Statistics for ML

Tutorial 1 : What is statistics and its type?

Statistsics is the science of collecting, organizing and analyzing data

what is Data: Facts or pieces of information e.g 1. Hight of students in a class 2. Gender of person visiting a doctor(8 AM TO 4 PM)

Types of Statistics:

a. **Descriptive Stat** : It consisit of organizing and summaring of data

```
1. Measure central tendency
2. Measure of dispersion (variance & standard deviation)
3. Different type of distribution of data
4. Histrogram, Probability Density Function, Probability Mass
Function
```

b. **Inferential Stat** : It consist of using data you have measure to form conclusion

```
Hypothesis Testing:
1. Z-test
2. T-test
3. ChiSQURE Test
4. Annova Test
```

e.g on both type : lets consider there are 20 classrooms in a college and lets say you have collected ages of students in one class

Ages in class : {91, 21, 18, 34, 22 ,23}

Descriptive Stat ouestion: what is the common age in your statts class? > mean(ages)

Inferential Stat question: Are the ages of student in the classroom simillar to ages of the students in the university?

Tutorial 2 : Population & Sample

Population (N) : The group that we are interested in study.

Sample (n) : It is subset of the population

```
e.g > Exit Poll (Media)

State A : Age > 18 Years (population)
State A : region wise small group of peoples (sample)
State B : Avg height of the people (population)
State B : Small Group of people in specific region (sample)
```

Sampling Technique : The goal of sampling is to create a sample that is represented of the entire population.

Types of Sampling :

1.**Simple Random Sampling** : when performing simple random sampling */every member of population(N) has an equal chance of being selected for your sample(n)/*

2.**Stratified Sampling** (layering) layring from non overlapping group : e.g : population : 1. male 2. female

3.**Systematic Sampling** : e.g : Mall (survey) > every 4th person survey

4.**Convinence Sampling** (Volentory Response Sampling) : e.g : Data science knowldege person will be required for survey in data science

5.**Cluster Sampling** Cluster sampling is a method where the researchers divide the entire population into sections or clusters that represent a population.

Questions:

1. Exit Poll sampling technique : random sampling
2. Desies information : convinient sampling
3. Household expence : stratified sampling

Tutorial 3 : Variables

What are the Variable and its types?

Varible : Variable is a property that can take on many values

Varible >> Singular mode

```
Example : age
          12
          13
          14
          40
```

```
Height
172
172.5
180
111
112
Weight = 72kg , 86kg
```

Variables >> Plural mode

Example : Ages = [12,14,16,18,20]

Varible Types :

1. **Quantitative Variable**

```
a) **Discrete Variable**
       Example : No. of Childrens : 1, 2, 3
                 No. of Bank Account : 2, 3, 4

b) **Continous Variable**
       Example : Height > 175cm, 180.10cm
                 Weight > 96kg, 80kg, 100kg
```

2. **Catagorical / Qualitative Variable** Based on some properties variables will be classify Example : Classification Gender : Male Female Type of Flowers : Rose, Lilly

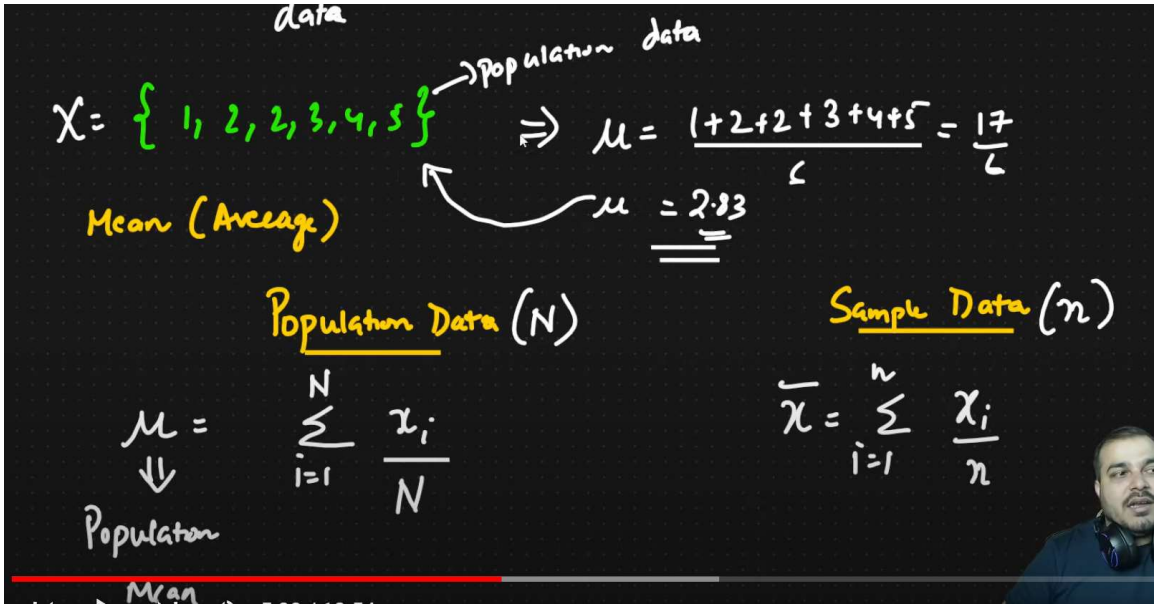
Tutorial 4 : Measure of Central Tendency - Mean, Median & Mode

Central Tendency : Central Tendency referes to the measure used to determine the "Center" of the distribution of data

1. **Mean(Average):** Measure of central tendency

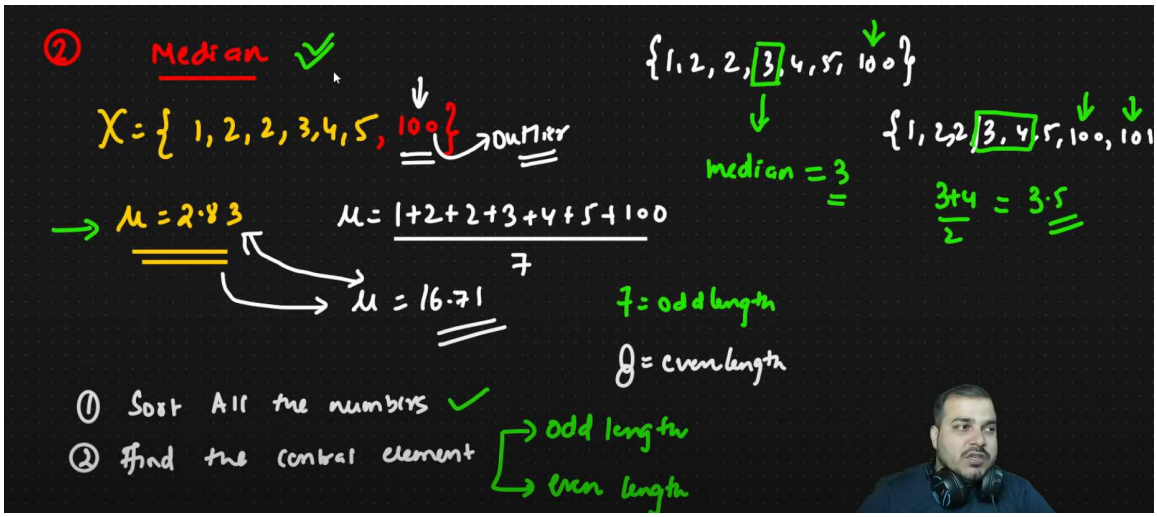
   Population Mean (N):

   Sample Data (n):



when outliers are not present in data that time we can use mean

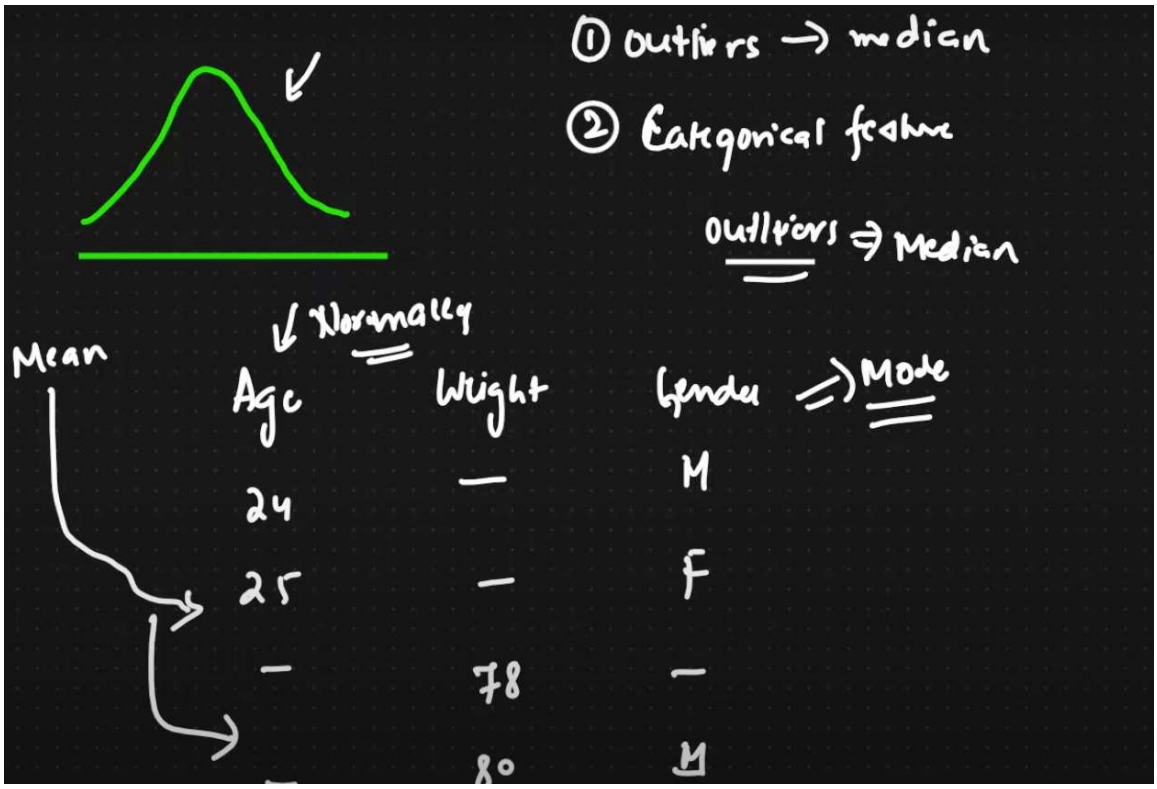
2. **Median** : Measure of central tendency



when outliers in the data that time we can use median

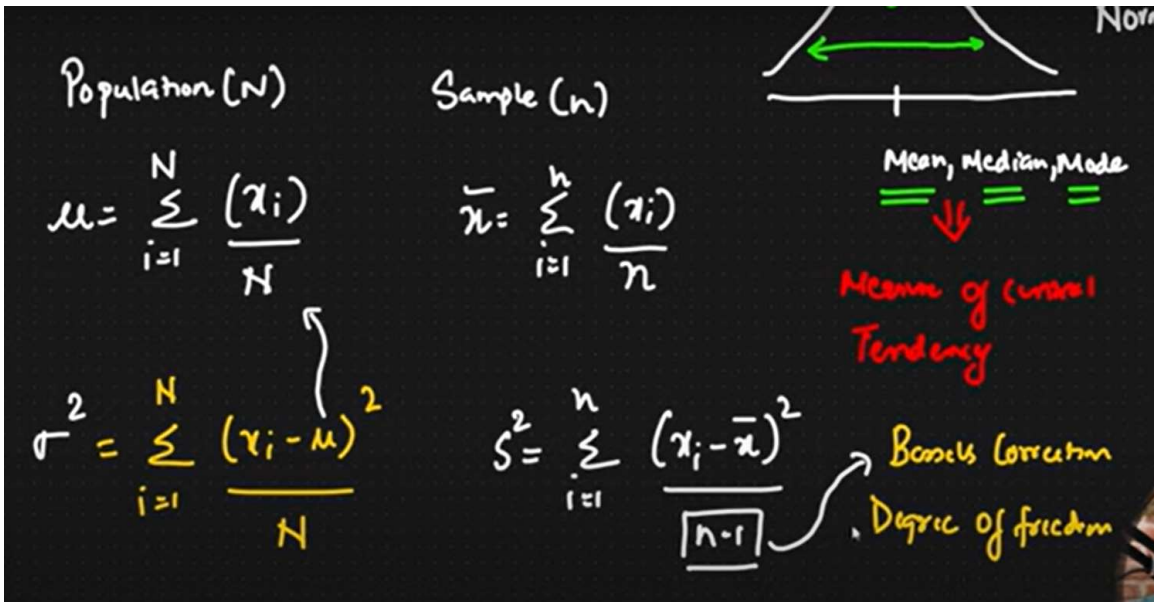
3. **Mode** : Measure of most frequent element

when categorical features will be there we can use mode

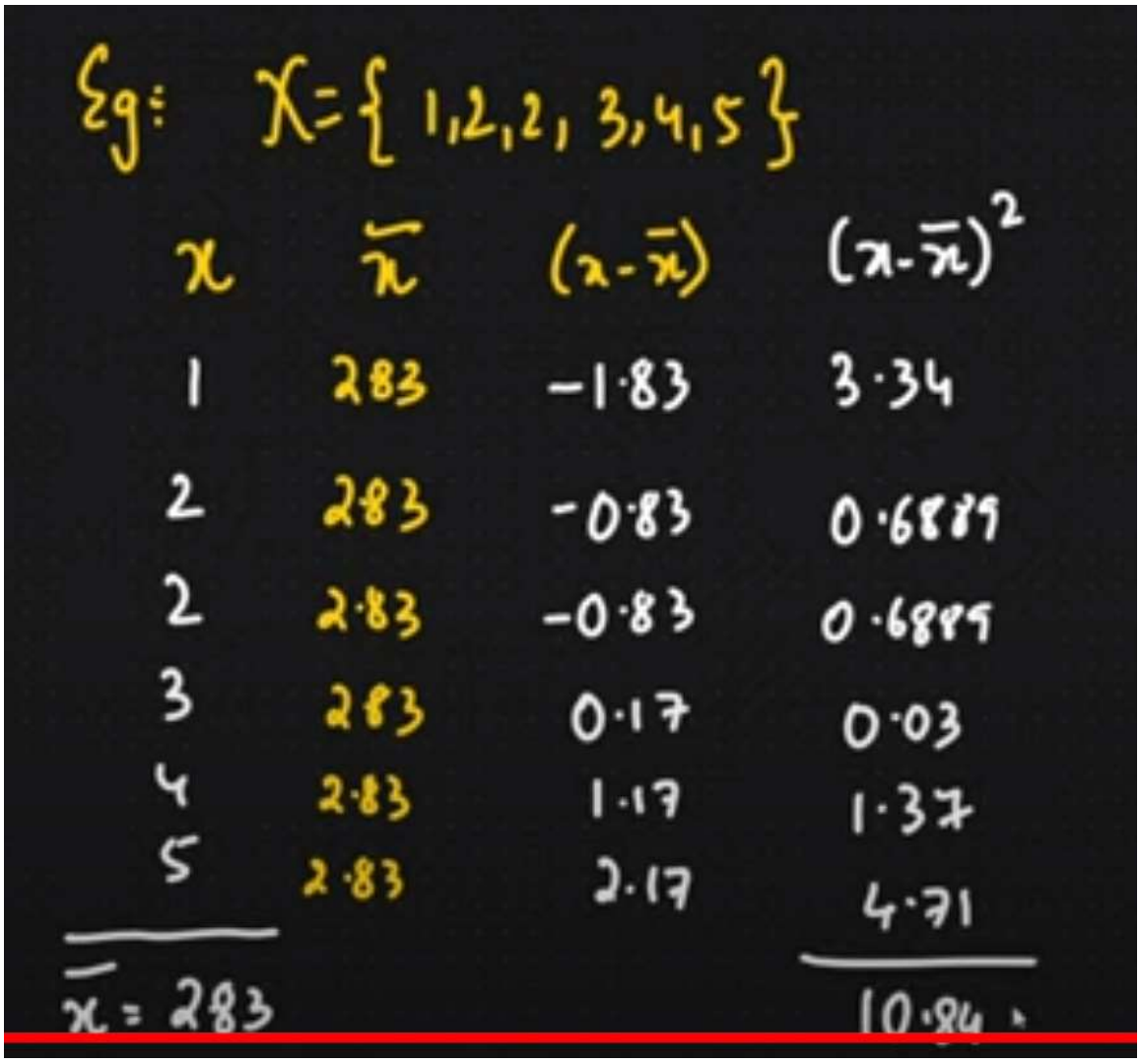


Tutorial 5 : Measure of Dispersion - Variance & Standard Deviation

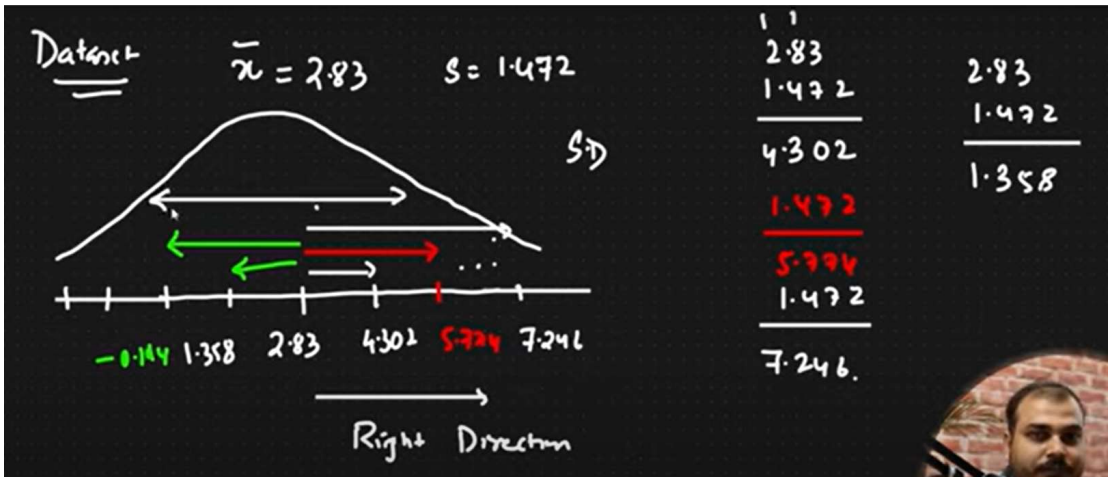
Measure of dispersion : This tells us about distribution of data

