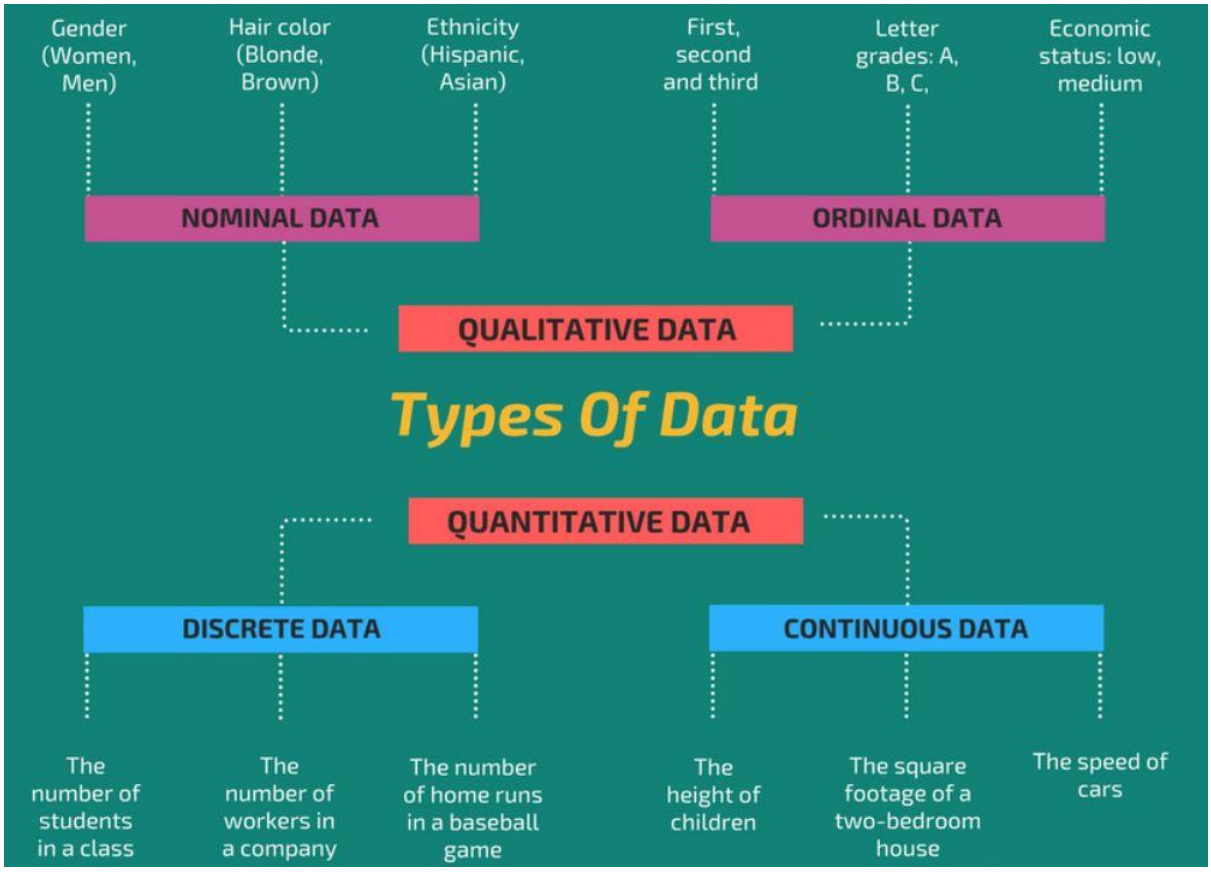
Statistics for Data Science :

Types of Data :



Qualitative or Categorical Data:

The qualitative data, also known as the categorical data, describes the data that fits into a category. There are no numerical values in qualitative data. In categorical information, categorical variables describe the features of a person, such as gender, home town, etc. The categorical measures are defined in natural language specifications rather than numerically.

Categorical data can sometimes have numerical values (quantitative value), but the values are not mathematically meaningful.

For example:
Birthdate : 5-9-1992,17-04-1994
Favourite sport – Football,kabaddi,
School postcode : 629181,636346,629675.

Nominal Data

There is no meaningful zero with nominal data, as it represents discrete units, which is why it cannot be ordered and measured. They are used to label variables without providing any quantitative value.

Data scientists use hot encoding, to transform nominal data into a numeric feature.

Gender (Women, Men)
Hair color (Blonde, Brown, Brunette, Red, etc.)
Marital status (Married, Single, Widowed)
Ethnicity (Hispanic, Asian)

Ordinal Data :

Unlike nominal values, ordinal values represent discrete and ordered units. However, there is no consistency between adjacent categories, and ordinal data also lack a meaningful zero.

Data scientists use label encoding to transform ordinal data into a numeric feature.

The first, second and third person in a competition.
Letter grades: A, B, C, and etc.
When a company asks a customer to rate the sales experience on a scale of 1-10.
Economic status: low, medium and high

Examples

Opinion (agree, mostly agree, neutral, mostly disagree, disagree)
Socioeconomic status (low income, middle income, high income)

Quantitative or Numerical Data

Discrete data – a count that involves integers. Only a limited number of values is possible. The discrete values cannot be subdivided into parts. For example, the number of children in a school is discrete data. You can count whole individuals. You can't count 1.5 kids.

