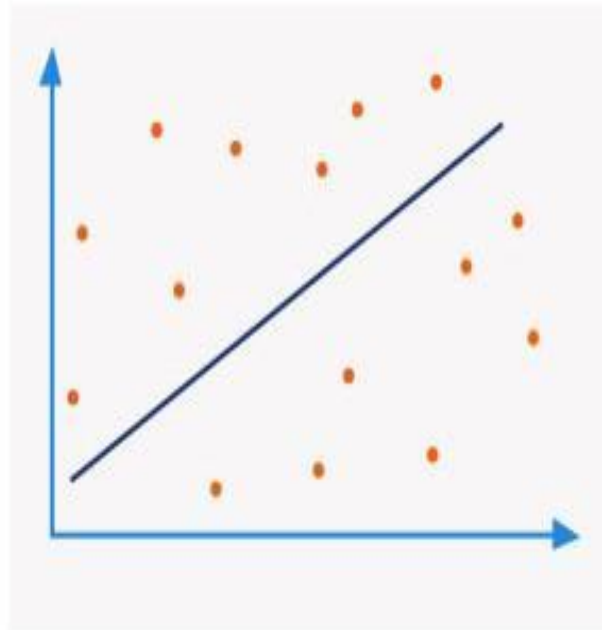
1.
Large positive
correlation



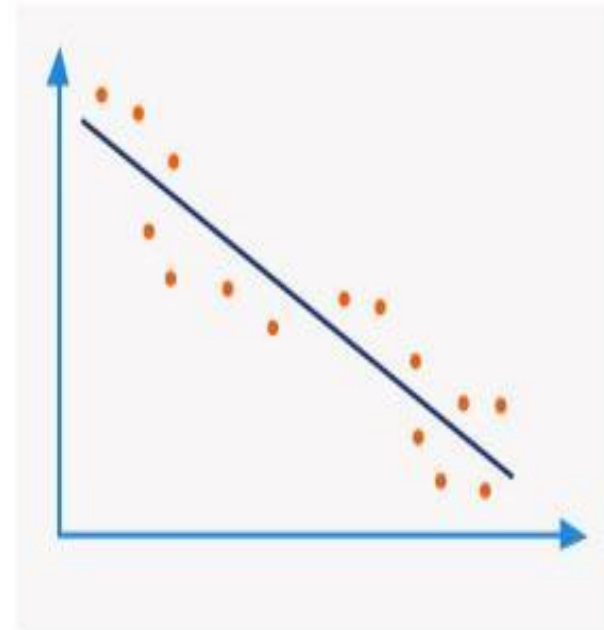
2.
Medium positive
correlation



4.
Weak / no
correlation




3.
Small negative
correlation





Advantages of correlation analysis

- ❑ Some of the most notorious benefits of correlation analysis are:
 - ❑ Awareness of the behavior between two variables: A correlation helps to identify the absence or presence of a relationship between two variables. It tends to be more relevant to everyday life.
 - ❑ A good starting point for research: It proves to be a good starting point when a researcher starts investigating relationships for the first time.
 - ❑ Uses for further studies: Researchers can identify the direction and strength of the relationship between two variables and later narrow the findings down in later studies.
 - ❑ Simple metrics: Research findings are simple to classify. The findings can range from -1.00 to 1.00. There can be only three potential broad outcomes of the analysis.
- 



Pearson correlation coefficient:

- Pearson's correlation coefficient calculates the effect of change in one variable when the other variable changes.
- The Pearson coefficient correlation has a high statistical significance.
- It looks at the relationship between two variables.
- It seeks to draw a line through the data of two variables to show their relationship.
- The relationship of the variables is measured with the help Pearson correlation coefficient calculator.
- This linear relationship can be positive or negative.

Pearson correlation coefficient formula

- The correlation coefficient formula finds out the relation between the variables. It returns the values between -1 and 1. Use the below Pearson coefficient correlation calculator to measure the strength of two variables.
- Where:
- N = the number of pairs of scores
- Σxy = the sum of the products of paired scores
- Σx = the sum of x scores
- Σy = the sum of y scores
- Σx^2 = the sum of squared x scores
- Σy^2 = the sum of squared y scores

$$r = \frac{N\Sigma xy - (\Sigma x)(\Sigma y)}{\sqrt{[N\Sigma x^2 - (\Sigma x)^2][N\Sigma y^2 - (\Sigma y)^2]}}$$

Calculating correlation in Python

- There are various Python packages that can help us measure correlation we will focus on the correlation functions available in three well-known packages: SciPy, NumPy, and pandas.
- Example:
- imagine we want to study the relationship between work experience (measured in years) and salary (measured in dollars) in a small yet successful startup comprising 16 workers.
- `experience = [1, 3, 4, 5, 5, 6, 7, 10, 11, 12, 15, 20, 25, 28, 30, 35]`
- `salary = [20000, 30000, 40000, 45000, 55000, 60000, 80000, 100000, 130000, 150000, 200000, 230000, 250000, 300000, 350000, 400000]`



SciPy

- ❑ SciPy is a great library to perform statistical operations. The `scipy.stats` module includes the `pearsonr(x, y)` function to calculate Pearson's correlation coefficient between two data samples.
- ❑ `import scipy.stats as stats`
- ❑ `corr, _ = stats.pearsonr (experience, salary)`
- ❑ `corr`
- ❑ `0.9929845761480398`

Cont....