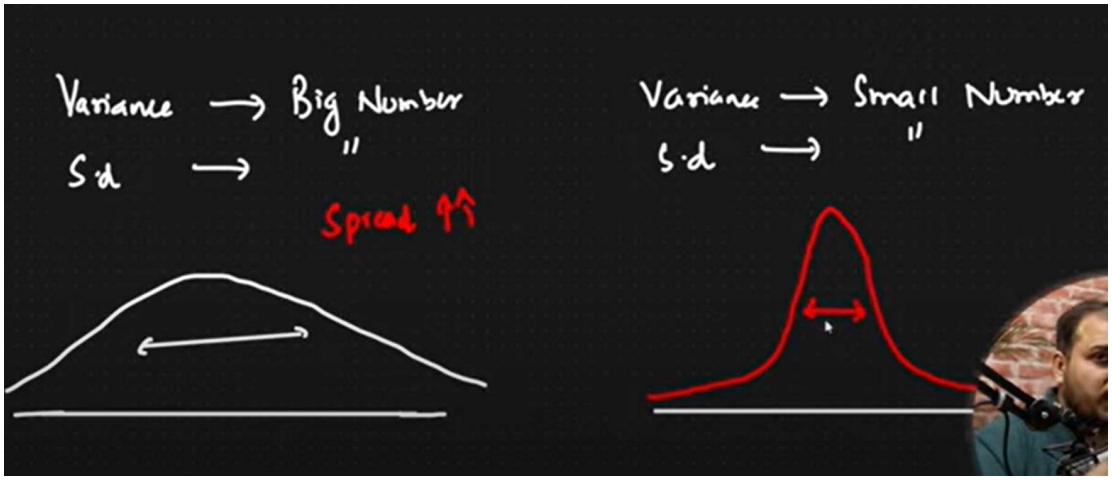


s square is sample variance which tells us about variance in samples

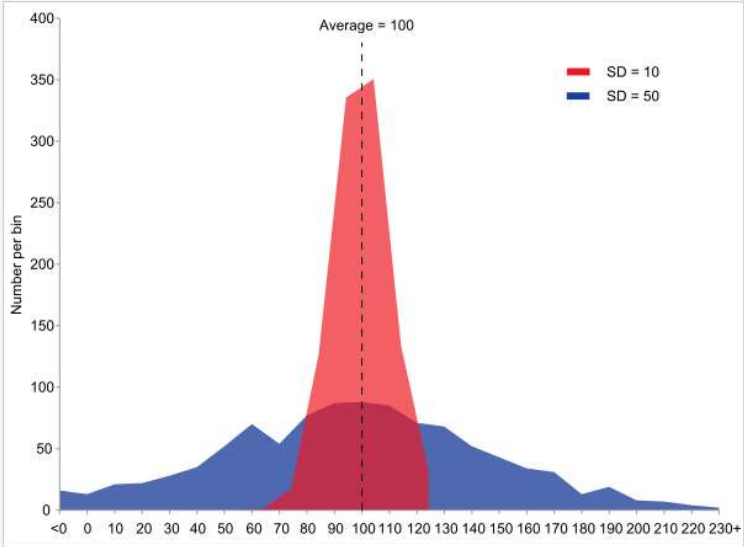


sample standard deviation tells us about spred of dataset around its mean





A large standard deviation indicates that the data points can spread far from the mean and a small standard deviation indicates that they are clustered closely around the mean.



Example of samples from two populations with the same mean but different standard deviations. Red population has mean 100 and SD 10; blue population has mean 100 and SD 50.

For example, each of the three populations {0, 0, 14, 14}, {0, 6, 8, 14} and {6, 6, 8, 8} has a mean of 7. Their standard deviations are 7, 5, and 1, respectively. The third population has a much smaller standard deviation than the other two because its values are all close to 7. These standard deviations have the same units as the data points themselves. If, for instance, the

variance measures variability from the average or mean. It is calculated by taking the differences between each number in the data set and the mean

$$\sigma^2 = \frac{\sum (x - \mu)^2}{N}$$ Population Variance

$$s^2 = \frac{\sum (x - \bar{x})^2}{n - 1}$$ Sample Variance

covariance is a measure of the joint variability of two random variables

---

Population Covariance Formula

$$Cov(x,y) = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{N}$$

Sample Covariance

$$Cov(x,y) = \frac{\Sigma(x_i - \bar{x})(y_i - y)}{N-1}$$

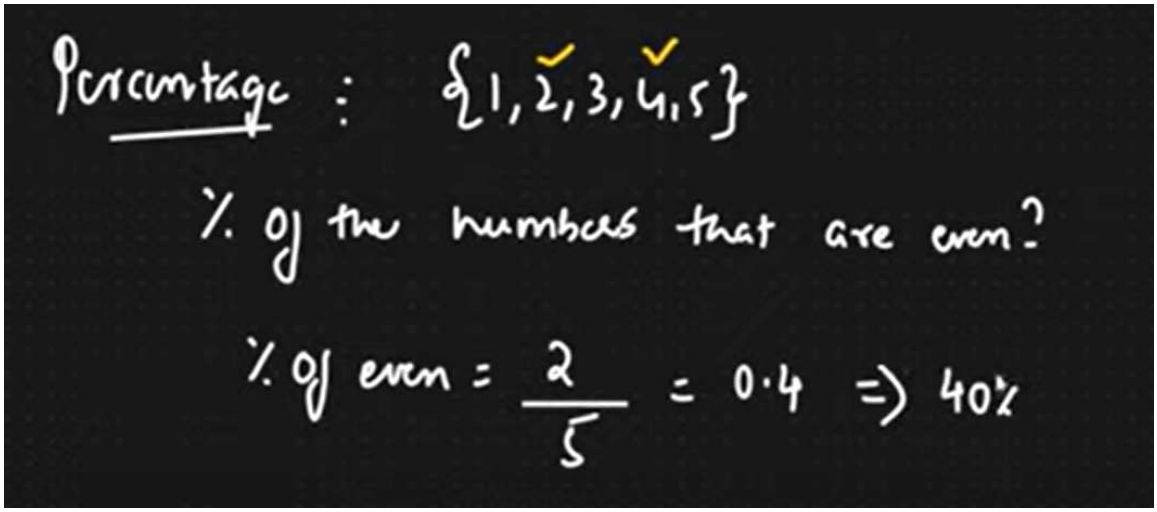
13. Standard Deviation

Standard deviation measures the dispersion of a dataset relative to its mean.

It is calculated as the square root of the variance.

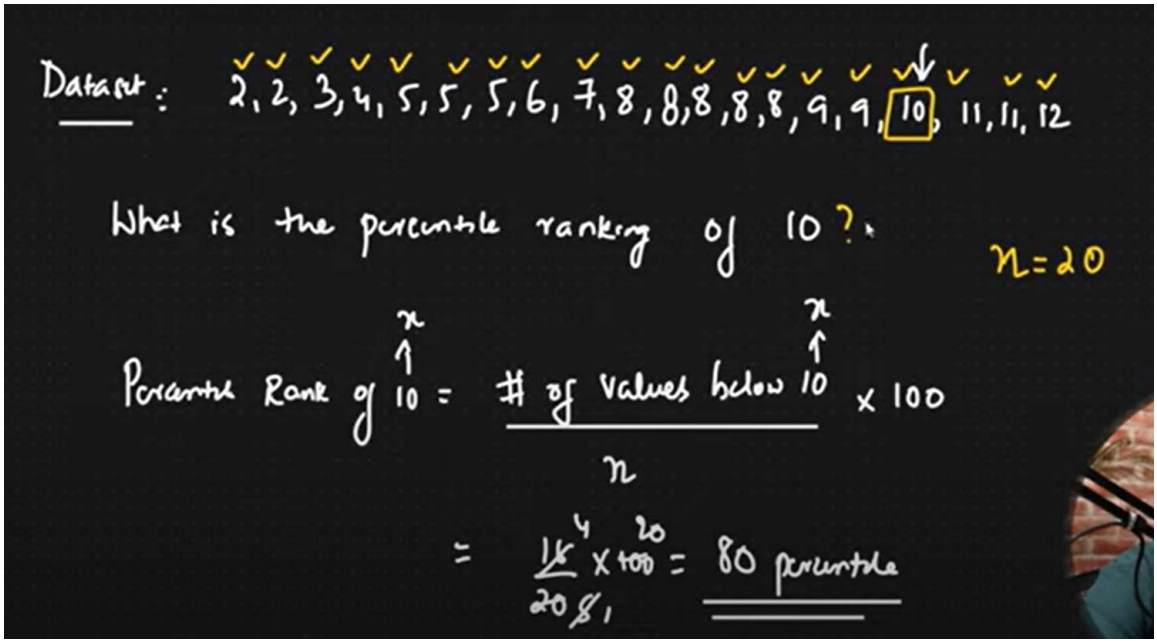
$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N}}$$

Tutorial 6 : Percentile & Quartile

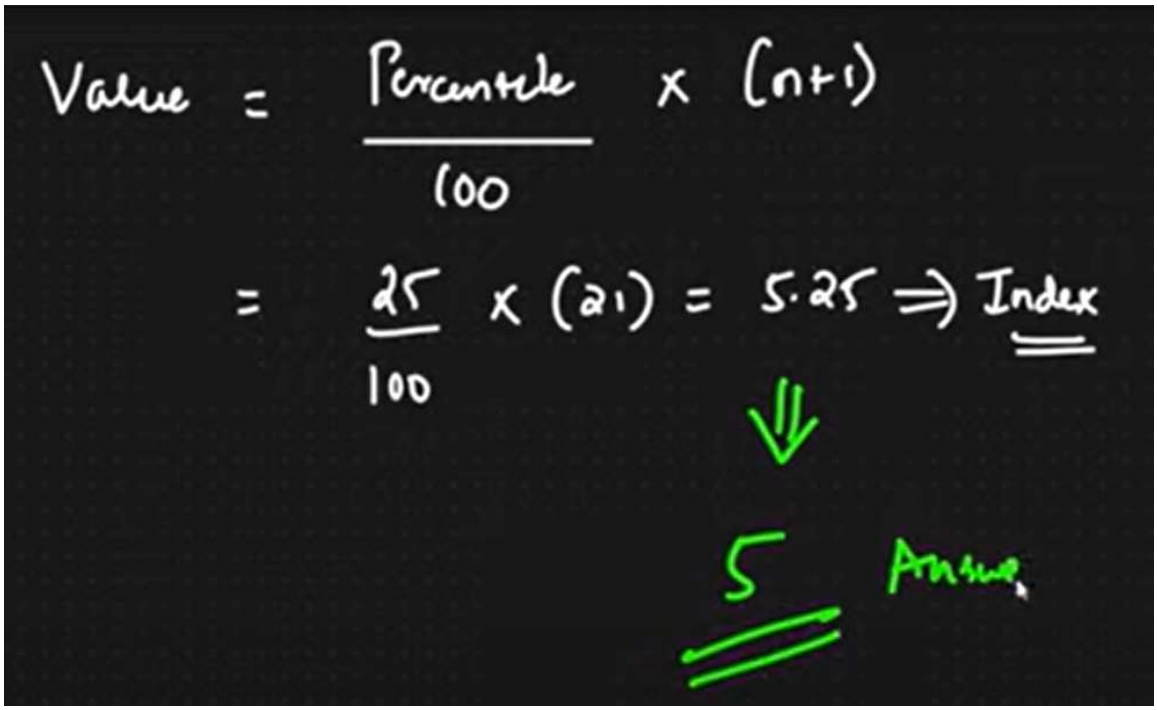


Percentile : A percentile is a value below which a certain percentage of observation lies.

95 percentile : the person has got better marks than 95% of the entire students.



for example how many values are below 10? 80 percentile : 10 percentile is greater than 80%



what value exsits at percentile rank of 25? value = percentile/100 * n+1 = 25/100 * 21 = 5.25 (index) = 5

Quartile :

```
25 Percentile : 1st Quartile (Q1)
75 Percentile : 3rd Quartile (Q3)

IQR = Q3 - Q1
```

with the help of box plot we can visualize the 5numbe summery

Tutorial 7 : How to Construct a Box Plot for Outliers (Five Number Summery)

```
1.Minimum
2.First Quartile (25%) Q1
3.Median
4.Third Quartile (75%) Q3
5.Maximum
```

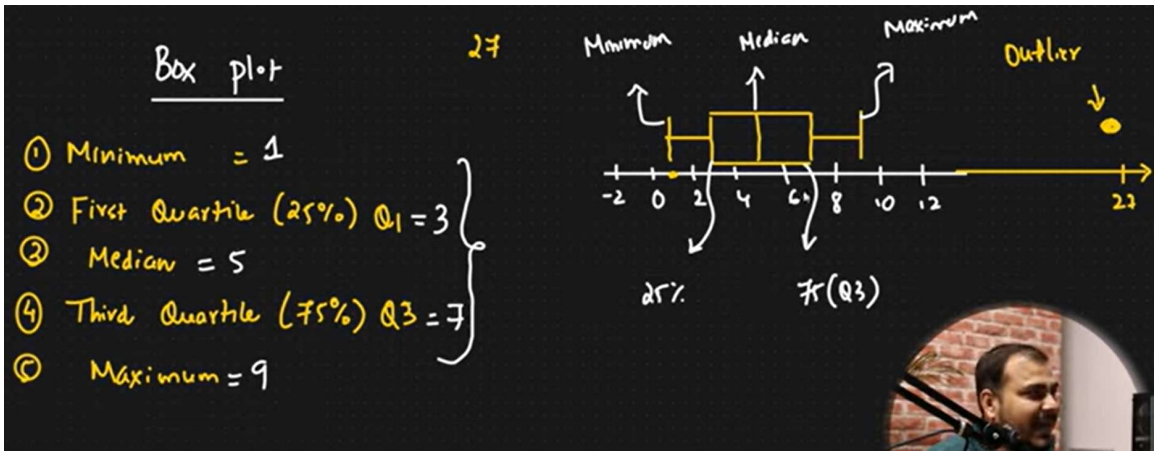
Removing the outliers : Box plot used to detect outliers {1,2,2,2,3,3,4,5,5,5,6,6,6,6,7,8,8,9,27}