Continuous data – information that could be meaningfully divided into finer levels. It can be measured on a scale or continuum and can have almost any numeric value. For example, you can measure your height at very precise scales — meters, centimeters, millimeters and etc

A probability distribution determines the probability of all the outcomes a random variable takes. The distribution can either be continuous or discrete distribution depending upon the values that a random variable takes. There are several types of probability distribution like Normal distribution, Uniform distribution, exponential distribution, etc. In this article, we will see about Normal distribution and we will also see how we can use Python to plot the Normal distribution.

Central Limit Theorem:

It states if we take the mean of large no data points collected from independent and identical distributed random variables then these mean will follow a normal distribution regardless of their original distribution.

Normal Distribution

The normal distribution is a continuous probability distribution function also known as Gaussian distribution which is symmetric about its mean and has a bell-shaped curve. It is one of the most used probability distributions. Two parameters characterize it

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

```
import numpy as np
import matplotlib.pyplot as plt

# Mean of the distribution
Mean = 100

# satndard deviation of the
distribution
Standard_deviation  = 5

# size
size = 100000

# creating a normal distribution data
values = np.random.normal(Mean,
Standard_deviation, size)

# plotting histograph
plt.hist(values, 100)
# plotting mean line
plt.axvline(values.mean(), color='k',
linestyle='dashed', linewidth=2)
plt.show()
```

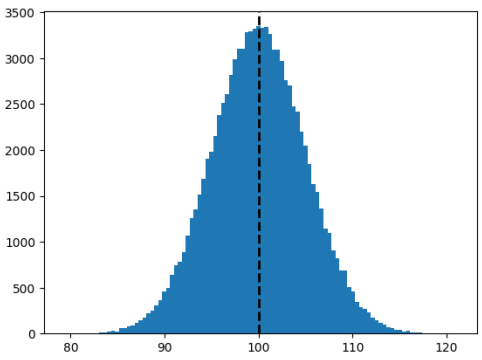
https://www.geeksforgeeks.org/

Characteristics for a Normal Distribution:

Symmetric distribution – The normal distribution is symmetric about its mean point. It means the distribution is perfectly balanced toward its mean point with half of the data on either side.

Bell-Shaped curve – The graph of a normal distribution takes the form bell-shaped curve with most of the points accumulated at its mean position. The shape of this curve is determined by the mean and standard deviation of the distribution

Empirical Rule – The normal distribution curve follows the empirical rule where 68% of the data lies within 1 standard deviation from the mean of the graph, 95% of the data lies within 2 standard deviations from the mean and 97% of the data lies within 3 standard



Descriptive Statistics:

Descriptive statistics summarize and organize characteristics of a data set. A data set is a collection of responses or observations from a sample or entire population

Measures of Central Tendency:

Measures of central tendency estimate the centre, or average, of a data set. The mean, median and mode are 3 ways of finding the average.

Dataset : 15, 3, 12, 0, 24, 3

Ordered Dataset: 0, 3, 3, 12, 15, 24

Data set	15, 3, 12, 0, 24, 3	
Mean	Sum/N	9.5
Median	(3 + 12)/2	7.5
Mode		3

Raw data	**Deviation from mean**	**Squared deviation**
15	15 – 9.5 = 5.5	30.25
3	3 – 9.5 = -6.5	42.25
12	12 – 9.5 = 2.5	6.25
0	0 – 9.5 = -9.5	90.25
24	24 – 9.5 = 14.5	210.25
	3 – 9.5 = -6.5	42.25

The **mean**, or *M*, is the most commonly used method for finding the average.

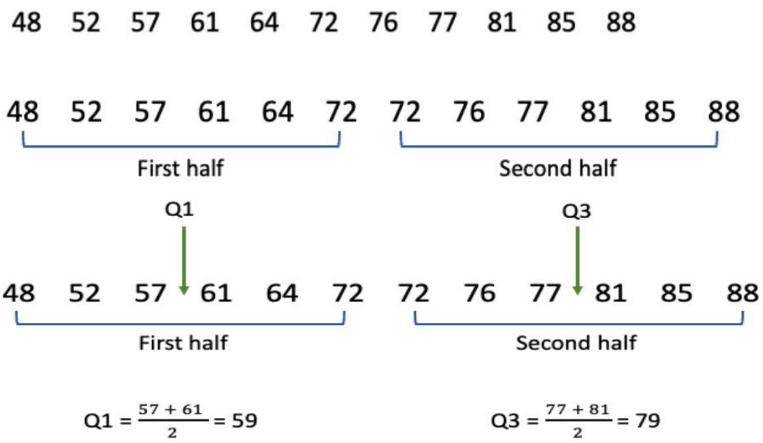
The **median** is the value that's exactly in the middle of a data set.

The **mode** is the simply the most popular or most frequent response value. A data set can have no mode, one mode, or more than one mode.

The **interquartile range** tells you the spread of the middle half of your distribution.

Interquartile Range (IQR)

The **interquartile range** tells you the spread of the middle half of your distribution.