- We could calculate Spearman's and Kendall's coefficient in the same fashion:
- `spearman_corr, _ = stats.spearmanr(experience, salary)`
- `spearman_corr`
- 0.9992644353546791

- `kendall_corr, _ = stats.kendalltau(experience, salary)`
- `kendall_corr`
- 0.9958246164193105

NumPy

- NumPy is a popular package that offers an extensive collection of advanced mathematical functions, including `np.corrcoef()` that returns a matrix of Pearson's correlation coefficients:
- `import numpy as np`
- `np.corrcoef(experience, salary)`
- `array([[1. , 0.99298458],`
□ `[0.99298458, 1.]])`
- A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables. The diagonal of the matrix includes the coefficients between each variable and itself, which is always equal to 1.0.

Pandas DataFrame corr() Method

- ❑ Pandas **dataframe.corr()** is used to find the pairwise correlation of all columns in the Pandas Dataframe in Python.
- ❑ Any NaN values are automatically excluded. To ignore any non-numeric values, use the parameter `numeric_only = True`.
- ❑ **Pandas DataFrame corr() Method Syntax:**
 - ❑ **DataFrame.corr(self, method='pearson', min_periods=1, numeric_only = False)**
 - ❑ Parameters: method:
 - ❑ **pearson:** standard correlation coefficient (Default): evaluates the linear relationship between two continuous variables.
 - ❑ **kendall:** Kendall Tau correlation coefficient : measures the ordinal association between two measured quantities.
 - ❑ **spearman:** Spearman rank correlation evaluates the monotonic relationship between two continuous or ordinal variables.
 - ❑ **min_periods:** Minimum number of observations required per pair of columns to have a valid result. Currently only available for pearson and spearman correlation
 - ❑ **numeric_only:** Whether only the numeric values are to be operated upon or not. It is set to False by default.

pandas

- If we wanted to calculate the correlation between two columns, we could use the pandas method `.corr()`, as follows:
- **`import pandas as pd`**
- **`df['experience'].corr(df['salary'])`**
- 0.9929845761480398
- The `.corr()` includes the parameter "method", which can be used to calculate the three correlation coefficients. By default, it calculated Pearson's.
- **`print(df['experience'].corr(df['salary'], method='spearman'))`**
- **`print(df['experience'].corr(df['salary'], method='kendall'))`**
- 0.9992644353546791
- 0.9958246164193105



correlation matrix with the coefficient of all the pairs of variables

- In case we wanted to explore the correlation between all the pairs of variables, we could simply use the `.corr()` method directly to our DataFrame, which results again in a correlation matrix with the coefficient of all the pairs of variables:
- `df.corr()`



df.corr()

	age	years_in_company	experience	height	children	salary
age	1.000000	0.400034	0.791901	-0.037573	0.239927	0.767315
years_in_company	0.400034	1.000000	0.551120	0.182134	-0.540313	0.549378
experience	0.791901	0.551120	1.000000	0.188684	-0.063857	0.992985
height	-0.037573	0.182134	0.188684	1.000000	-0.330951	0.173324
children	0.239927	-0.540313	-0.063857	-0.330951	1.000000	-0.098195
salary	0.767315	0.549378	0.992985	0.173324	-0.098195	1.000000

Result Explained

- ❑ The Result of the `corr()` method is a table with a lot of numbers that represents how well the relationship is between two columns.
- ❑ The number varies from -1 to 1.
- ❑ 1 means that there is a 1 to 1 relationship (a perfect correlation), and for this data set, each time a value went up in the first column, the other one went up as well.
- ❑ 0.9 is also a good relationship, and if you increase one value, the other will probably increase as well.
- ❑ -0.9 would be just as good relationship as 0.9, but if you increase one value, the other will probably go down.
- ❑ 0.2 means NOT a good relationship, meaning that if one value goes up does not mean that the other will.

Perfect Correlation and

❑ **Perfect Correlation:**

- ❑ We can see that "age" and "age" got the number 1.000000, which makes sense, each column always has a perfect relationship with itself.

❑ **Good Correlation:**

- ❑ "Duration" and "Calories" got a 0.922721 correlation, which is a very good correlation, and we can predict that the longer you work out, the more calories you burn, and the other way around: if you burned a lot of calories, you probably had a long work out.

❑ **Bad Correlation:**

- ❑ "Duration" and "Maxpulse" got a 0.009403 correlation, which is a very bad correlation, meaning that we can not predict the max pulse by just looking at the duration of the work out, and vice versa.

Heatmap()

- The correlation matrix can be very big and difficult to interpret if our DataFrame has many columns.
- we could use a heatmap; a data visualization technique where each value is represented by a color, according to its intensity in a given scale. The fastest way to create a heatmap is by using the function `heatmap()`, available in the seaborn library:
- `import seaborn as sns`
- `sns.heatmap(df.corr(), vmin=-1, vmax=1,`
- `annot=True, cmap="rocket_r")`
- `plt.show()`

Heatmap()