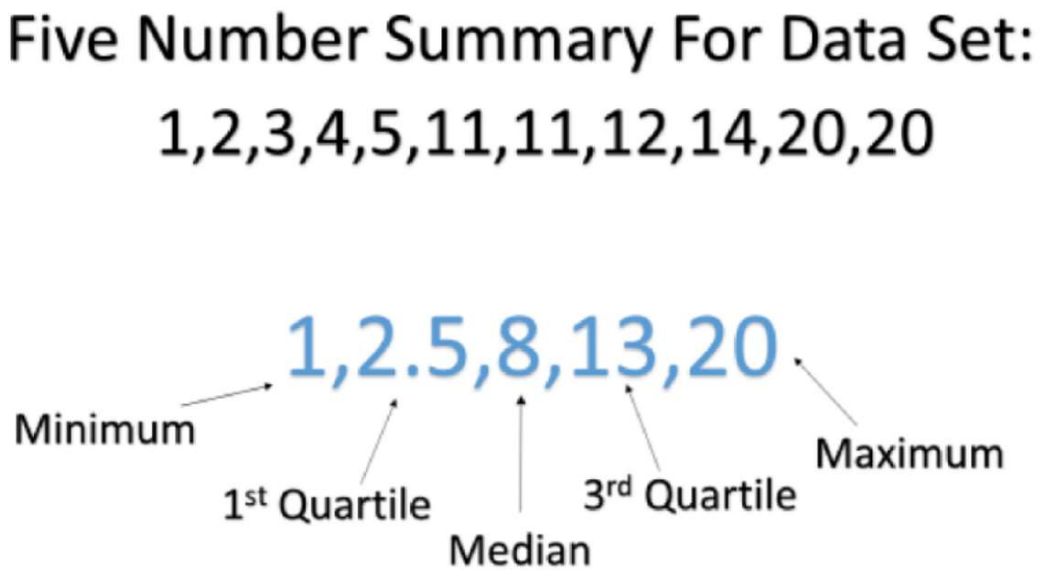
```
lower fense ------------ higher fense
(Q1-1.5(IQR))            (Q3+1.5(IQR))
```

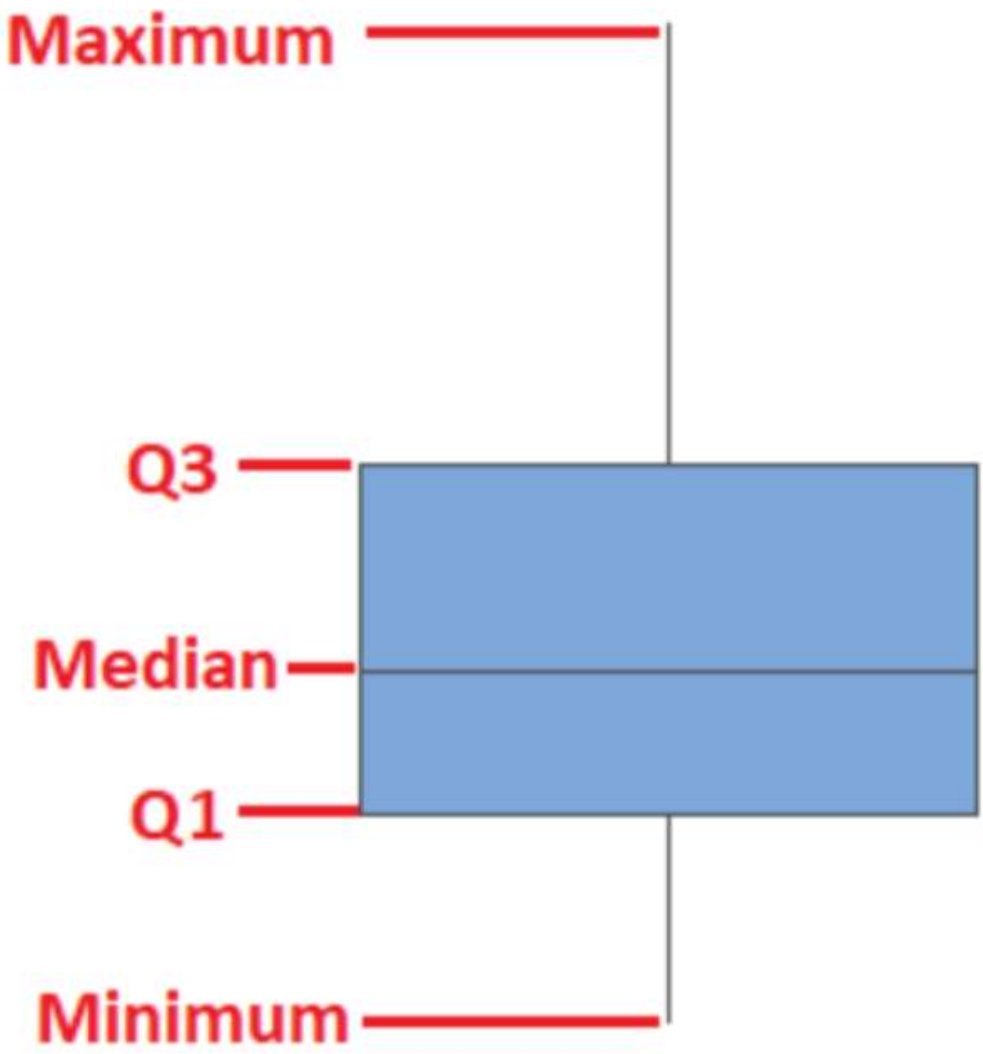
Q1 = 25/100 X 19+1 = 5th(INDEX) = 3 Q3 = 75/100 X 19+1 = 15th(INDEX) = 7 IQR = Q3 - Q1 = 7 - 3 = 4 LF = 3-1.5(4) = -3 HF = 3+1.5(4) = 13

Boxplot

1.Minimum = 1
2.First Quartile (25%) Q1 = 3
3.Median = 5
4.Third Quartile (75%) Q3 = 7
5.Maximum = 9





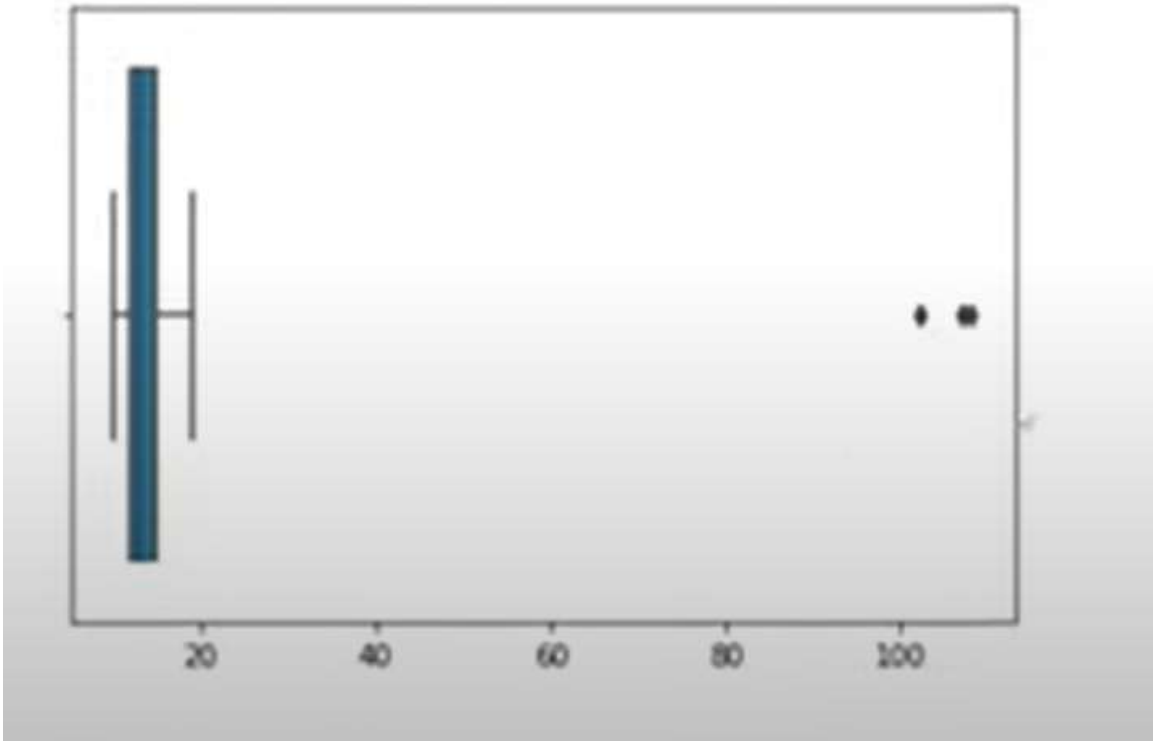


Tutorial 8 : Finding Outliers Using Python

1. Z-Score
2. Boxplot

```
sns.boxplot(dataset)
```

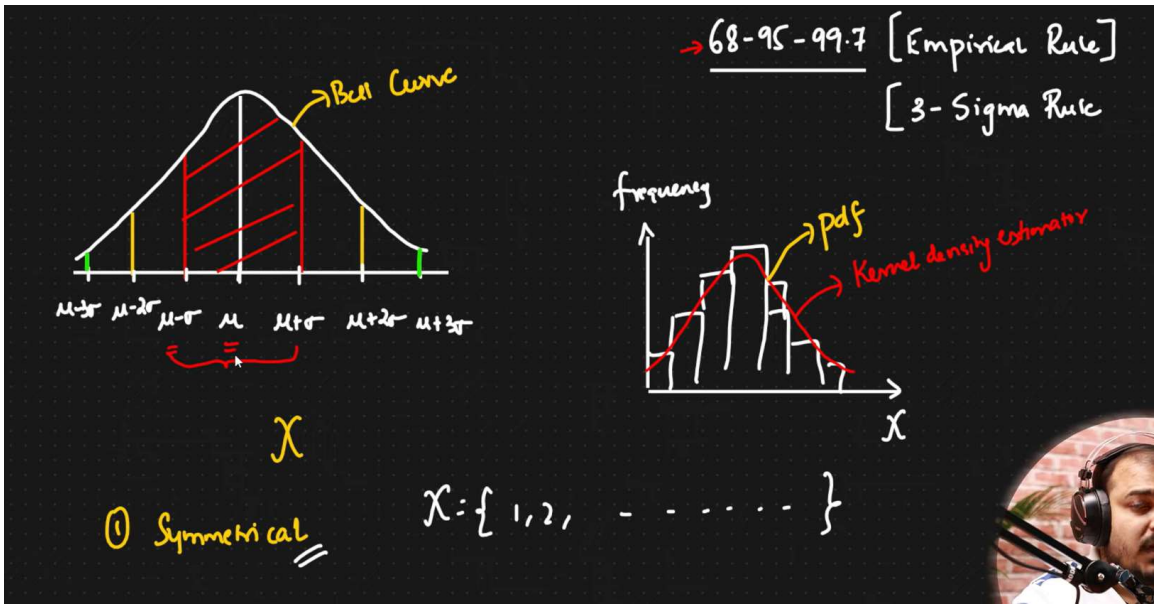
```
/usr/local/lib/python3.7/dist-packages/seaborn/_dec
  FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe6bc
```



Tutorial 9 : Normal Distribution and Its Empirical Formula

The normal distribution is a continuous probability distribution that is symmetrical around its mean (data is equally distributed right of mean and left of mean)

symmetrical > Symmetrical meance 50% data points left side & 50% data points right side of mean



Empirical rule

The empirical rule predicts that 68% of observations falls within the first standard deviation (μ ± σ),

95% within the first two standard deviations (μ ± 2σ),

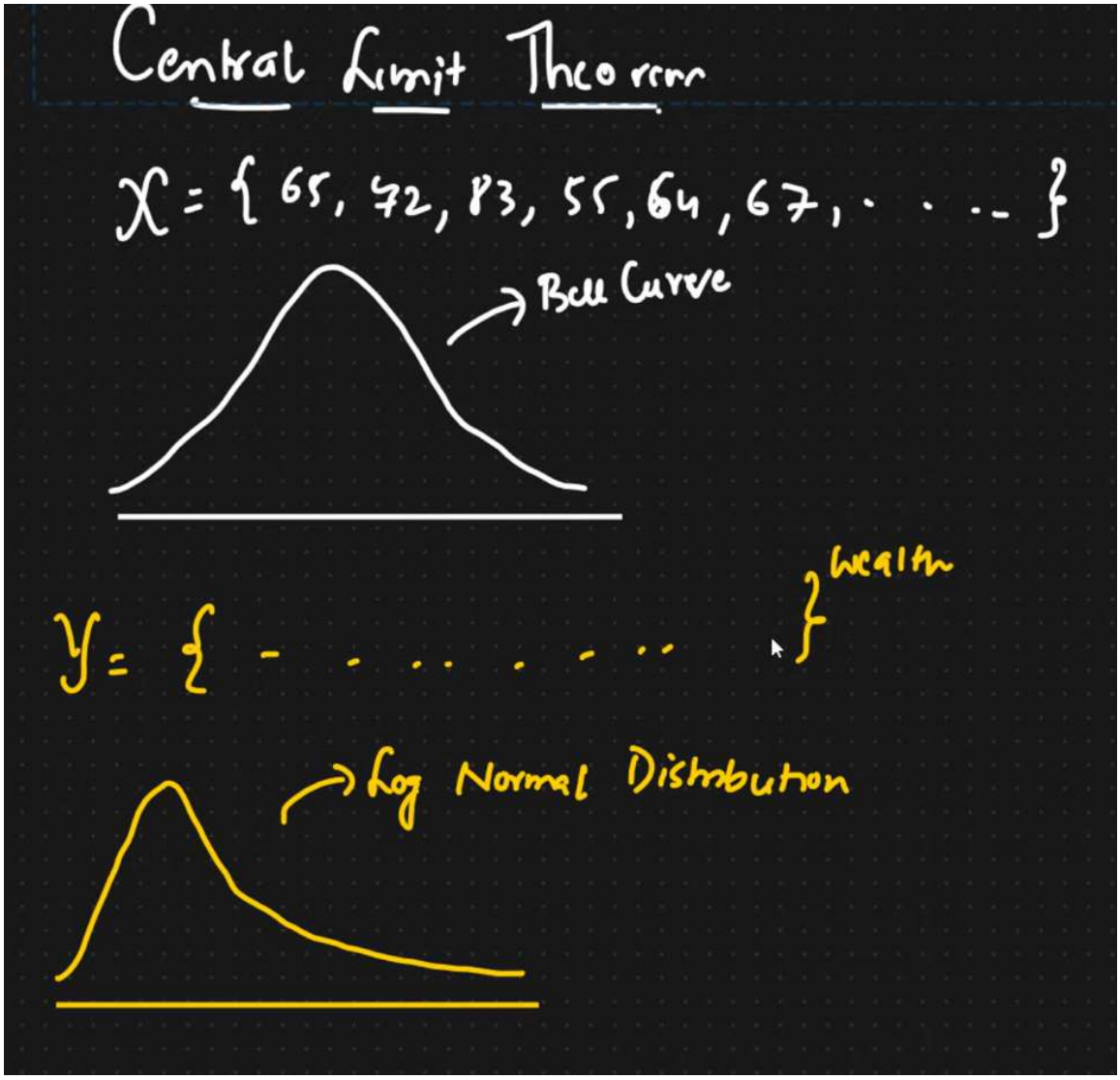
and 99.7% within the first three standard deviations (μ ± 3σ).