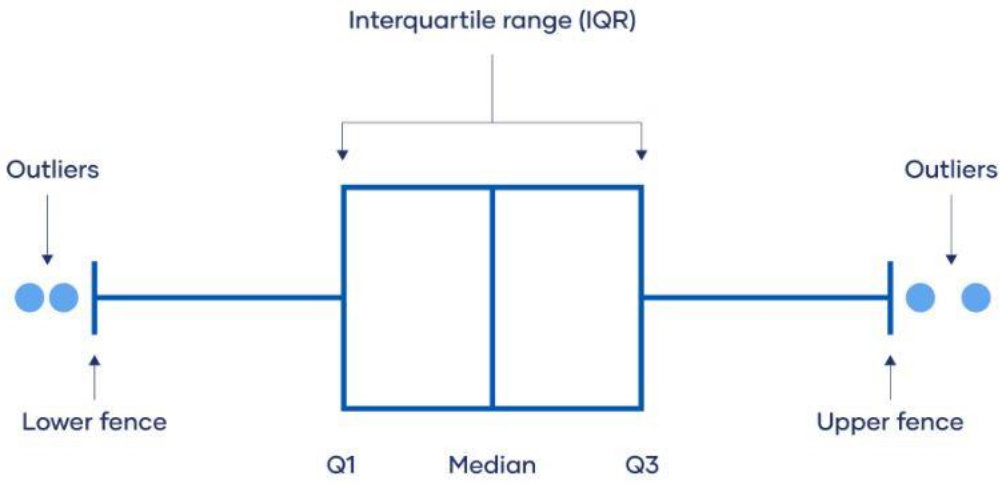
$Q1 = \frac{57 + 61}{2} = 59$

$Q3 = \frac{77 + 81}{2} = 79$

$$IQR = Q3 - Q1$$
$$IQR = 79 - 59 = 20$$

Every distribution can be organized using these five numbers:

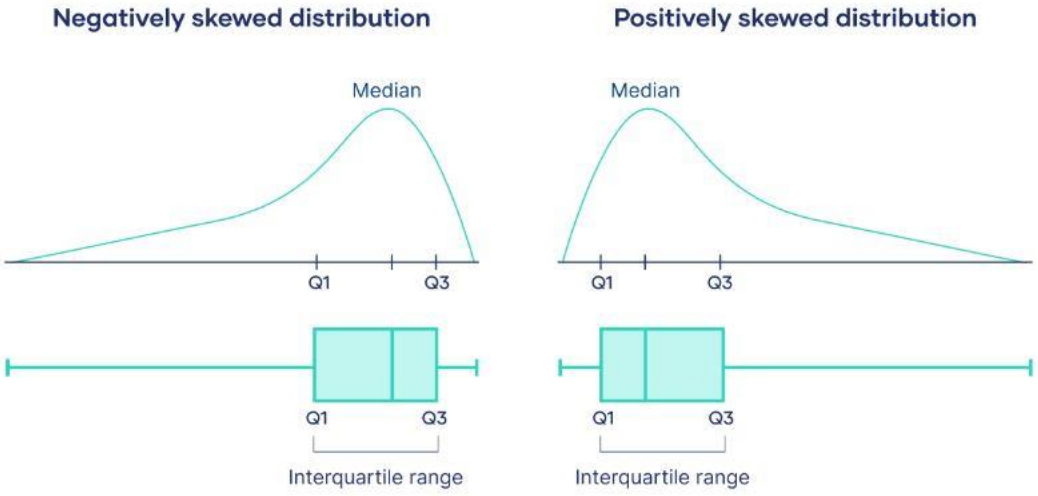
- Lowest value
- Q1: 25th percentile
- Median
- Q3: 75th percentile
- Highest value (Q4)



Skewness — The measure of asymmetry in a probability distribution is defined by Skewness. It can either be positive, negative, or undefined.

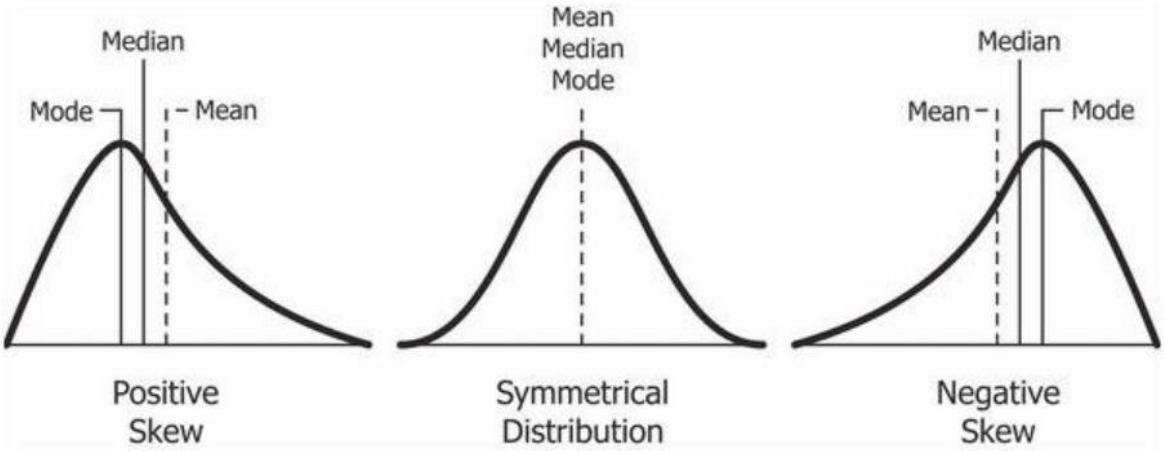
Positive Skew — This is the case when the tail on the right side of the curve is bigger than that on the left side. For these distributions, the mean is greater than the mode.