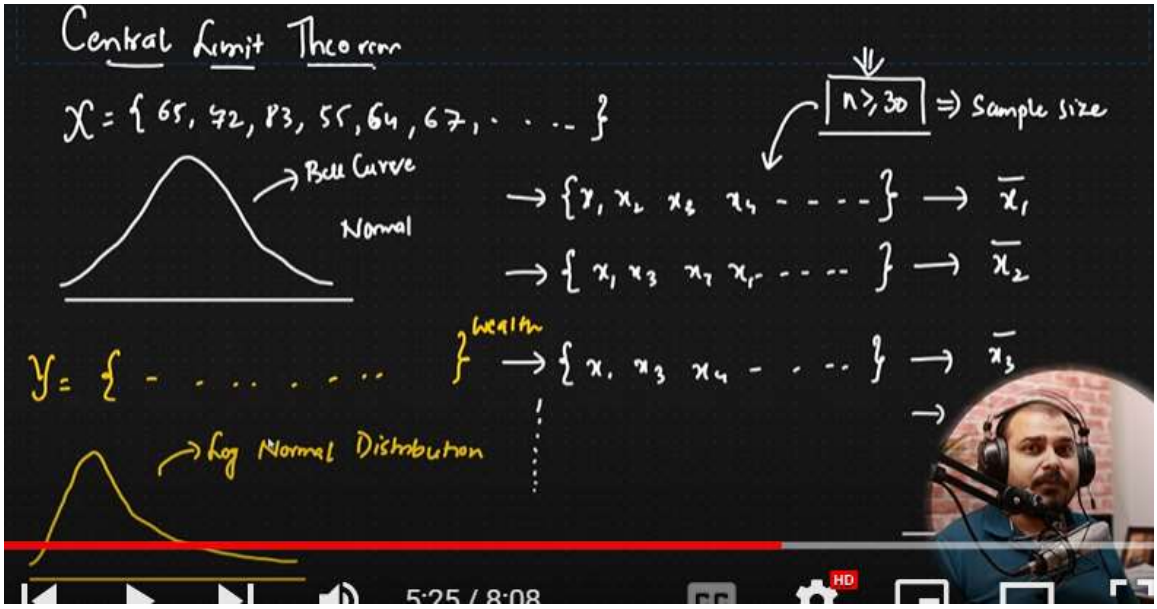
Example : Iris Data set follows the normal distribution

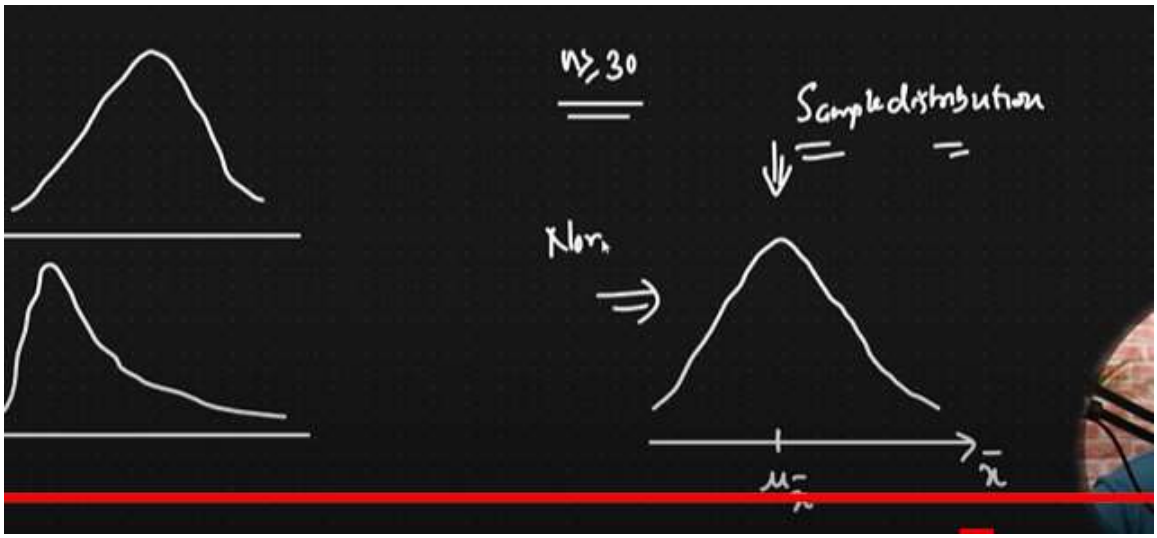
Tutorial 10 : Central Limit Theorem (CLT)

The central limit theorem (CLT) states that the distribution of sample means approximates a normal distribution as the sample size gets larger or even if the original variables themselves are not normally distributed.

1. Sample sizes equal to or greater than 30 are often considered sufficient for the CLT to hold.
2. A key aspect of CLT is that the average of the sample means and standard deviations will equal the population mean and standard deviation.
3. A sufficiently large sample size can predict the characteristics of a population more accurately


Central Limit Theorem
X = { 65, 72, 83, 55, 64, 67, ... }
Bell Curve
Y = { - - . . . . - . . }
Wealth
Log Normal Distribution


Central Limit Theorem
X = { 65, 72, 83, 55, 64, 67, ... }
n ≥ 30
⇒ Sample size
Bell Curve
Normal
→ { x, x2 x3 x4 - - - - } → x̄1
→ { x1 x3 x7 x1- - - - } → x̄2
Y = { - . . . . - . . }
Wealth
→ { x1 x3 x4 - . . . } → x̄3
Log Normal Distribution
5:25 / 8:08
HD

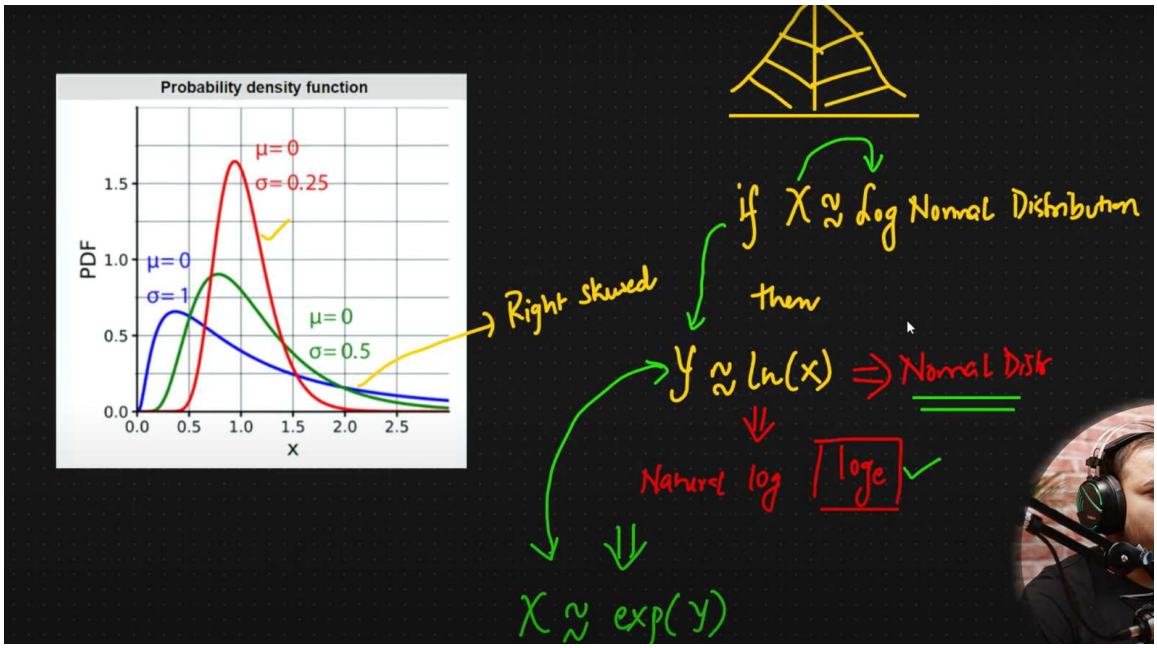


Tutorial 11 : Log Normal Distribution

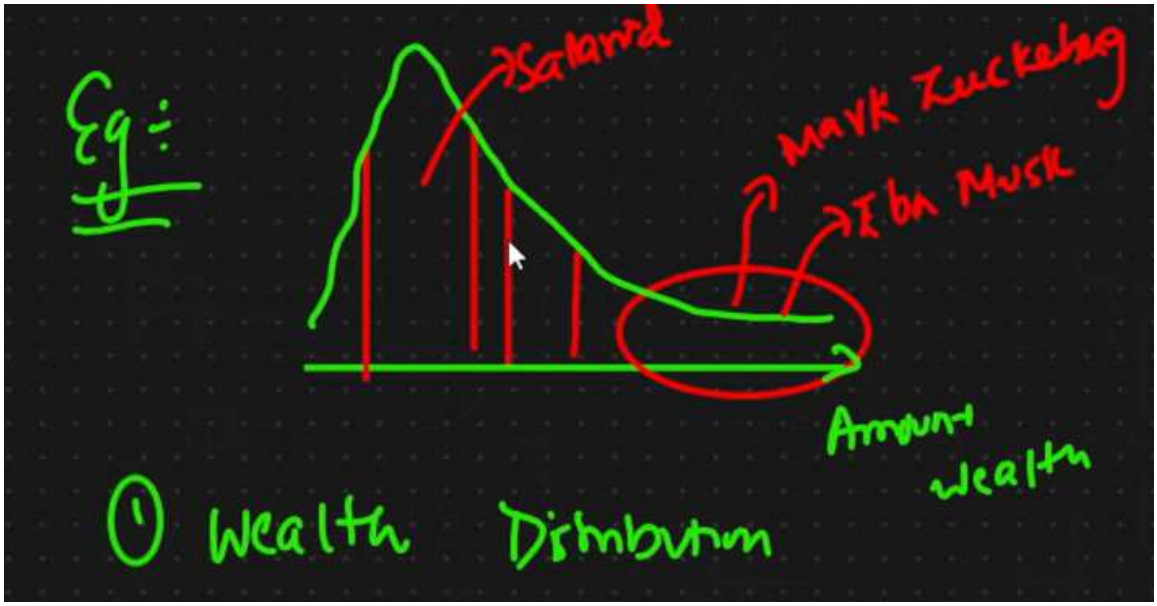
A log-normal (or lognormal) distribution is a continuous probability distribution of a random variable whose logarithm is normally distributed.

if x belongs to log normal distribution then y = log(x) has normal distribution

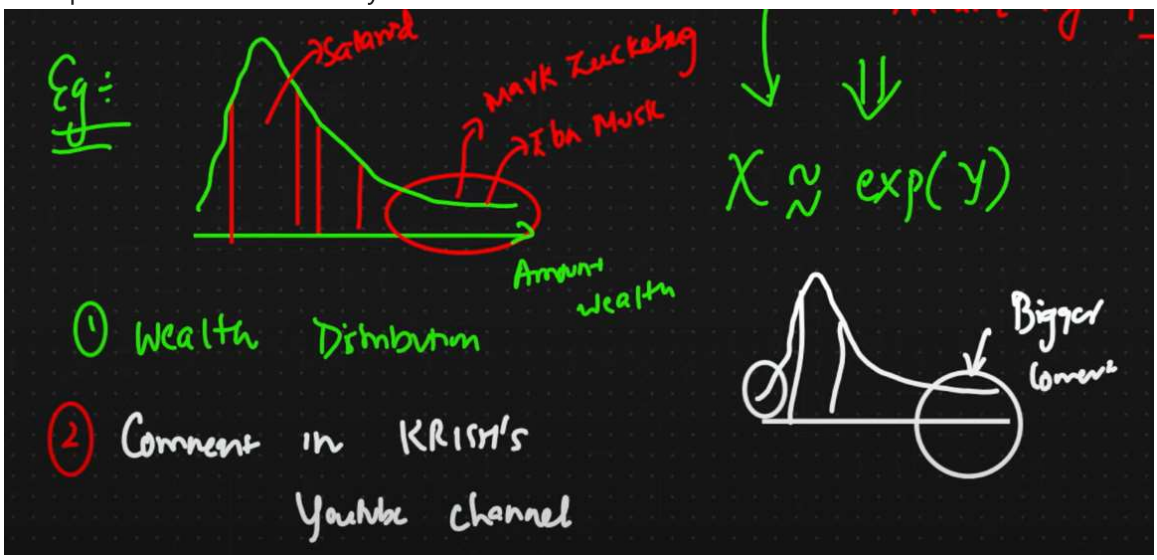
if y belongs to log normal distribution then x = exp(y) has normal distribution



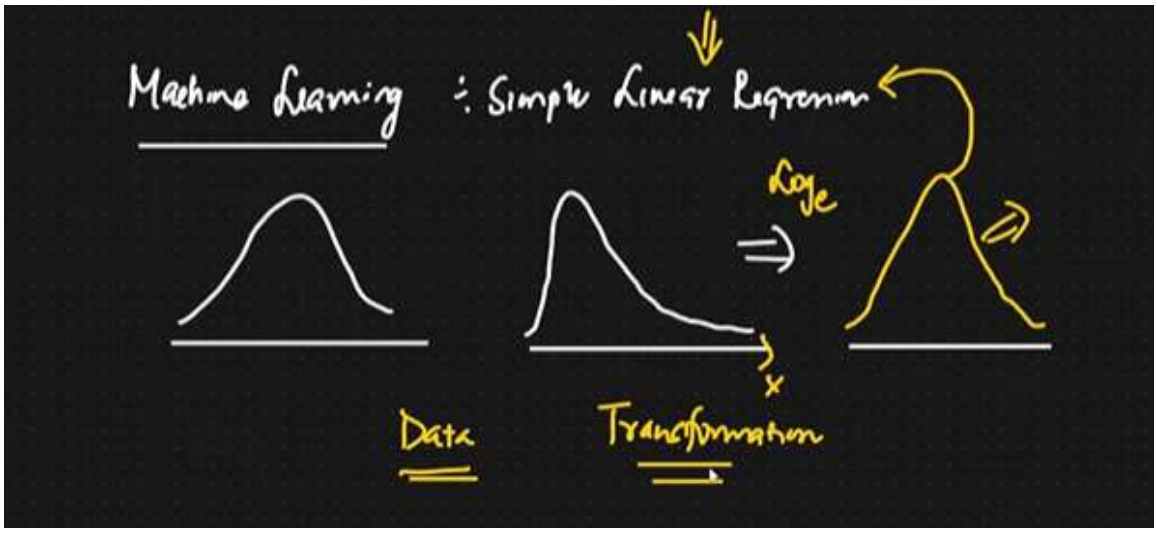
Example : Wealth Distribution



Example : Comments in krish youtube channel

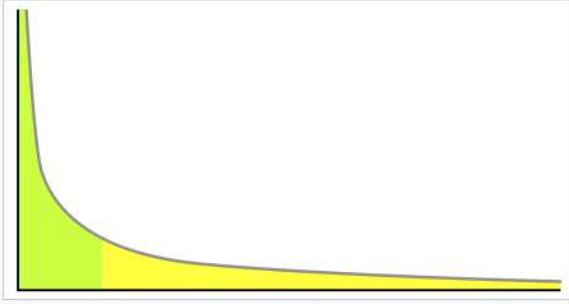


WHere we are using log normal distribution : Machine Learning Data Transformation



Tutorial 12 : Power Law Distribution

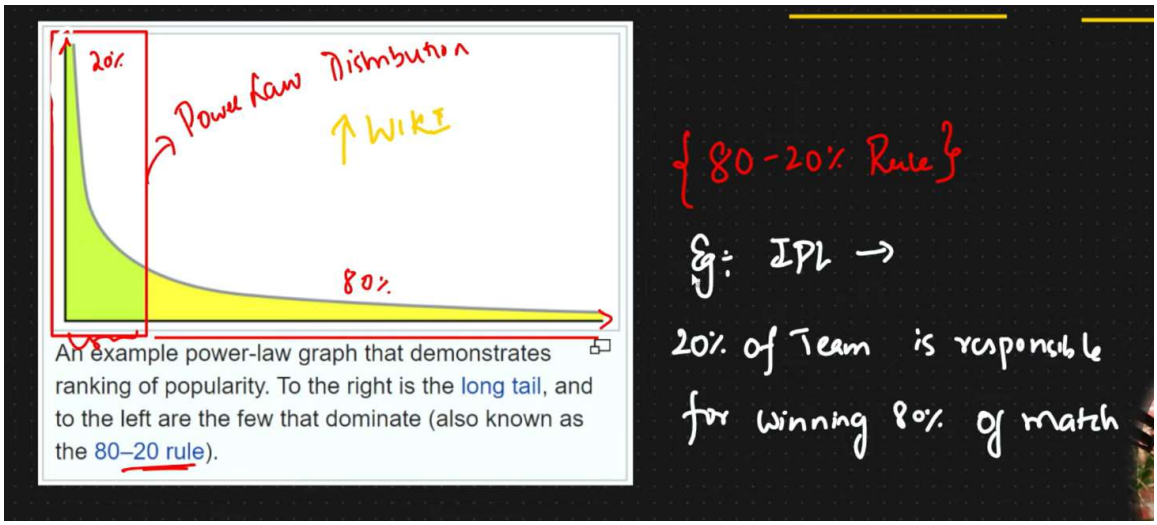
In statistics, a **power law** is a functional relationship between two quantities, where a relative change in one quantity results in a relative change in the other quantity proportional to a power of the change, independent of the initial size of those quantities: one quantity varies as a power of another. For instance, considering the area of a square in terms of the length of its side, if the length is doubled, the area is multiplied by a factor of four.[1]



An example power-law graph that demonstrates ranking of popularity. To the right is the long tail, and to the left are the few that dominate (also known as the 80–20 rule).

Example :

1. IPL 20% Team is responsible for winning 80% of match
2. 80% of wealth is responsible for 20% of the total population
3. 80% of the total oil is with 20% of the nation

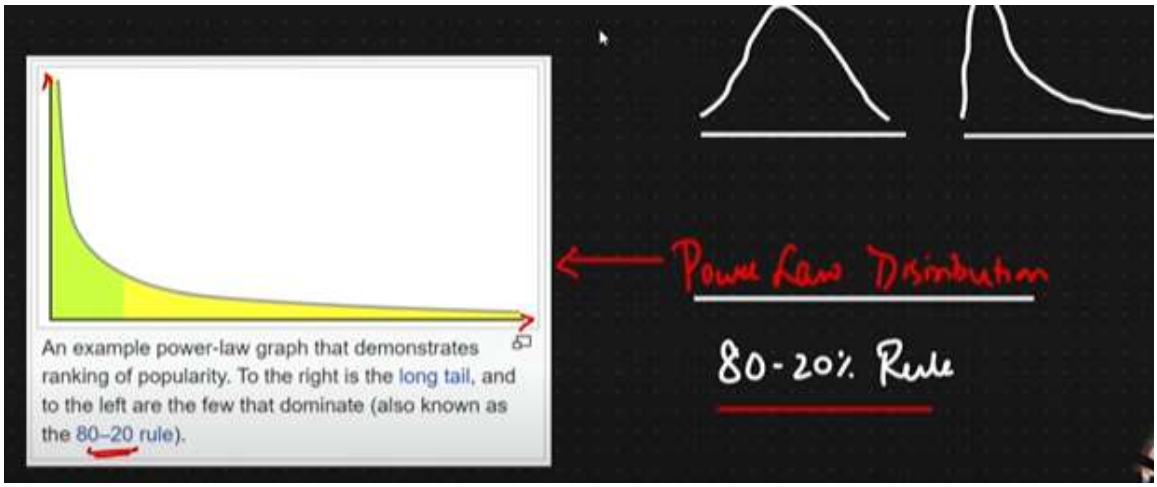


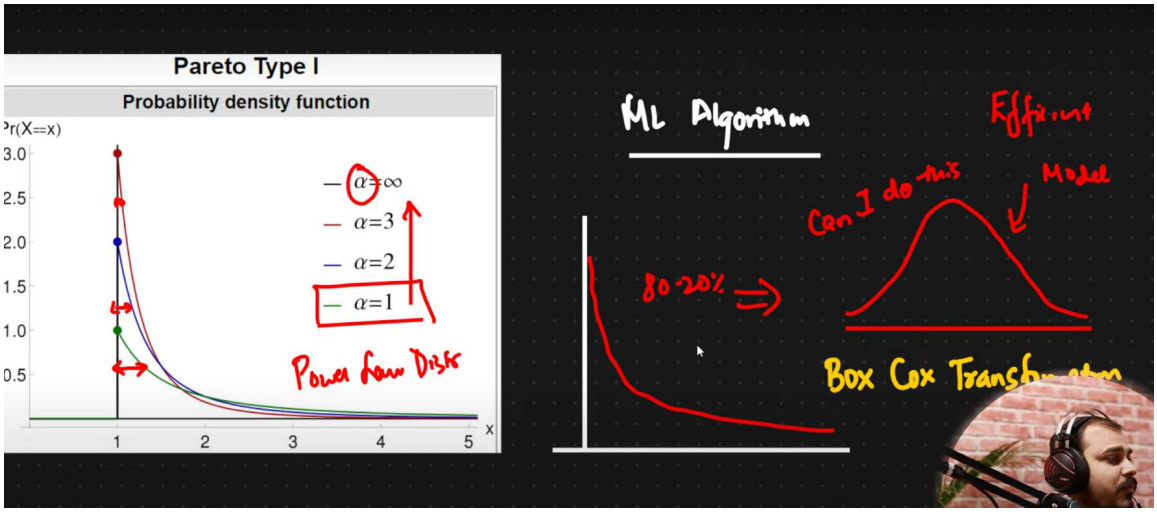
Power log distribution can be transformed by using "box cox transform" technique

Tutorial 13 : Pareto Distribution

It is a example of power law distribution.

Pareto distribution is not noraml or gausian distribution it is a non gausin distribution and it folllows the power law distribution.





Example :

1. 20% of peoducts in AMAZON is resposible of 80% of sales
2. 20% of diffect solves the 80% of upcoming defect
3. 20% of team is resposible in delevelable 80% of project

Interview question : Relation between log normal and pareto distribution

The log-normal distribution is a continuous probability distribution of a random variable whose logarithm is normally distributed.

```
if X follows a log-normal distribution,
then the natural logarithm of X, ln(X),
follows a normal distribution.
```

Pareto distribution is also a continuous probability distribution, but it is characterized by a heavy right tail and follows the power law distribution with follows the 80-20 rule.

```
Pareto(x; α, xₘ)

random variable (x)
```

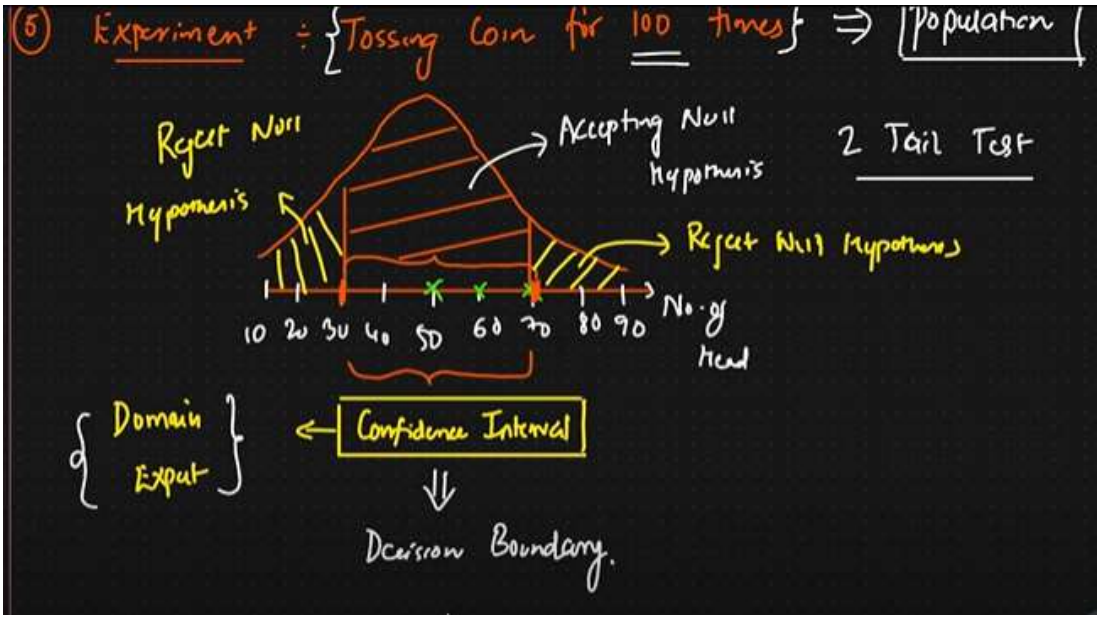
```
shape parameter (α) : determines the tail behavior
scale parameter (xₘ).
```

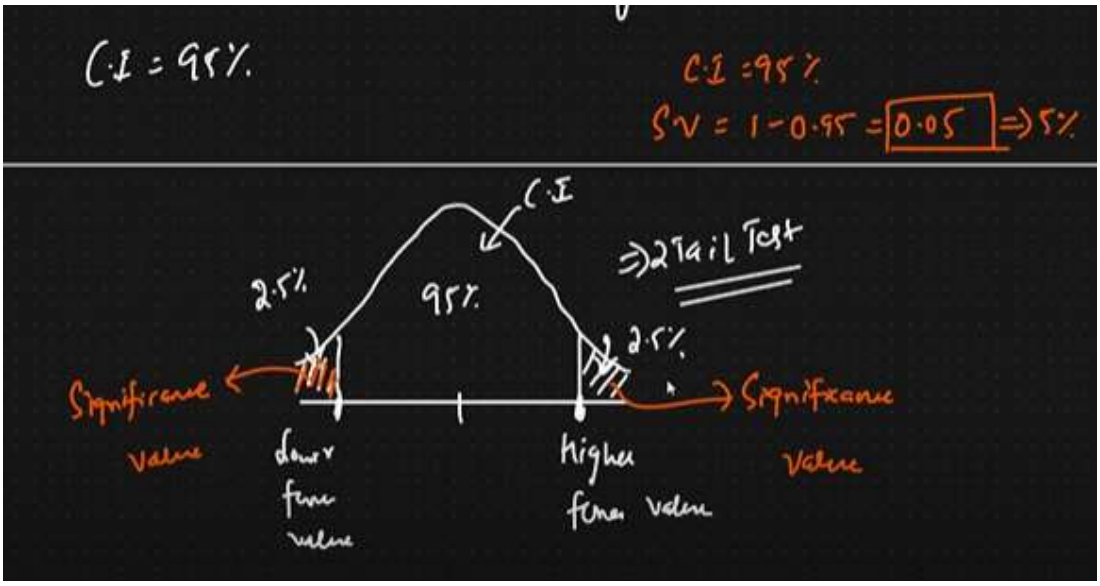
Tutorial 14 : What is Hypothesis Testing

Hypothesis Testing : Hypothesis testing is a statistical procedure used to draw conclusions about a population based on a sample of data.

Steps :

1. Null Hypothesis(H0) : Coin is fair
2. Alternate Hypothesis(H1) : Coin is not fair
3. Experiment : Tossing coin for 100 times and my no. heads = 50 Tossing coin for 100 times and my no. heads = 60 Tossing coin for 100 times and my no. heads = 70 so i will assume coin is not fair



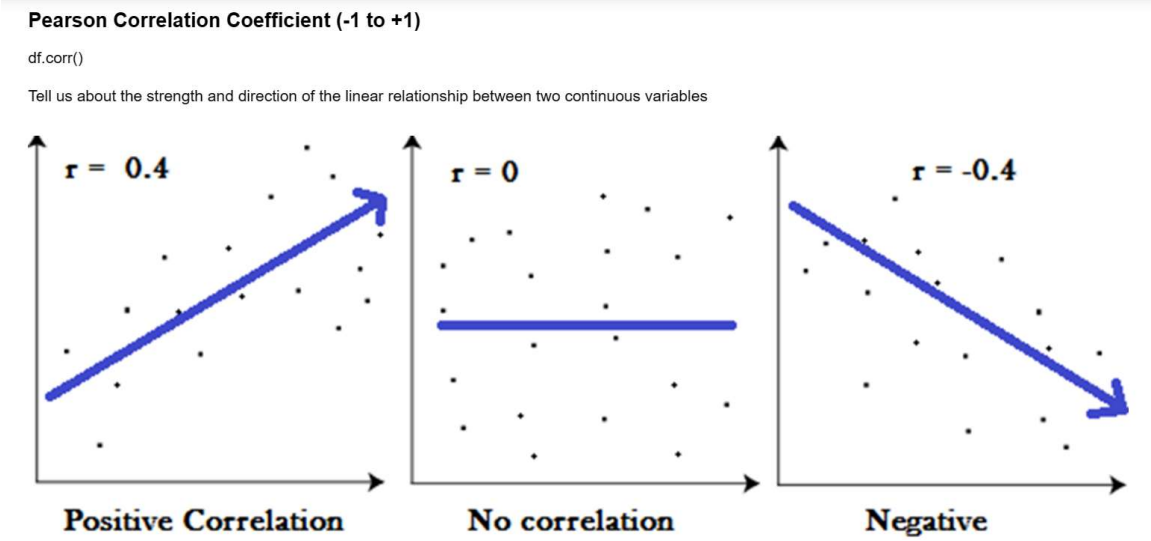


15. Correlation coefficients

Correlation coefficients are used to measure how strong a relationship is between two variables. There are several types of correlation coefficient, but the most popular is Pearson's. Pearson's correlation

---

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}}$$



The Pearson correlation coefficient is represented by the symbol "r" and takes values between -1 and 1. The value of r indicates the degree and direction of the linear relationship between the variables:

If r = 1, it indicates a perfect positive linear relationship, meaning that as one variable increases, the other variable increases proportionally.

If r = -1, it indicates a perfect negative linear relationship, meaning that as one variable increases, the other variable decreases proportionally.

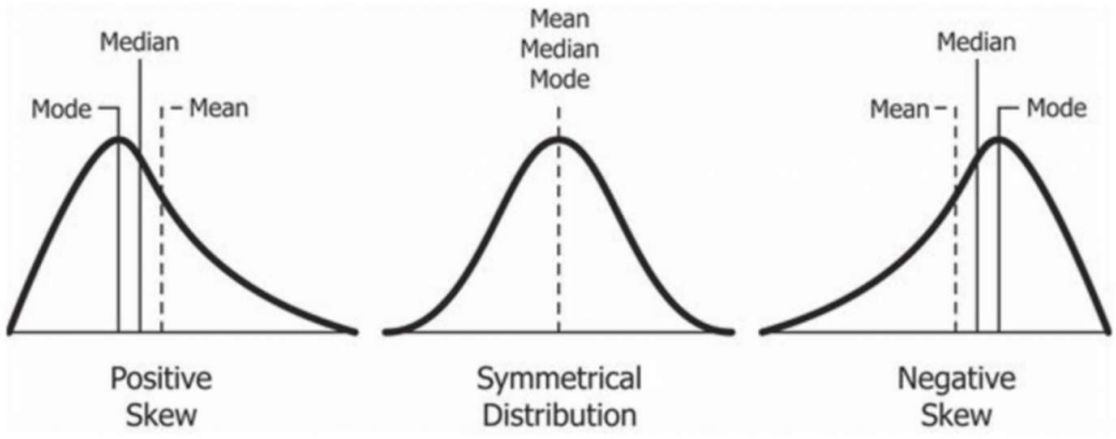
If r = 0, it indicates no linear relationship or correlation between the variables.

16. Skewness

skewness is the measure of how much the probability distribution of a random variable deviates from the normal distribution.

Positively Skewed Distribution : A positively skewed distribution is the distribution with the tail on its right side. The value of skewness for a positively skewed distribution is greater than zero

Negatively Skewed Distribution : Negatively skewed distribution is the distribution with the tail on its left side. The value of skewness for a negatively skewed distribution is less than zero.



17. Evaluation / Performance Matrix

Evaluation matrix quantifies the performance of a predicted model or measure the quality of the model.

There are various evaluation matrixs mentioned below,

1. Accuracy
2. Confusion Matrix
3. Log-loss function
4. Precision & Recall
5. f1-score
6. AUC-ROC Curve

1. Accuracy

The accuracy is the ratio between number of correct predictions to the total number of predictions

Accuracy Score = TP + TN/ TP+TN+FP+FN

2. Confusion Matrix

A confusion matrix is a performance evaluation tool used in machine learning and statistics to assess the accuracy of a classification model.

or

The confusion matrix is a square matrix that shows the performance of a classification model by comparing the predicted class labels with the actual class labels of the data

The confusion matrix is a table that organizes the model's predictions into four categories:

1.True Positives (TP): Actual Positive and predicted also Positive

2.False Positives (FP) : Actual Negative and predicted Postive >> **Type 1 Error**

3.True Negatives (TN): Actual Negative and predicted also Negative

4.False Negatives (FN): Actual Postive and predicted Negative >> **Type 2 Error**

Predicted Positive Predicted Negative Actual Positive TP FN Actual Negative FP TN

3. Log_loss Function

Log_loss function indicates how close the prediction probability is to the corresponding actual value

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

```
y >> target variable which ranges in between 0 & 1
p >> is our predicted probability

case 1 : y=1 p=1 >> correct prediction
case 2 : y=0 p=0 >> correct prediction
case 1 : y=1 p=0 >> Incorrect prediction
case 1 : y=0 p=1 >> Incorrect prediction
```

4. Precision & Recall

Precision and recall are two important performance metrics used to evaluate the effectiveness of a classification model, particularly in binary classification tasks.

Precision : Precision is the proportion of true positive predictions **among all positive predictions** made by the model. It focuses on the accuracy of positive predictions.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{.....}}$$

```
                (True Positives (TP) + False Positives (FP))
```

e.g important in medical cases

Recall : Recall is the proportion of true positive predictions **among all actual positive predictions** made by the model. It focuses on the accuracy of positive predictions.

```
                    True Positives (TP)
Recall = ..........................................
         (True Positives (TP) + False Negative (FN))
```

e.g spam folder

5. F1_Score

The F1 score is a metric that combines precision and recall into a single value and is commonly used to evaluate the performance of a classification model,

```
The F1 score is the harmonic mean of precision and recall and is
defined as follows:

                              (Precision * Recall)
   F1 Score = (1 x 1 ) ...................................
                                (Precision + Recall)
```

When both FN & FP are important that time f1 score came into picture

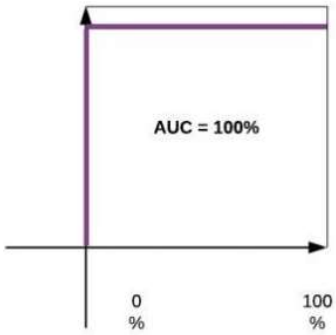
```
f-beta score

b > 0.5 >> FP is important >> Precision (f0.5 score)
b = 2 >> FN is important >> Recall (f2 score)
b = 1 >> FP & FN are important >> Precision & Recall (f1 score)
```

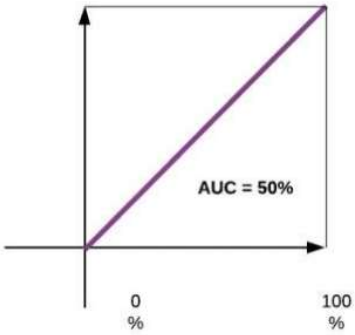
6. AUC - ROC Curve : AUC stands for "Area Under the ROC Curve." ROC stands for "Receiver Operating Characteristic."