Negative Skew — This is the case when the tail on the left side of the curve is bigger than that on the right side. For these distributions, the mean is smaller than the mode.



Skewness:

$$Skewness = \frac{3\,(Mean - Median)}{Std\ \ Deviation}$$



The distribution of skewness values is as below:

- Skewness = 0 when the distribution is normal.
- Skewness > 0 or positive when more weight is on the left side of the distribution.
- Skewness < 0 or negative when more weight is on the right side of the distribution.

Example:

Consider the following 10-number sequence that represents the scores of a competitive exam.
X = [54, 73, 59, 98, 68, 45, 88, 92, 75, 96]

By calculating the mean of X, we can get: $\bar{x} = 74.8$

Solving it with the skewness formula:

$$m_3 = [(54 - 74.8)^3 - (73 - 74.8)^3 - \ldots\ldots - (96 - 74.8)^3]/10$$

The Fisher-Pearson Coefficient of Skewness is equal to 0.745631. You can see that there is a positive skew in the data.

Skewness

Pearson's Skewness Coefficient

$$Skewness = \frac{\bar{x} - median}{s}$$

If skewness < -.20 severe left skewness
If skewness > +.20 severe right skewness

Fisher's Measure of Skewness has a complicated formula but most software packages compute it.

Fisher's Skewness > 1.00 moderate right skewness
> 2.00 severe right skewness
Fisher's Skewness < -1.00 moderate left skewness
< -2.00 severe right skewness

Kurtosis

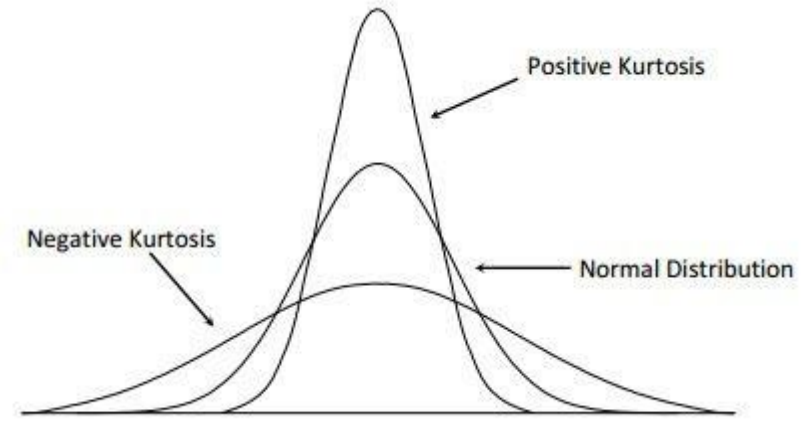
$$\text{Kurtosis} = \frac{\sum(x_i - \bar{x})^4}{nS^4}$$

$\bar{x}$ = mean of the given data
S = standard deviation of the data
n = total number of observations

Example:

We again consider a sequence of 10 numbers that represent the scores of a competitive exam. X = [54, 73, 59, 98, 68, 45, 88, 92, 75, 96]

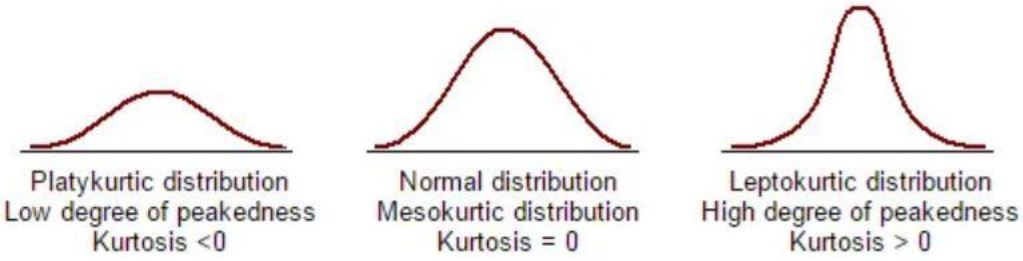
By calculating the mean of X, we can get: $\bar{x} = 74.8$



Kurtosis of a normal distribution is equal to 3. When the kurtosis is less than 3, it is known as platykurtic, and when it is greater than 3, it is leptokurtic. If it is leptokurtic, it will signify that it produces outliers rather than a normal distribution.

Kurtosis — Kurtosis describes whether the data is light-tailed (lack of outliers) or heavy-tailed (outliers present) when compared to a Normal distribution. There are three kinds of Kurtosis:

- **Mesokurtic** — This is the case when the kurtosis is zero, similar to the normal distributions.
- **Leptokurtic** — This is when the tail of the distribution is heavy (outlier present) and kurtosis is higher than that of the normal distribution.
- **Platykurtic** — This is when the tail of the distribution is light(no outlier) and



- kurtosis is lesser than that of the normal distribution.

Calculating Skewness and Kurtosis in python :

```
# importing
import SciPy
```

Step 2: Creating a dataset

The next step is to create a dataset. The code below shows how.

```
# creating a data set
dataset = [10, 25, 14, 26, 35, 45, 67, 90, 40, 50, 60, 10, 16, 18, 20]
```

Step 3: Computing skewness

Use the following syntax to calculate the skewness by using the in-built skew() function.

```
spicy.stats.skew(array, axis = 0, bias = True)
```

Step 4: Computing kurtosis

Calculate the kurtosis with the help of the in-built kurtosis() function using the syntax below:

```
spicy.stats.kurtosis(array, axis = 0, fisher = True, bias = True)
```

where the array is the input object that has the elements, and the axis represents the axis along with the kurtosis value that needs to be measured.