The AUC-ROC curve is a plot of TPR (sensitivity) against FPR (1 - specificity) at different threshold settings.

The AUC-ROC curve is a graphical representation of the performance of a binary classification model at various classification thresholds.

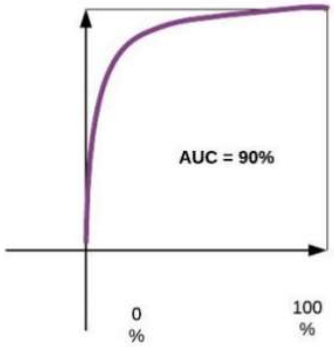
AUC for ROC curves

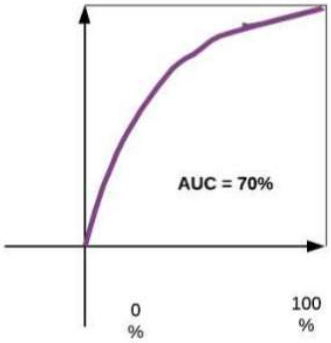




X axis = False Positive Rate

Y axis = True Positive Rate





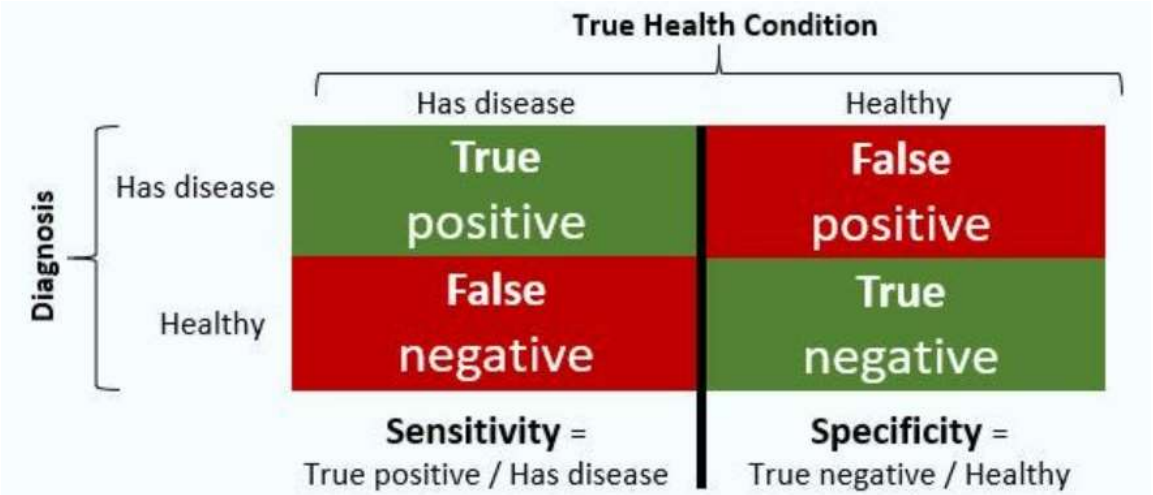
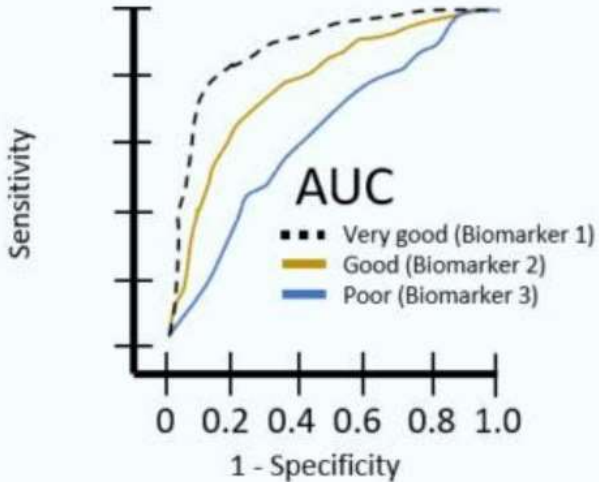


Figure 2. Calculation of senstivity and specificty.



Statistics Practical Implementation

In [8]:
```
ages=[23,24,32,45,12,43,67,45,32,56,32]
import numpy as np
print(np.mean(ages))
print(np.median(ages))
import statistics
statistics.mode(ages)
```

```
37.36363636363637
32.0
```

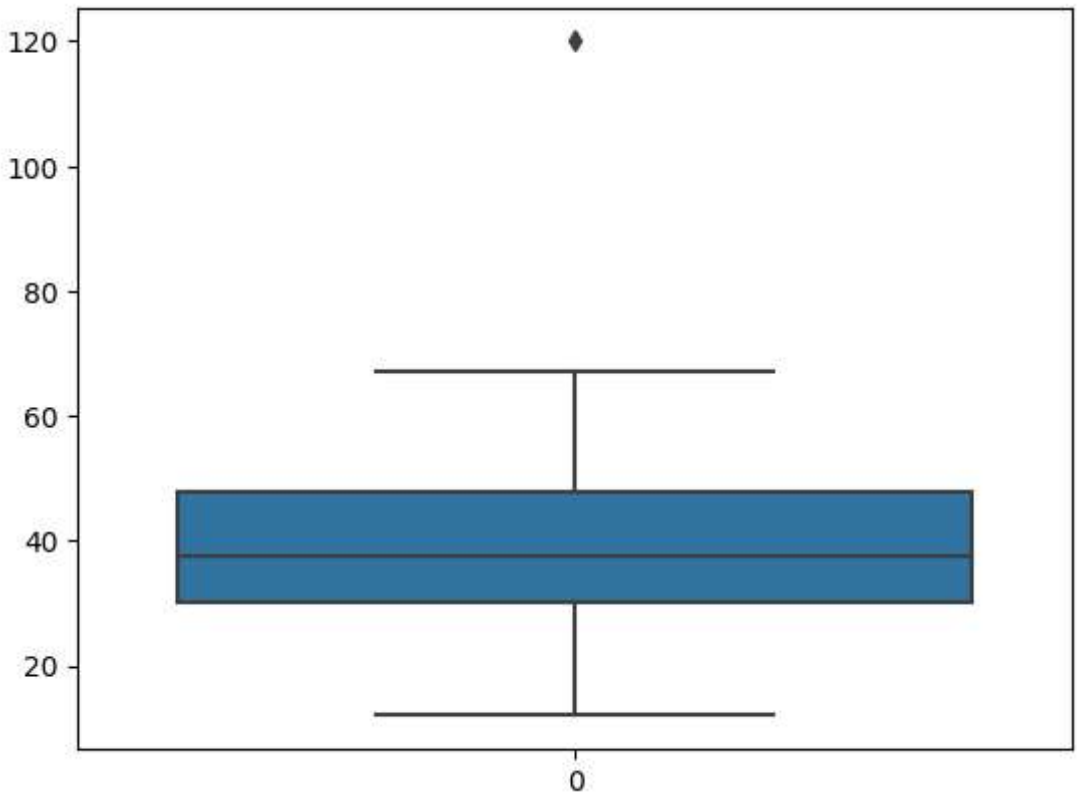
Out[8]: 32

In [10]:
```
ages=[23,24,32,45,12,43,67,45,32,56,32,120] # 120 added
import numpy as np
print(np.mean(ages))
print(np.median(ages))
import statistics
print(statistics.mode(ages)) #
print(statistics.median(ages))
```

```
44.25
37.5
32
37.5
```

In [11]:
```
# how to detect ouliers
import seaborn as sns
sns.boxplot(ages)
```

Out[11]: <Axes: >



In [12]:
```
# 5 number summery
q1,q3=np.percentile(ages,[25,75])
```

In [13]:
```
print(q1)
print(q3)
```

```
30.0
47.75
```

In [14]:
```
# To check outliers [lower fense - higher fence]
IQR = q3-q1
lower_fence =q1- 1.5*(IQR)
higher_fence =q3+ 1.5*(IQR)
print(lower_fence,higher_fence)
```

```
3.375 74.375
```

In [15]:
```
# Measure of Dispersion
# 1. Variance
# 2. Std Deviation
```

In [16]:
```
statistics.variance(ages) # sample varaince
```

Out[16]:
```
795.2954545454545
```

In [17]:
```
np.var(ages,axis=0) # population variation
```

Out[17]:
```
729.0208333333334
```

In [18]:
```
def variance(data):
    n=len(ages)
    # mean of the data
    mean=sum(data)/n # population variance
    # variance
    deviation =[(x-mean)**2 for x in data]
    variance =sum(deviation)/n
    return variance

variance(ages)
```

Out[18]:
```
729.0208333333334
```

In [19]:
```
def variance(data):
    n=len(ages)
    # mean of the data
    mean=sum(data)/n
    # variance
    deviation =[(x-mean)**2 for x in data]
    variance =sum(deviation)/(n-1)  # sample variance
    return variance

variance(ages)
```

Out[19]:
```
795.2954545454545
```

In [20]:
```
statistics.pvariance(ages)
```

Out[20]:
```
729.0208333333334
```

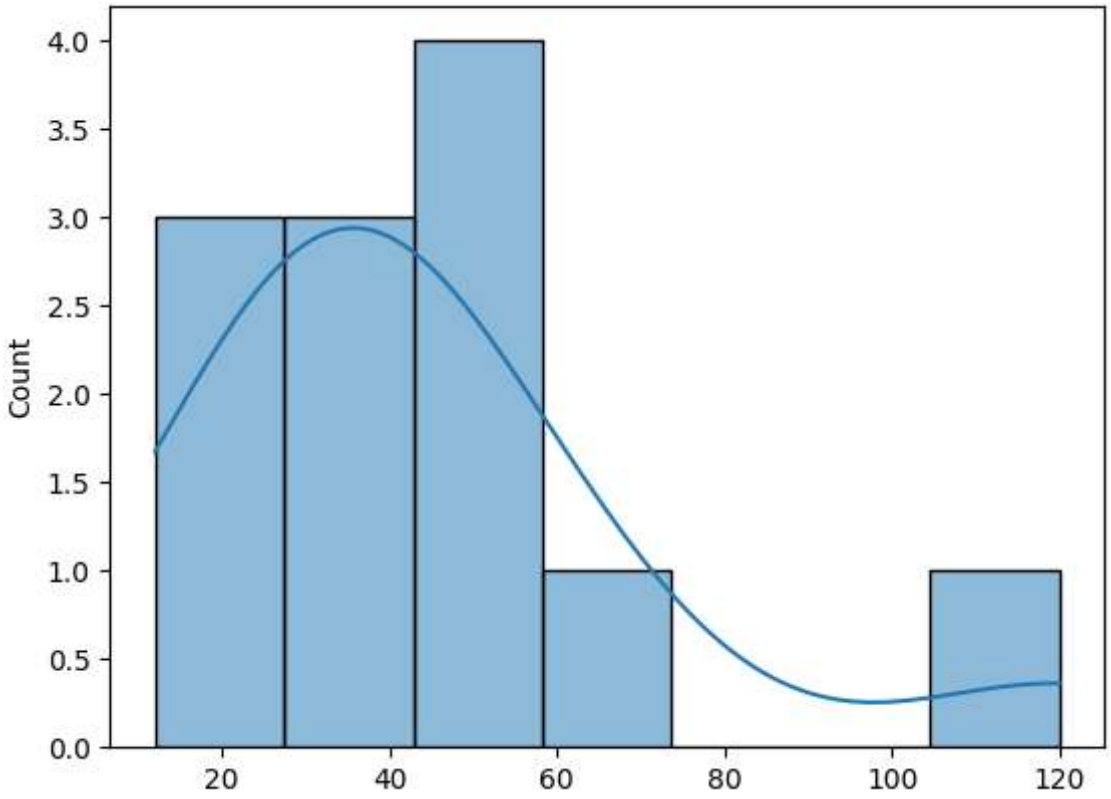
In [21]:
```
import math
math.sqrt(statistics.pvariance(ages))
```

Out[21]:
```
27.000385799712813
```

In [22]:
```
# Histogram and PDF
```

In [24]:
```
import seaborn as sns
sns.histplot(ages,kde=True) # kernal density estimator
```

Out[24]:
```
<Axes: ylabel='Count'>
```



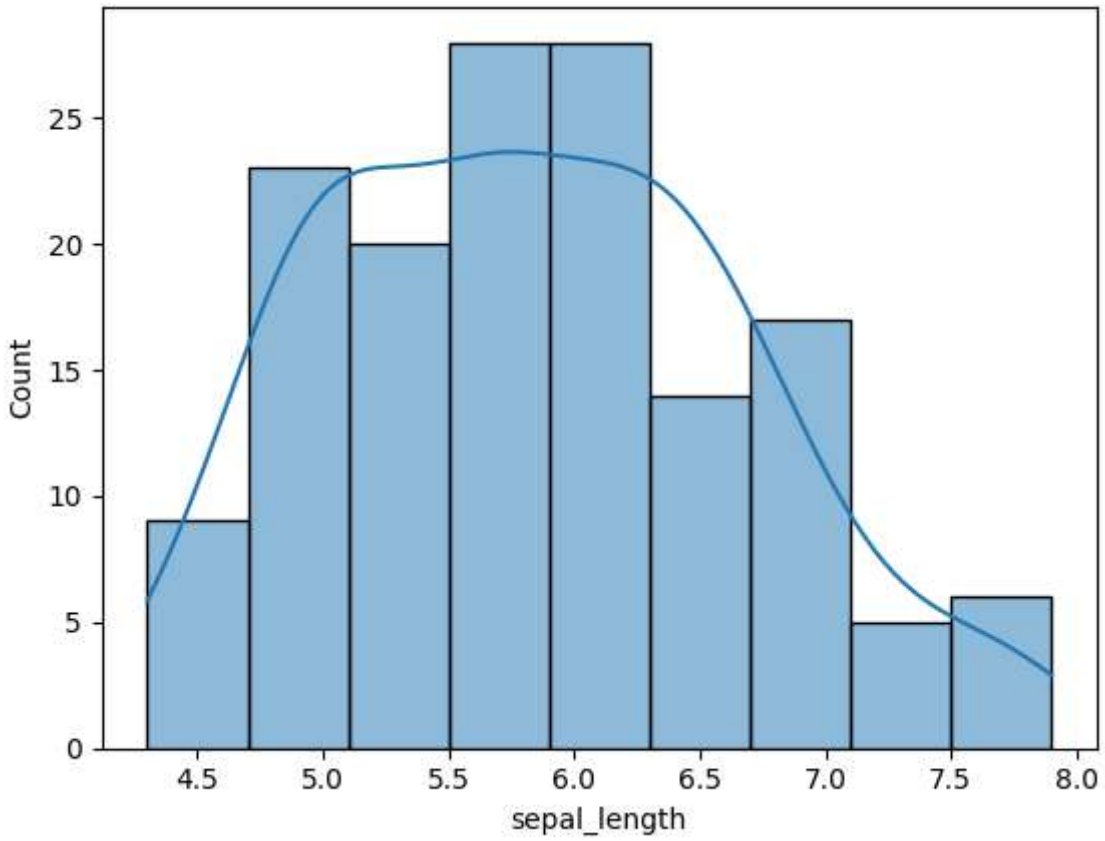
In [26]:
```
df=sns.load_dataset('iris')
df.head()
```

Out[26]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

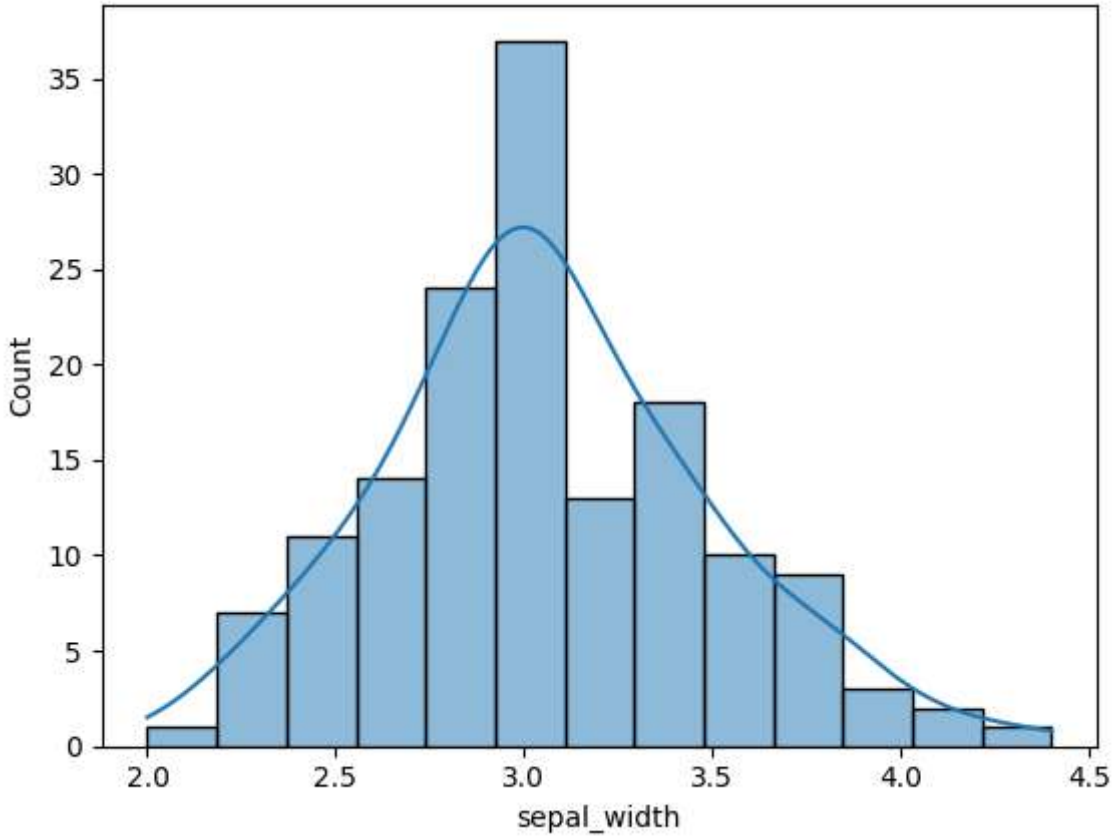
In [27]:
```
sns.histplot(df['sepal_length'],kde=True)
```

Out[27]:
```
<Axes: xlabel='sepal_length', ylabel='Count'>
```



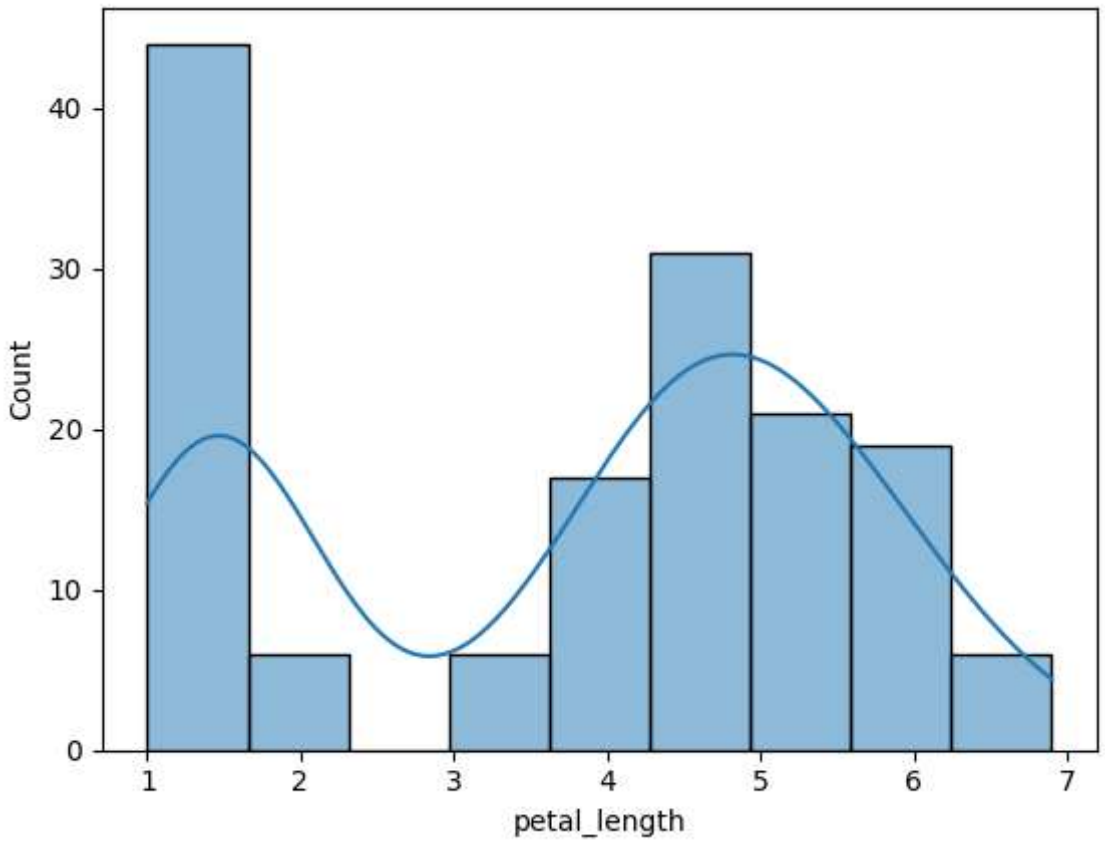
In [28]:
```
sns.histplot(df['sepal_width'],kde=True)
```

Out[28]: <Axes: xlabel='sepal_width', ylabel='Count'>



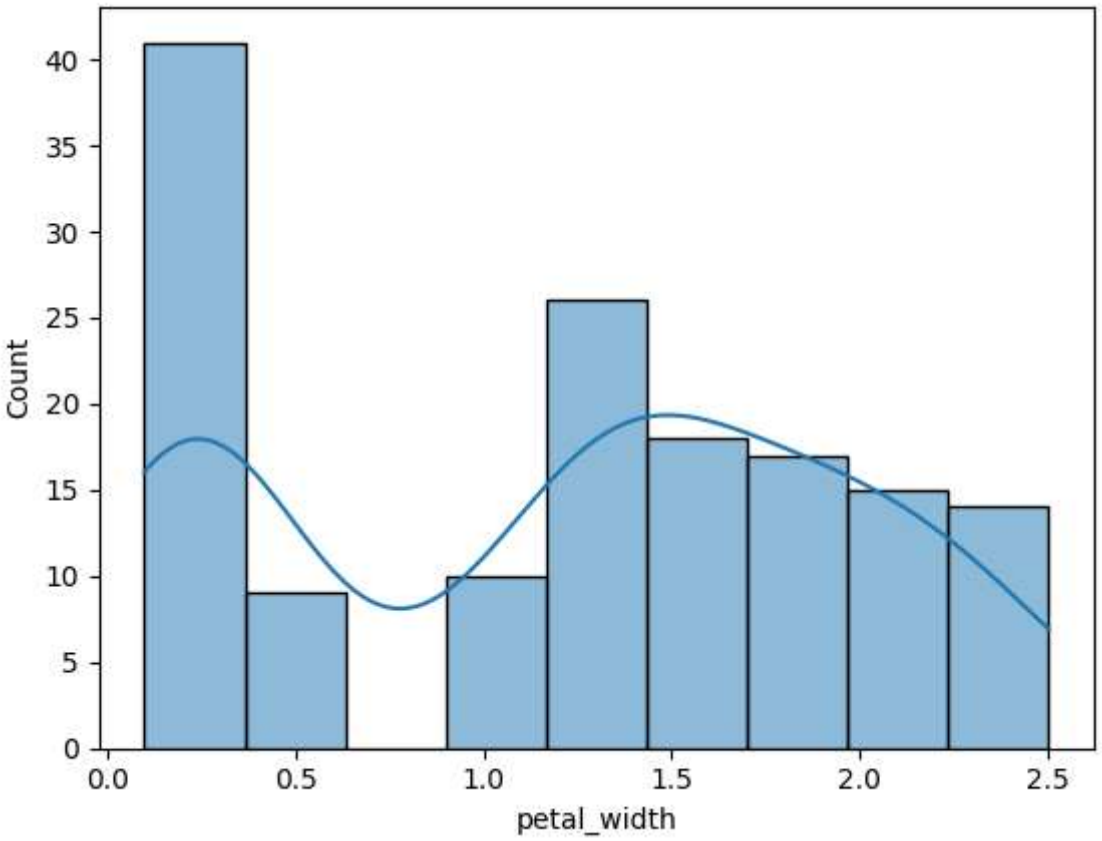
In [29]:
```
sns.histplot(df['petal_length'],kde=True)
```

Out[29]: <Axes: xlabel='petal_length', ylabel='Count'>



In [30]:
```
sns.histplot(df['petal_width'],kde=True)
```

Out[30]: <Axes: xlabel='petal_width', ylabel='Count'>

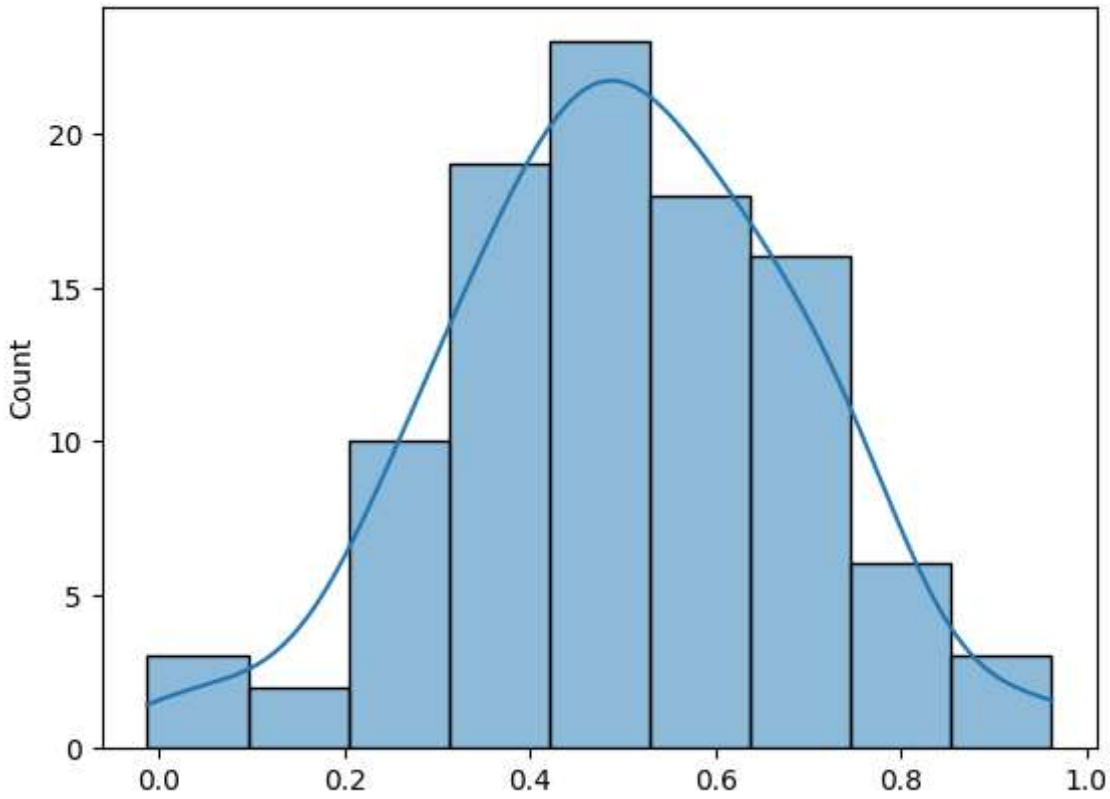


In [31]:
```
# create a normal distribution data
s=np.random.normal(0.5,0.2,100) # mean,std, datapoint
s
```

Out[31]:
```
array([ 0.37653346,  0.38075708,  0.06520672,  0.52408743,  0.57424717,
        0.38313584,  0.71913258,  0.47228634,  0.41298611,  0.64754665,
        0.28948587,  0.45438674,  0.26565306,  0.47970901,  0.7000287 ,
        0.74557389,  0.59108516,  0.35570623,  0.48940942,  0.61818313,
        0.70293293,  0.56337772,  0.40106111,  0.31862212,  0.6371365 ,
        0.44763941,  0.49858126,  0.46793598,  0.72004235,  0.48484315,
        0.60156679,  0.56098168,  0.19894175,  0.39823624,  0.34828535,
        0.66052238,  0.2935562 ,  0.64097448,  0.51111277,  0.62277104,
        0.47331013,  0.58915519,  0.73116969,  0.47901121,  0.58318132,
        0.55283312,  0.47824108,  0.5441097 ,  0.40099676,  0.53071753,
        0.96206704,  0.25975088,  0.46819768,  0.27016269,  0.20555176,
        0.86088436,  0.5140813 ,  0.4173587 ,  0.32417446,  0.96086709,
        0.52241553,  0.64147517, -0.01259536,  0.55905377,  0.66541601,
        0.58386947,  0.23745376,  0.63720331,  0.45846489,  0.52038251,
        0.45551529,  0.14467132,  0.29872828,  0.37990902,  0.59752585,
        0.55000142,  0.65544317,  0.68797806,  0.48702056,  0.33624846,
        0.45942151,  0.22551828,  0.36134879,  0.27311145,  0.78521269,
        0.77380731,  0.0474938 ,  0.77618568,  0.43863636,  0.36962059,
        0.59153998,  0.78398268,  0.69271116,  0.41404028,  0.44570065,
        0.72895758,  0.75269798,  0.32822367,  0.73820101,  0.36877755])
```

In [32]:
```
sns.histplot(s,kde=True)
```

Out[32]:
```
<Axes: ylabel='Count'>
```

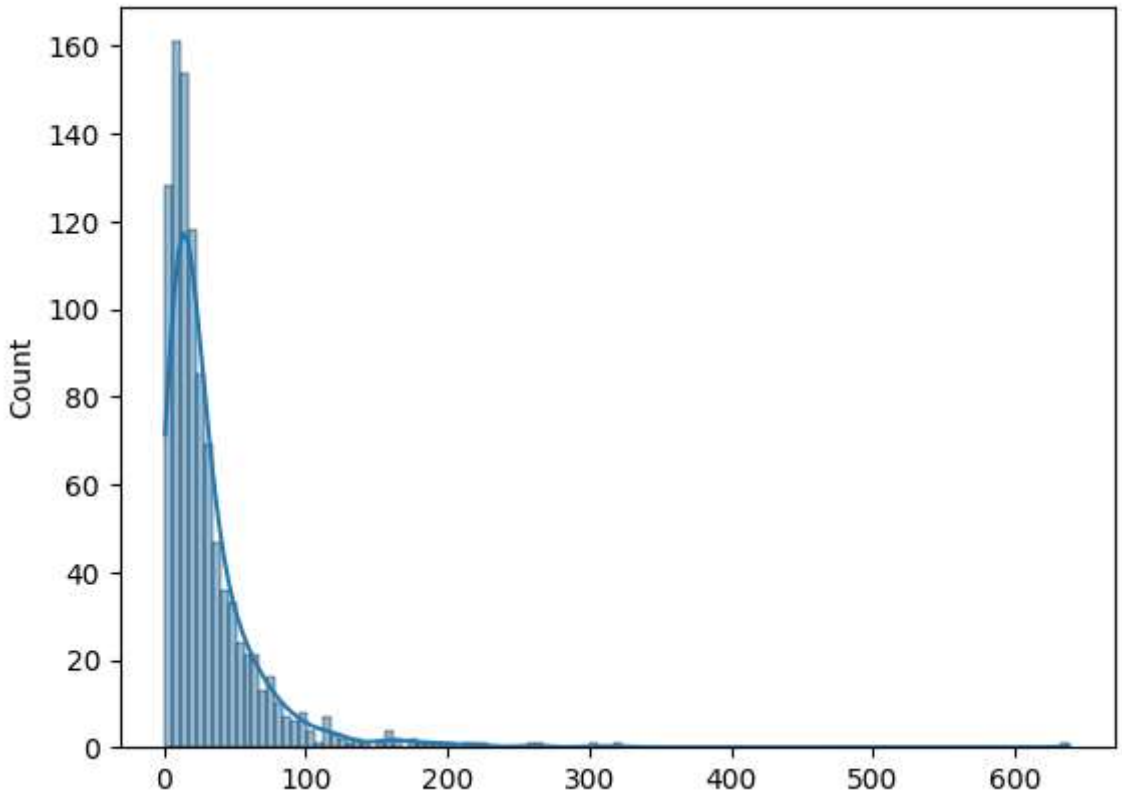


In [33]:
```
# Log Normal Distribution, Power Law Distribution
```

In [38]:
```
mu, sigma = 3., 1. # mean and standard deviation
s=np.random.lognormal(mu,sigma,1000)
```