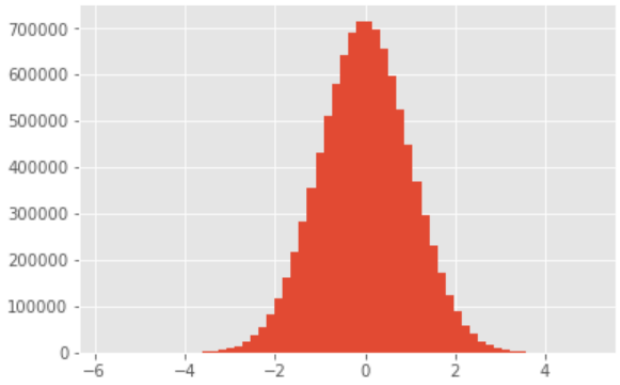
Fisher = True when normal is 0.0. It will be False when the normal is 3.0. Bias is True or False, based on statistical bias.

VISUALISING DISTRIBUTIONS

```
%matplotlib inline
import numpy as np
import pandas as pd
from scipy.stats import kurtosis
from scipy.stats import skew

import matplotlib.pyplot as plt

plt.style.use('ggplot')

data = np.random.normal(0, 1, 10000000)
np.var(data)

plt.hist(data, bins=60)

print("mean : ", np.mean(data))
print("var  : ", np.var(data))
print("skew : ",skew(data))
print("kurt : ",kurtosis(data))
```

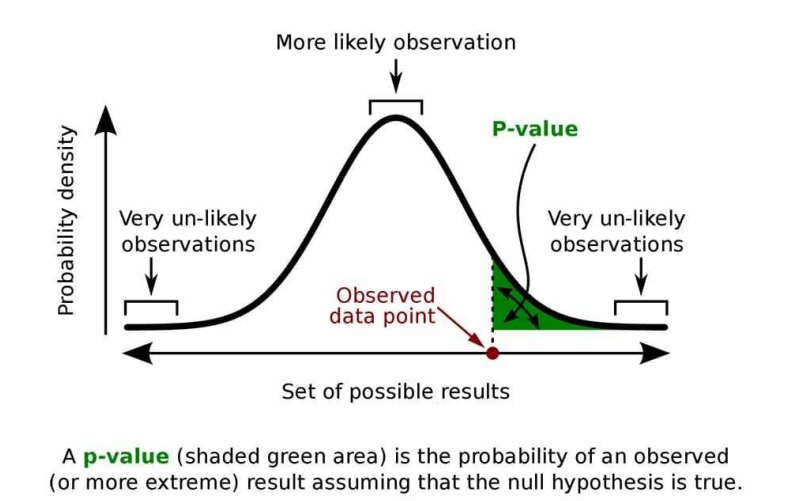
Output:

```
mean :  0.0004102135008847
var  :  0.9998277169790
skew :  0.00012294118186476907
kurt :  0.0033554829466604374
```



Understanding P-values

The ***p* value** is a number, calculated from a statistical test, that describes how likely you are to have found a particular set of observations if the null hypothesis were true.



A **p-value** (shaded green area) is the probability of an observed (or more extreme) result assuming that the null hypothesis is true.

What A P-Value Tells You:

The null hypothesis (H0) states that there is no relationship between the two variables being studied (one variable does not affect the other). It states the results are due to chance and are not significant in terms of supporting the idea being investigated. Thus, the null hypothesis assumes that whatever you try to prove did not happen.

The alternative hypothesis (Ha or H1) is the one you would believe if the null hypothesis is concluded to be untrue.

The level of statistical significance is often expressed as a *p* -value between 0 and 1. The smaller the p-value, the stronger the evidence that you should reject the null hypothesis.

- A *p* -value less than 0.05 (typically ≤ 0.05) is statistically significant. It indicates strong evidence against the null hypothesis, as there is less than a 5% probability the null is correct (and the results are random). Therefore, we reject the null hypothesis and accept the alternative hypothesis.

- However, if the *p* -value is below your threshold of significance (typically $p < 0.05$), you can reject the null hypothesis, but this does not mean that

DEFNITION

P values are used in hypothesis testing to help decide whether to reject the null hypothesis. The smaller the *p* value, the more likely you are to reject the null hypothesis.

A *p* -value higher than 0.05 (> 0.05) is not statistically significant and indicates strong evidence for the null hypothesis. This means we retain the null hypothesis and reject the alternative hypothesis. You should note that you cannot accept the null hypothesis; we can only reject it or fail to reject it.

there is a 95% probability that the alternative hypothesis is true.

-

Performing a T-Test in Python:

The t-test is a statistical test that can be used to determine if there is a significant difference between the means of two independent samples of data.

$$t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

One sample t-test:

```
sys_bp=[183, 152, 178, 157, 194, 163, 144, 114, 178, 152, 118, 158, 172, 138]
mu=165
from scipy import stats
t_value,p_value=stats.ttest_1samp(sys_bp,mu)
one_tailed_p_value=float("{:.6f}".format(p_value/2))
# Since alternative hypothesis is one tailed, We need to divide the p value by 2.
print('Test statistic is %f'%float("{:.6f}".format(t_value)))
print('p-value for one tailed test is %f'%one_tailed_p_value)
alpha = 0.05
if one_tailed_p_value<=alpha:
print('Conclusion','n','Since p value(=%f)'%p_value,'<','alpha(=%.2f)'%alpha,'''We reject the null hypothesis H0. So we conclude that there is no significant mean difference in systolic blood pressure. i.e., μ = 165 at %.2f level of significance'''%alpha)
else:
print('Conclusion','n','Sincep-value(=%f)'%one_tailed_p_value, '>', 'alpha(=%.2f)'%alpha,'We do not reject the null hypothesis H0.')
```

Assignments :
Data:
Systolic blood pressures of 14 patients are given below:
183, 152, 178, 157, 194, 163, 144, 114, 178, 152, 118, 158, 172, 138
Test, whether the population mean, is less than 165
Hypothesis
H_0: There is no significant mean difference in systolic blood pressure. i.e., μ = 165
H_1: The population mean is less than 165. i.e., μ < 165

Two sample t-test

Compare the effectiveness of ammonium chloride and urea, on the grain yield of paddy, an experiment was conducted. The results are given below:

Ammonium chloride (X_1)	13.4	10.9	11.2	11.8	14	15.3	14.2	12.6	17	16.2	16.5	15.7
Urea (X_2)	12	11.7	10.7	11.2	14.8	14.4	13.9	13.7	16.9	16	15.6	16

Hypothesis
H_0: The effect of ammonium chloride and urea on grain yield of paddy are equal i.e., $\mu_1 = \mu_2$
H_1: The effect of ammonium chloride and urea on grain yield of paddy is not equal i.e., $\mu_1 \neq \mu_2$

Code :

```
Ammonium_chloride=[13.4,10.9,11.2,11.8,14,15.3,14.2,12.6,17,16.2,16.5,15.7]
Urea=[12,11.7,10.7,11.2,14.8,14.4,13.9,13.7,16.9,16,15.6,16]
from scipy import stats
t_value,p_value=stats.ttest_ind(Ammonium_chloride,Urea)
print('Test statistic is %f'%float("{:.6f}".format(t_value)))
print('p-value for two tailed test is %f'%p_value)
alpha = 0.05
if p_valu<=alpha:
   print('Conclusion','n','Since p-value(=%f)'%p_value,'<','alpha(=%.2f)'%alpha,'''We reject the null hypothesis H0. So we conclude that the effect of ammonium chloride and urea on grain yield of paddy are not equal i.e., µ1 = µ2 at %.2f level of significance.'''%alpha)
else:
print('Conclusion','n','Since p-value(=%f)'%p_value,'>','alpha(=%.2f)'%alpha,'''We do not reject the null hypothesis H0.
```

Paired T – test

Use Case:

Eleven schoolboys were given a test in Statistics. They were given a Month's tuition and a second test were held at the end of it. Do the marks give evidence that the students have benefited from the exam coaching?
Marks in 1st test: 23 20 19 21 18 20 18 17 23 16 19
Marks in 2nd test: 24 19 22 18 20 22 20 20 23 20 18

Hypothesis
H_0: The students have not benefited from the tuition class. i.e., d = 0
H_1: The students have benefited from the tuition class. i.e., d < 0
Where, d = x-y; d is the difference between marks in the first test (say x) and marks in the second test (say y).

Python Code 01:

```
alpha = 0.05
first_test =[23, 20, 19, 21, 18, 20, 18, 17, 23, 16, 19]
second_test=[24, 19, 22, 18, 20, 22, 20, 20, 23, 20, 18]
from scipy import stats
t_value,p_value=stats.ttest_rel(first_test,second_test)
one_tailed_p_value=float("{:.6f}".format(p_value/2))
print('Test statistic is %f'%float("{:.6f}".format(t_value)))
print('p-value for one_tailed_test is %f'%one_tailed_p_value)
alpha = 0.05
if one_tailed_p_value<=alpha:
   print('Conclusion','n','Since p-
value(=%f)'%one_tailed_p_value,'<','alpha(=%.2f)'%alpha,'''We reject the null hypothesis
H0.
So we conclude that the students have benefited by the tuition class. i.e., d = 0 at %.2f level
of significance.'''%alpha)
else:
   print('Conclusion','n','Since p-
value(=%f)'%one_tailed_p_value,'>','alpha(=%.2f)'%alpha,'''We do not reject the null
hypothesis H0.
So we conclude that the students have not benefited by the tuition class. i.e., d = 0 at %.2f
level of significance.'''%alpha)
```

Python Code 02:

```
# Python program to conduct
# two-sample t-test using statsmodels

# Importing library
from statsmodels.stats.weightstats import ttest_ind
import numpy as np
import pingouin as pg

# Creating data groups
data_group1 = np.array([160, 150, 160, 156.12,
                        163.24,
                        160.56, 168.56, 174.12,
                        167.123, 165.12])
data_group2 = np.array([157.97, 146, 140.2, 170.15,
                        167.34, 176.123, 162.35,
                        159.123, 169.43, 148.123])

# Conducting two-sample ttest
ttest_ind(data_group1, data_group2)
```

Referrences :
https://thedatascientist.com/how-to-do-a-t-test-in-python/
https://www.geeksforgeeks.org/how-to-conduct-a-two-sample-t-test-in-python/

ANOVA – Analysis of variance

ANOVA, which stands for Analysis of Variance, is a statistical test used to analyze the difference between the means of more than two groups.

A one-way ANOVA uses one independent variable, while a two-way ANOVA uses two independent variables.

One-way ANOVA: Testing the relationship between shoe brand (Nike, Adidas, Saucony, Hoka) and race finish times in a marathon.
Two-way ANOVA: Testing the relationship between shoe brand (Nike, Adidas, Saucony, Hoka), runner age group (junior, senior, master's), and race finishing times in a marathon.

NOTE : *All ANOVAs are designed to test for differences among three or more groups. If you are only testing for a difference between two groups, use a t-test instead.*

Problem Statement :

A dataset, students.csv, contains 8239 rows of student particular data. Each row represents a unique student. It consists of 16 features related to the student and we will only focus on 3 features major, gender and salary .

Based on the two factor, major and gender, is there significant difference in average annual salary for graduates of different gender and major and also if there is any interaction between gender and major at 5% significance level?

Analysis vis python code :

ANOVA- Analysis Of Varience

ANOVA uses the F test, a groupwise comparison test, for statistical significance. It compares the variance in each group's mean under different factors (factor A, factor B, interaction between factor A & factor B) to the overall variance in the dependent variable. Finally, based on the F-test statistic, a conclusion is made

Students.csv – preview

	major	gender	count
0	Biology	Female	40
1	Biology	Male	40
2	Economics and Finance	Female	40
3	Economics and Finance	Male	40
4	Environmental Sciences	Female	40
5	Environmental Sciences	Male	40
6	Mathematics and Statistics	Female	40
7	Mathematics and Statistics	Male	40
8	Political Science	Female	40
9	Political Science	Male	40
10	Social Sciences	Female	40
11	Social Sciences	Male	40

```
import random
import pandas as pd

# read original dataset
student_df = pd.read_csv('students.csv')

# filter the students who are graduated
graduated_student_df = student_df[student_df['graduated'] == 1]

# random sample for 40 students in each group
sample_df = graduated_student_df[['major', 'gender', 'salary']].groupby(['major',
'gender']).apply(lambda x: x.sample(n=40, random_state=7)).reset_index(drop=True)

# three variables of interest
sample_df = sample_df[['major', 'gender', 'salary']]
groups = sample_df.groupby(['major', 'gender']).agg({'salary':
'count'}).reset_index().rename(columns={'salary':'count'})
display(groups)
```

Hypothesis Testing

According to five steps process of hypothesis testing:
Set 1:
H_o: $\mu_{a1} = \mu_{a2} = \mu_{a3} = \ldots = \mu_{a6}$
H_1: Not all salary means are equal under different major
Set 2:
H_o: $\mu_{\beta 1} = \mu_{\beta 2}$
H_1: Not all salary means are equal under different gender
Set 3:
H_o: There is no interaction between major and gender
H_1: There is an interaction between major and gender
α = 0.05
According to F test statistics:

Code continued …….

```
# Create
ANOVA
backbone
table with
interaction
                data = [['major', '', '', '', '', '', ''],
                        ['gender', '', '', '', '', '', ''],
                        ['major x gender', '', '', '', '', '', ''],
```

```
        ['Within Groups', '', '', '', '', '', ''],
        ['Total', '', '', '', '', '', '']]
anova_table = pd.DataFrame(data, columns = ['Source of Variation', 'SS', 'df',
'MS', 'F', 'P-value', 'F crit'])
anova_table.set_index('Source of Variation', inplace = True)

# calculate SSA and update anova table - A = major
x_bar = sample_df['salary'].mean()
SSA = sample_df.groupby('major').agg({'salary':'count'}) *
(sample_df.groupby('major').agg({'salary':'mean'}) - x_bar)**2
anova_table['SS']['major'] = SSA['salary'].sum()

# calculate SSB and update anova table - B = gender
SSB = sample_df.groupby('gender').agg({'salary':'count'}) *
(sample_df.groupby('gender').agg({'salary':'mean'}) - x_bar)**2
anova_table['SS']['gender'] = SSB['salary'].sum()

# calculate SSE and update anova table
SSE = (sample_df.groupby(['gender', 'major']).agg({'salary':'count'}) - 1) *
sample_df.groupby(['gender', 'major']).agg({'salary':'std'})**2
anova_table['SS']['Within Groups'] = SSE['salary'].sum()

# calculate SSTO and update anova table
SSTO = pd.DataFrame((sample_df['salary'] - x_bar)**2, columns=['salary'])
anova_table['SS']['Total'] = SSTO['salary'].sum()

# calculate SSAB and update anova table - AB = interaction between major and
gender
SSAB = SSTO['salary'].sum() - SSA['salary'].sum() - SSB['salary'].sum() -
SSE['salary'].sum()
anova_table['SS']['major x gender'] = SSAB


# update degree of freedom
anova_table['df']['major'] = sample_df['major'].nunique() - 1
anova_table['df']['gender'] = sample_df['gender'].nunique() - 1
anova_table['df']['major x gender'] = (sample_df['major'].nunique() - 1) *
(sample_df['gender'].nunique() - 1)
anova_table['df']['Within Groups'] = sample_df.shape[0] -
sample_df['major'].nunique() * sample_df['gender'].nunique()
anova_table['df']['Total'] = sample_df.shape[0] - 1

# calculate MS
anova_table['MS']['major'] = anova_table['SS']['major'] /
anova_table['df']['major']
```

```
anova_table['MS']['gender'] = anova_table['SS']['gender'] /
anova_table['df']['gender']
anova_table['MS']['major x gender'] = anova_table['SS']['major x gender'] /
anova_table['df']['major x gender']
anova_table['MS']['Within Groups'] = anova_table['SS']['Within Groups'] /
anova_table['df']['Within Groups']

# calculate F
anova_table['F']['major'] = anova_table['MS']['major'] /
anova_table['MS']['Within Groups']
anova_table['F']['gender'] = anova_table['MS']['gender'] /
anova_table['MS']['Within Groups']
anova_table['F']['major x gender'] = anova_table['MS']['major x gender'] /
anova_table['MS']['Within Groups']

# p-value
anova_table['P-value']['major'] = 1 - stats.f.cdf(anova_table['F']['major'],
anova_table['df']['major'], anova_table['df']['Within Groups'])
anova_table['P-value']['gender'] = 1 - stats.f.cdf(anova_table['F']['gender'],
anova_table['df']['gender'], anova_table['df']['Within Groups'])
anova_table['P-value']['major x gender'] = 1 - stats.f.cdf(anova_table['F']['major x
gender'], anova_table['df']['major x gender'], anova_table['df']['Within Groups'])

# F critical
alpha = 0.05
anova_table['F crit']['major'] = stats.f.ppf(1-alpha, anova_table['df']['major'],
anova_table['df']['Within Groups'])
anova_table['F crit']['gender'] = stats.f.ppf(1-alpha, anova_table['df']['gender'],
anova_table['df']['Within Groups'])
anova_table['F crit']['major x gender'] = stats.f.ppf(1-alpha,
anova_table['df']['major x gender'], anova_table['df']['Within Groups'])

# Final ANOVA Table
anova_table
```

Source of Variation	SS	df	MS	F	P-value	F crit
major	16878851261.383152	5	3375770252.27663	72.056863	0.0	2.233275
gender	8710635244.028683	1	8710635244.028683	185.931212	0.0	3.861405
major x gender	291187390.593399	5	58237478.11868	1.243097	0.287768	2.233275
Within Groups	21925190814.148491	468	46848698.32083			
Total	47805864710.153725	479				

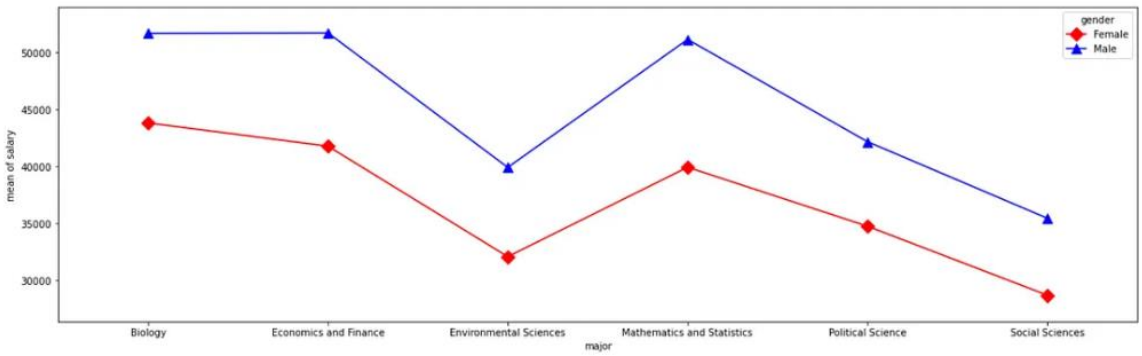
Using Statsmodels Package :

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
# perform two-way ANOVA with interaction
model = ols('salary ~ C(major) + C(gender) + C(major):C(gender)', data=sample_df).fit()
sm.stats.anova_lm(model, typ=2)
```

	sum_sq	df	F	PR(>F)
C(major)	1.687885e+10	5.0	72.056863	7.559506e-56
C(gender)	8.710635e+09	1.0	185.931212	6.924772e-36
C(major):C(gender)	2.911874e+08	5.0	1.243097	2.877676e-01
Residual	2.192519e+10	468.0	NaN	NaN

interaction plot of major and gender on salary:

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from statsmodels.graphics.factorplots import interaction_plot
fig, ax = plt.subplots(figsize=(20, 6))
fig = interaction_plot(
    x=sample_df['major'],
    trace=sample_df['gender'],
    response=sample_df['salary'],
    colors=["red", "blue"],
    markers=["D", "^"],
    ms=10,
    ax=ax,
)
```



Result Analysis :

For Set 1 & Set 2: Null hypothesis is rejected since F score > F critical or p-value is < 0.05. We have enough evidence that not all average salaries are the same for graduates of different study subjects or gender, at 5% significance level.

For Set 3: Failed to reject the null hypothesis. ∴We do not have enough evidence that study subjects and gender has interaction, at 5% significance level. Moreover from interaction plot, it shows that there is no interaction, and both main effects, major and gender effects, are significant. For example, the average salaries of graduates will be significantly higher for males who graduated in Biology.

References :

https://towardsdatascience.com/two-way-anova-test-with-python-a112e2396d78