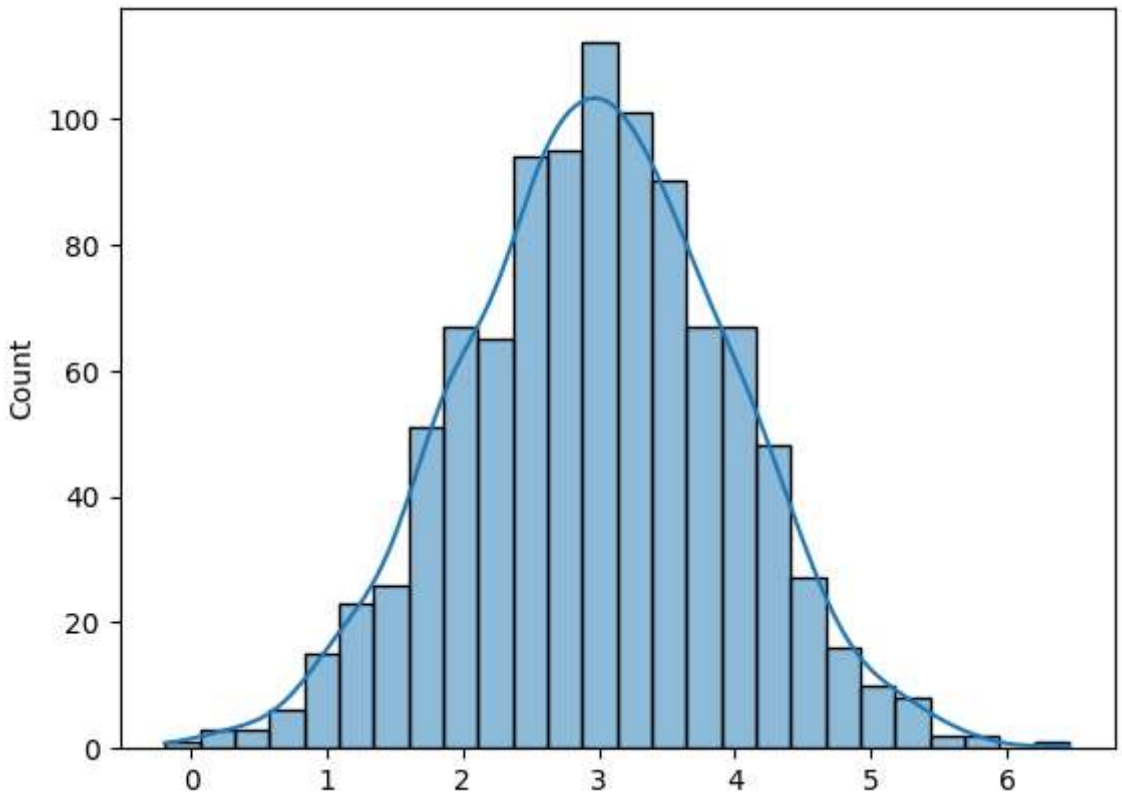
In [39]:
```
sns.histplot(s,kde=True)
```

Out[39]:
```
<Axes: ylabel='Count'>
```



In [40]:
```
sns.histplot(np.log(s),kde=True)
```

Out[40]: <Axes: ylabel='Count'>

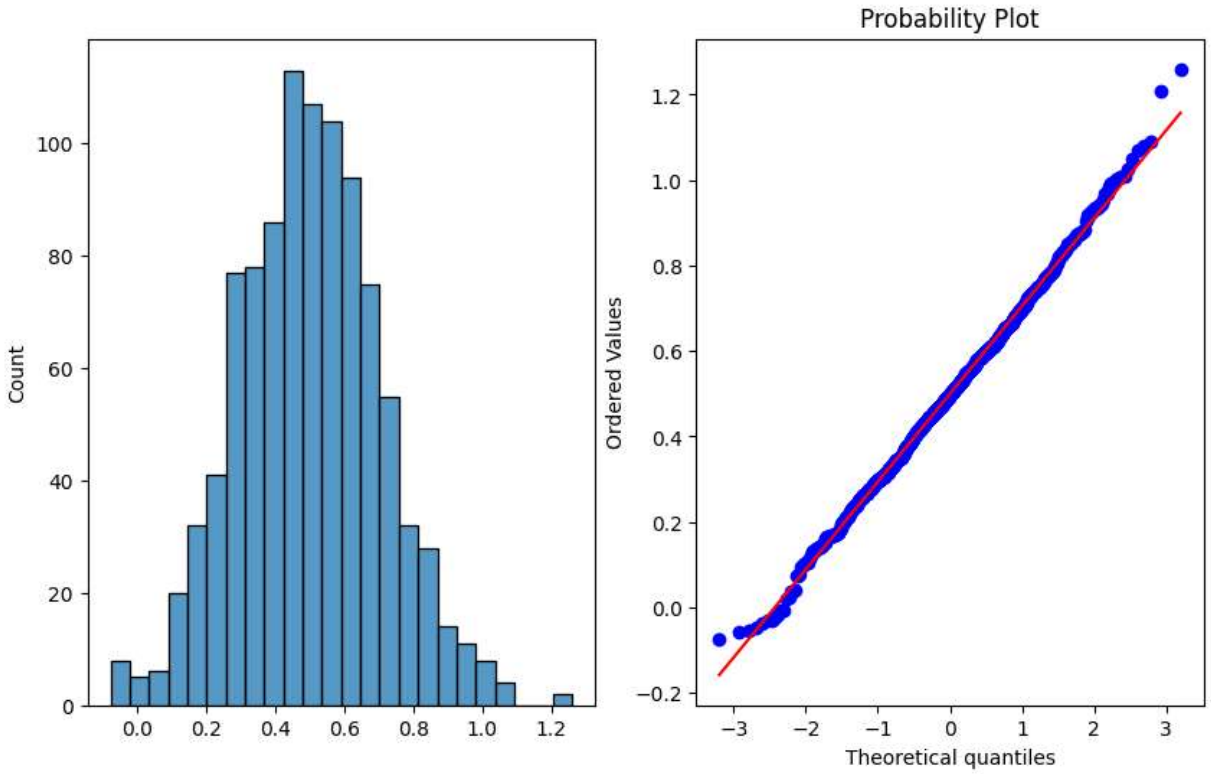


In [41]:
```
# Check weather the distribution is normal distribution
```

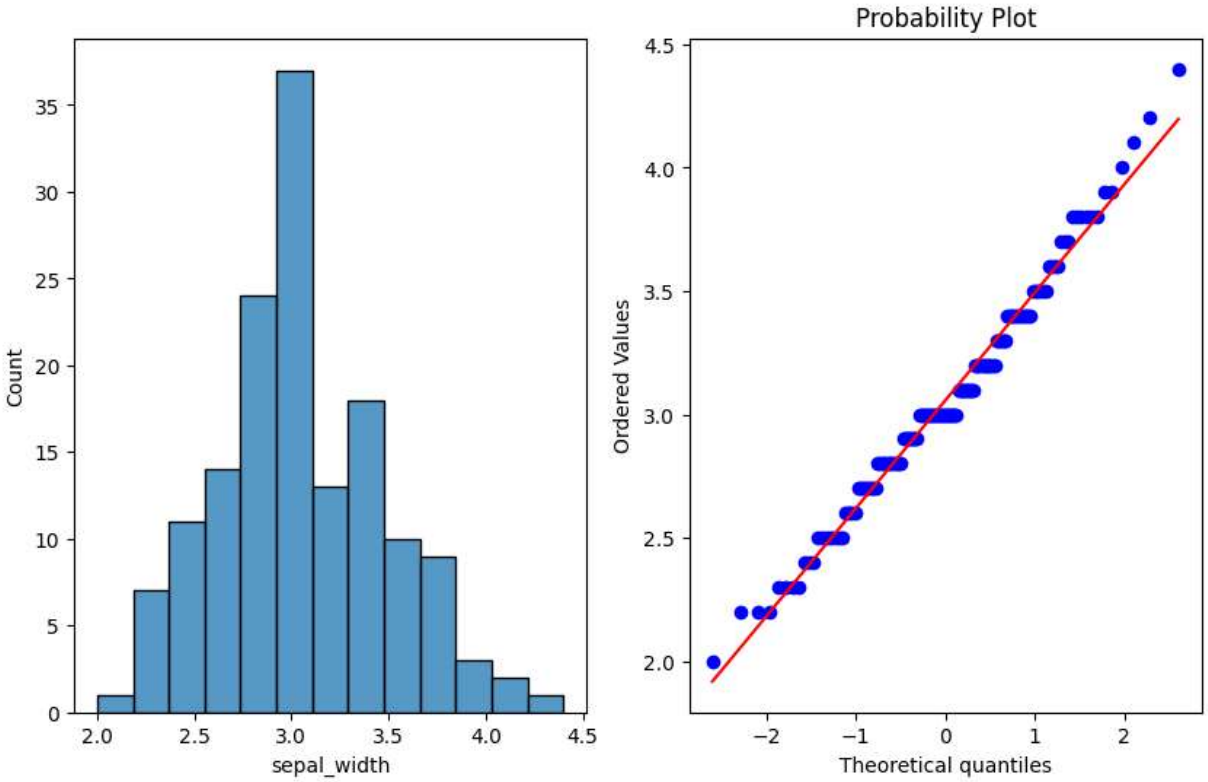
In [42]:
```
import matplotlib.pyplot as plt
import scipy.stats as stat
import pylab
```

In [43]:
```
def plot_data(sample):
    plt.figure(figsize=(10,6))
    plt.subplot(1,2,1)
    sns.histplot(sample)
    plt.subplot(1,2,2)
    stat.probplot(sample,dist='norm',plot=pylab)
    plt.show
```

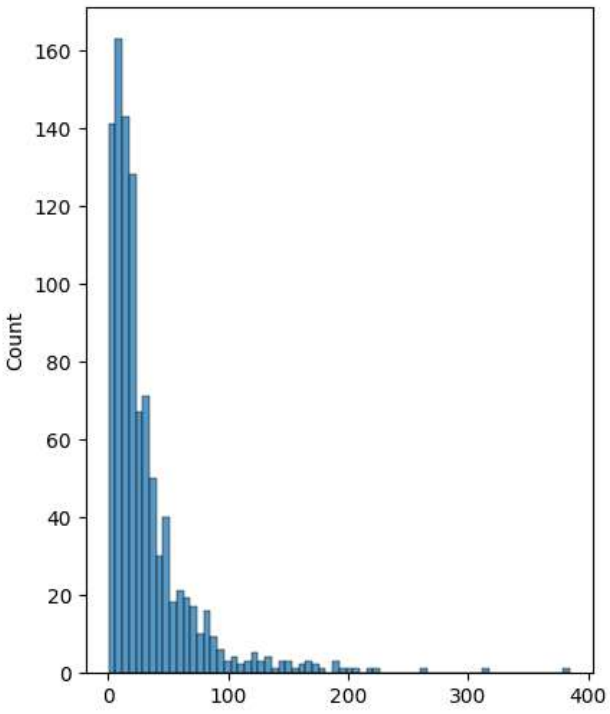
In [44]:
```
# create a normal distribution data
s=np.random.normal(0.5,0.2,1000)
plot_data(s)
```

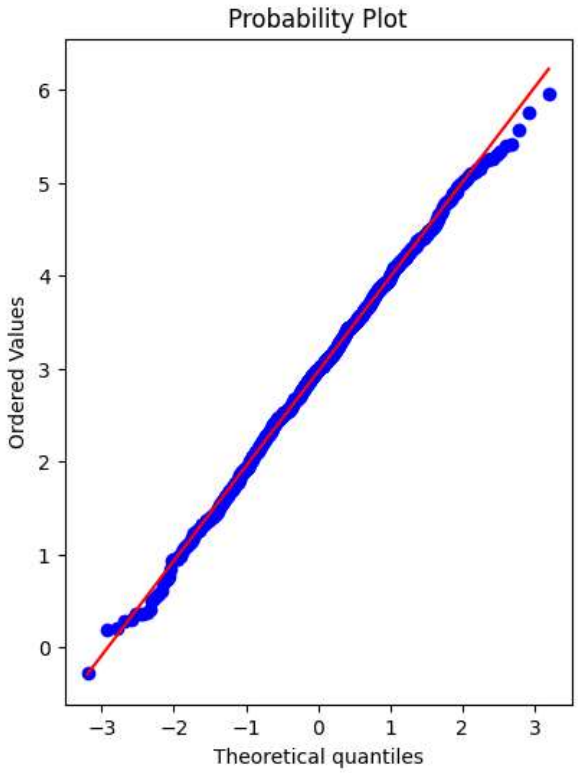


In [46]:
```
plot_data(df['sepal_width'])
```



In [48]:
```
mu, sigma = 3., 1. # mean and standard deviation
sample=np.random.lognormal(mu,sigma,1000)

def plot_data(sample):
    plt.figure(figsize=(10,6))
    plt.subplot(1,2,1)
    sns.histplot(sample)
    plt.subplot(1,2,2)
    stat.probplot(np.log(sample),dist='norm',plot=pylab)
    plt.show


plot_data(sample)
```





Pearson and Sperman Rank Correlation

In [49]:
```
df=sns.load_dataset('tips')
df.head()
```

Out[49]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

In [50]:
```
import pandas as pd
```

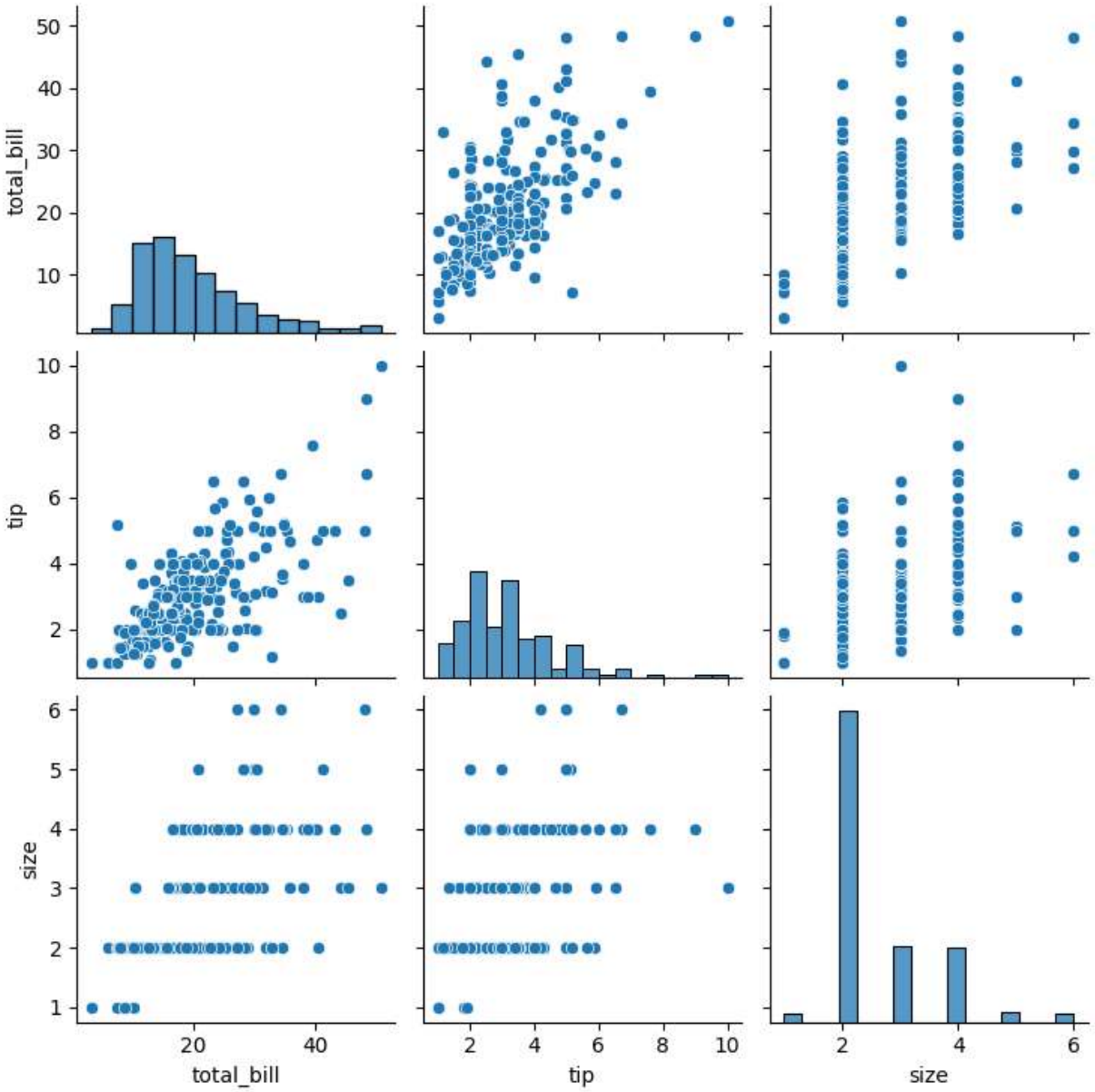
In [52]:
```
df=df.select_dtypes(include='number')
df.corr()
```

Out[52]:

	total_bill	tip	size
total_bill	1.000000	0.675734	0.598315
tip	0.675734	1.000000	0.489299
size	0.598315	0.489299	1.000000

In [53]:
```
sns.pairplot(df)
```

Out[53]: <seaborn.axisgrid.PairGrid at 0x248b1d222d0>



In [ ]: