Samit Dhawal

Q1) Identify the Data type for the Following:

Activity	Data Type
Number of beatings from Wife	Discrete
Results of rolling a dice	Discrete
Weight of a person	Continuous
Weight of Gold	Continuous
Distance between two places	Continuous
Length of a leaf	Continuous
Dog's weight	Continuous
Blue Color	Discrete
Number of kids	Discrete
Number of tickets in Indian railways	Discrete
Number of times married	Discrete
Gender (Male or Female)	Discrete

Q2) Identify the Data types, which were among the following

Nominal, Ordinal, Interval, Ratio.

Data	Data Type
Gender	Nominal
High School Class Ranking	Ordinal
Celsius Temperature	Interval
Weight	Ratio
Hair Color	Nominal
Socioeconomic Status	Ordinal
Fahrenheit Temperature	Interval
Height	Ratio
Type of living accommodation	Nominal
Level of Agreement	Ordinal
IQ(Intelligence Scale)	Ordinal
Sales Figures	Ratio
Blood Group	Nominal
Time Of Day	Ordinal
Time on a Clock with Hands	Interval
Number of Children	Ratio

Religious Preference	Nominal
Barometer Pressure	Interval
SAT Scores	Interval
Years of Education	Ordinal

Q3) Three Coins are tossed, find the probability that two heads and one tail are obtained?

Ans: If 3 coins are tossed, total_outcomes = 2^3 (2 for each coin, 3 times tossed)

The possible combination formed with two heads and one tail are,

{H, H, T}, {T,H, H}, {H, T, H} which is 3.

So, P(Two heads and 1 tail) = 3/total_outcomes => 3/8 => **0.375**

Q4) Two Dice are rolled, find the probability that sum is

a) Equal to 1
b) Less than or equal to 4
c) Sum is divisible by 2 and 3

Ans: Two dice rolled, total_outcomes = 6^2 (6 for each dice, 2 dice rolled)

(a) For sum equal to 1, minimum value for single die is 1, so adding two minimum values from two different rolled dice will be 2, hence it gives minimum sum of 2. So, there are no such combinations. So, probability is,

P(Sum Equal to 1) = 0/total_outcomes = 0/36 => **0**

(b) When rolling two dice the probability less than or equal to 4 is, So, the possible outcomes are, {1,1}, {1,2}, {1,3}, {2,1}, {2,2}, {3,1},
So,

P(Less than or Equal to 4) = 6/36 => **0.16666**

(c) Possible combinations are,
Combination formed are => {1,5},{2,4},{3,3},{4,2},{5,1},{6,6}
So, P(Sum divisible by 2 and 3) = 6/36 => **0.16666**

Q5) A bag contains 2 red, 3 green and 2 blue balls. Two balls are drawn at random. What is the probability that none of the balls drawn is blue?

Ans. Total number of outcomes 2 balls can be drawn is,

Combination => C(7,2) = 7! / (7-2)!*2! => 5040/(240) => **21**

Drawing 2 balls without any blue balls, means we have to select from 2 red and 3 green balls. So total balls = 5, and 2 balls drawn.

Combination => C(5,2) = 5!/ (5-2)!*2! => 120/12 => **10**

So, Probability is = 10/21 => **0.476**

Q6) Calculate the Expected number of candies for a randomly selected child

Below are the probabilities of count of candies for children (ignoring the nature of the child-Generalized view)

CHILD	Candies count	Probability
A	1	0.015
B	4	0.20
C	3	0.65
D	5	0.005
E	6	0.01
F	2	0.120

Child A – probability of having 1 candy = 0.015.

Child B – probability of having 4 candies = 0.20

Ans. To find expected number of candies, we have to multiply candies with the probability values and sum those values such as,

Expected number of candies = (1*0.015) + (4*0.20) + (3*0.65) + (5*0.005) + (6*0.01) + (2*0.120) => 3.09 => **<u>3</u>**

<u>3</u> candies to be expected for randomly selected child.

Q7) Calculate Mean, Median, Mode, Variance, Standard Deviation, Range & comment about the values / draw inferences, for the given dataset

- For Points, Score, Weigh>
  Find Mean, Median, Mode, Variance, Standard Deviation, and Range and also Comment about the values/ Draw some inferences.

Use Q7.csv file

Ans.

```
[8]
# Samit
# Q7

#Importing required libraries
import pandas as pd
import numpy as np
from scipy import stats
```

```
[9]
# Uploading dataset from local device
from google.colab import files
uploaded = files.upload()
```

Choose files Q7.csv

- **Q7.csv**(text/csv) - 962 bytes, last modified: 30/05/2023 - 100% done

```
Saving Q7.csv to Q7.csv
```

```
[11]
#Reading the csv file and storing to df dataframe
df = pd.read_csv("Q7.csv")
df.head()
```

	Unnamed: 0	Points	Score	Weigh
0	Mazda RX4	3.90	2.620	16.46
1	Mazda RX4 Wag	3.90	2.875	17.02
2	Datsun 710	3.85	2.320	18.61
3	Hornet 4 Drive	3.08	3.215	19.44
4	Hornet Sportabout	3.15	3.440	17.02

```
[20]
#Checking for null values
df.isna().sum()
```

```
Unnamed: 0    0
Points        0
Score         0
Weigh         0
dtype: int64
```

- The value for mean is, mean(data) = sum(data all points)/len(data)
- The value for median is,
  - if len(data) is even,
    - median =value at ((len(data)/2) + ((len(data)/2)+1))/2
  - if len(data) is odd,
    - median = value at len(data)/2
- The value for mode is, mode(data) = more frequent item in the data. It can have more than one mode value
- variance = sum(datapoint – mean(datapoints))/len(datapoints)-1
- std. dev. = sqrt(sum(datapoint – mean(datapoints))/len(datapoints)-1)
- range = It is the difference between maximum value and minimum value of the dataset.

```
[32]
# Column = 'Points'


# Below, the calculation for mean, median, mode, variance, standard deviation, and range is calculated for Points column

mean_ = df.Points.mean()
print("Mean of the Points Columm: ",mean_)
median_ = df.Points.median()
print("Median of the Points Columm: ",median_)
mode_ = df.Points.mode()
print("Mode of the Points Columm: \n",mode_)
var_ = df.Points.var()
print("Variance of the Points Columm: ",var_)
std_ = df.Points.std()
print("Standard Deviation of the Points Columm: ",std_)
range_ = max(df.Points)-min(df.Points)
print("Range of the Points Columm: ",range_)
```

```
Mean of the Points Columm:  3.5965625
Median of the Points Columm:  3.6950000000000003
Mode of the Points Columm:
 0    3.07
1    3.92
Name: Points, dtype: float64
Variance of the Points Columm:  0.2858813508064516
Standard Deviation of the Points Columm:  0.5346787360709715
Range of the Points Columm:  2.17
```

```
[35]
# Column = 'Score'

# Below, the calculation for mean, median, mode, variance, standard deviation, and range is calculated for Score Column

mean_ = df.Score.mean()
print("Mean of the Score Columm: ",mean_)
median_ = df.Score.median()
print("Median of the Score Columm: ",median_)
mode_ = df.Score.mode()
print("Mode of the Score Columm: \n",mode_)
var_ = df.Score.var()
print("Variance of the Score Columm: ",var_)
std_ = df.Score.std()
print("Standard Deviation of the Score Columm: ",std_)
range_ = max(df.Score)-min(df.Score)
print("Range of the Score Columm: ",range_)
```

```
Mean of the Score Columm:  3.2172500000000004
Median of the Score Columm:  3.325
Mode of the Score Columm:
 0    3.44
Name: Score, dtype: float64
Variance of the Score Columm:  0.9573789677419354
Standard Deviation of the Score Columm:  0.9784574429896966
Range of the Score Columm:  3.9110000000000005
```

```
[36]
# Column = 'Weigh'


# Below, the calculation for mean, median, mode, variance, standard deviation, and range is calculated for Weigh Column

mean_ = df.Weigh.mean()
print("Mean of the Weigh Columm: ",mean_)
median_ = df.Weigh.median()
print("Median of the Weigh Columm: ",median_)
mode_ = df.Weigh.mode()
print("Mode of the Weigh Columm: \n",mode_)
var_ = df.Weigh.var()
print("Variance of the Weigh Columm: ",var_)
std_ = df.Weigh.std()
print("Standard Deviation of the Weigh Columm: ",std_)
range_ = max(df.Weigh)-min(df.Weigh)
print("Range of the Weigh Columm: ",range_)
```

```
Mean of the Weigh Columm:  17.848750000000003
Median of the Weigh Columm:  17.71
Mode of the Weigh Columm:
 0    17.02
1    18.90
Name: Weigh, dtype: float64
Variance of the Weigh Columm:  3.193166129032258
Standard Deviation of the Weigh Columm:  1.7869432360968431
Range of the Weigh Columm:  8.399999999999999
```

Q8) Calculate Expected Value for the problem below

a) The weights (X) of patients at a clinic (in pounds), are
108, 110, 123, 134, 135, 145, 167, 187, 199

Assume one of the patients is chosen at random. What is the Expected Value of the Weight of that patient?

```python
# Samit
# Q8


# Creating the function which calculates expected weight of the patient where iterating through each weights given
# multiplying them with probability of individual observations which 1/9, total observations are 9

def expected_val(weights):
  weight_ = 0
  for weight in weights:
    weight_ += weight*(1/9)
  return weight_



weights = [108,110,123,134,135,145,167,187,199]
print("Expected Weight of the patient is: ",round(expected_val(weights),2))
```

```
Expected Weight of the patient is:  145.33
```

Q9) Calculate Skewness, Kurtosis & draw inferences on the following data

Cars speed and distance

Use Q9_a.csv

[52]
```python
# Samit
# Q9

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

[46]
```python
from google.colab import files
uploaded = files.upload()
```

Choose files Q9_a.csv

- **Q9_a.csv**(text/csv) - 502 bytes, last modified: 30/05/2023 - 100% done

```
Saving Q9_a.csv to Q9_a.csv
```

[47]
```python
df = pd.read_csv("Q9_a.csv")
df.head()
```

	Index	speed	dist
0	1	4	2
1	2	4	10
2	3	7	4
3	4	7	22
4	5	8	16

[59]
```python
print("Speed skewness: ", df.speed.skew())
print("Distance skewness: ", df.dist.skew())
```

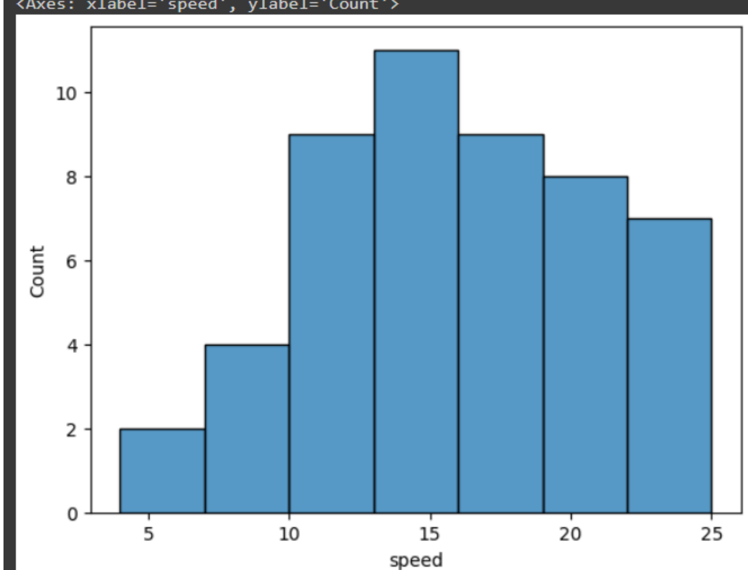
```
Speed skewness:  -0.11750986144663393
Distance skewness:  0.8068949601674215
```

[60]
```python
print("Speed kurtosis: ", df.speed.kurt())
print("Distance kurtosis: ", df.dist.kurt())
```

```
Speed kurtosis:  -0.5089944204057617
Distance kurtosis:  0.4050525816795765
```

```python
print(df.speed.mean())
print(df.speed.median())

sns.histplot(df.speed)
```

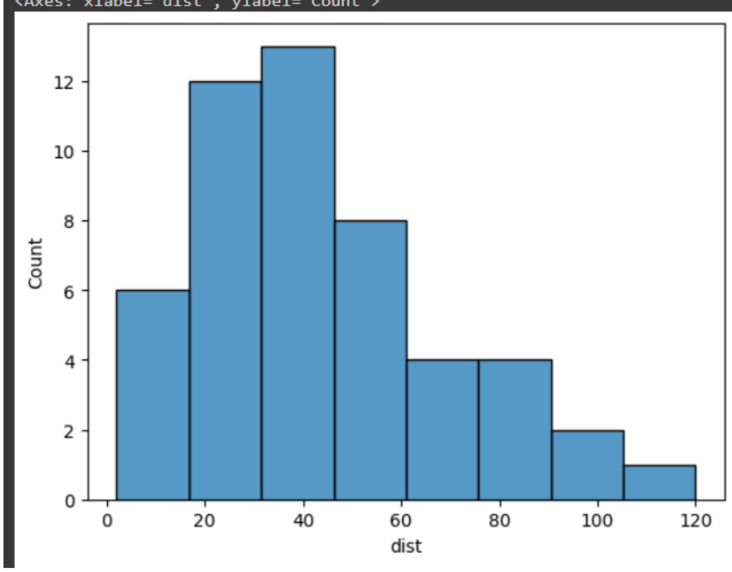
```
15.4
15.0
<Axes: xlabel='speed', ylabel='Count'>
```



In the above plot we can see the symmetric or normal distribution where mean and median are also same and the kurtosis value indicates that the distribution has light tails and there are fewer extreme values in the distribution than in a normal distribution.

```
print(df.dist.mean())
print(df.dist.median())

sns.histplot(df.dist)
```

```
42.98
36.0
<Axes: xlabel='dist', ylabel='Count'>
```



In the above plot we can see that the distribution is right skewed where the concentration of the lies towards the lower side. Whereas, kurtosis value indicated that it is slightly more peaked and has slightly heavier tails compared to normal distribution. Which means distribution has slightly sharper peak and more outliers in the tails compared to a normal distribution.

SP and Weight (WT)

Use Q9_b.csv

[63]
```
# Q9(b)

from google.colab import files
uploaded = files.upload()
```

Choose files Q9_b.csv

- **Q9_b.csv**(text/csv) - 2420 bytes, last modified: 30/05/2023 - 100% done

```
Saving Q9_b.csv to Q9_b.csv
```

[64]
```
df = pd.read_csv("Q9_b.csv")
df.head()
```

	Unnamed: 0	SP	WT
0	1	104.185353	28.762059
1	2	105.461264	30.466833
2	3	105.461264	30.193597
3	4	113.461264	30.632114
4	5	104.461264	29.889149

[65]
```
print("SP skewness: ", df.SP.skew())
print("Weight skewness: ", df.WT.skew())
```

```
SP skewness:  1.6114501961773586
Weight skewness:  -0.6147533255357768
```

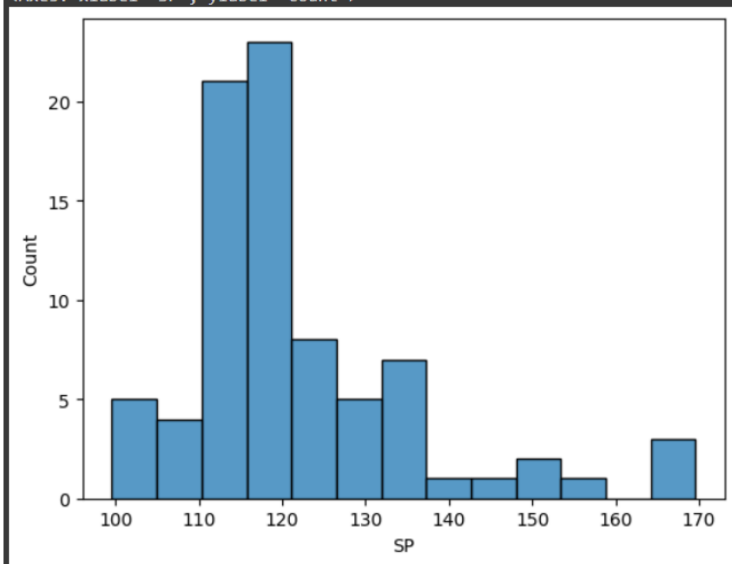
[66]
```
print("SP kurtosis: ", df.SP.kurt())
print("Weight kurtosis: ", df.WT.kurt())
```

```
SP kurtosis:  2.9773289437871835
Weight kurtosis:  0.9502914910300326
```

In the below plot we can see the distribution is right skewed which means the data is not normally distributed. However, the kurtosis value indicates that the distribution has heavier tails and a sharper peak compared to the normal distribution (which has a kurtosis value of 0). Positive value indicates that the distribution has more outliers or extreme values in the tails and a higher peak compared to a normal distribution.

```
print(df.SP.mean())
print(df.SP.median())

sns.histplot(df.SP)
```

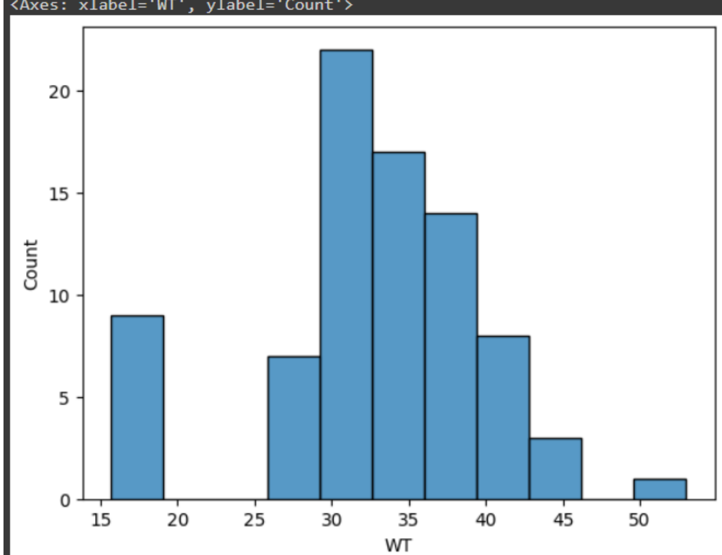
```
121.54027218037035
118.2086984
<Axes: xlabel='SP', ylabel='Count'>
```



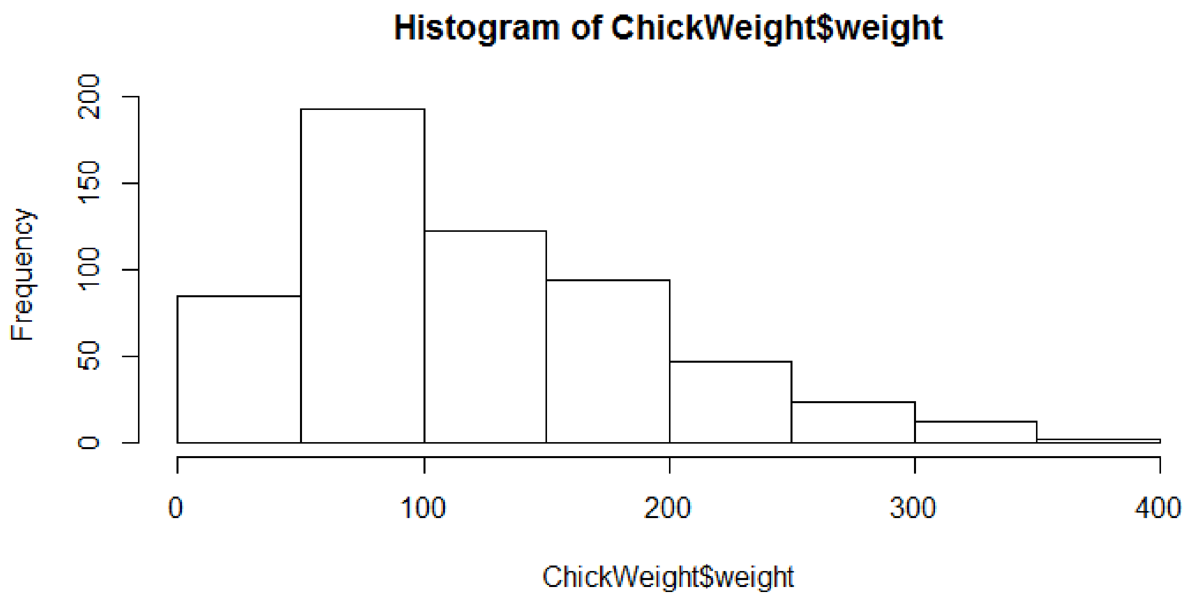
In the below plot we can see the data is almost normally distributed where mean and median are almost same with little left skewness. Whereas, kurtosis value indicates that the shape is slightly more peaked and has slightly heavier tails.

```
print(df.WT.mean())
print(df.WT.median())

sns.histplot(df.WT)
```

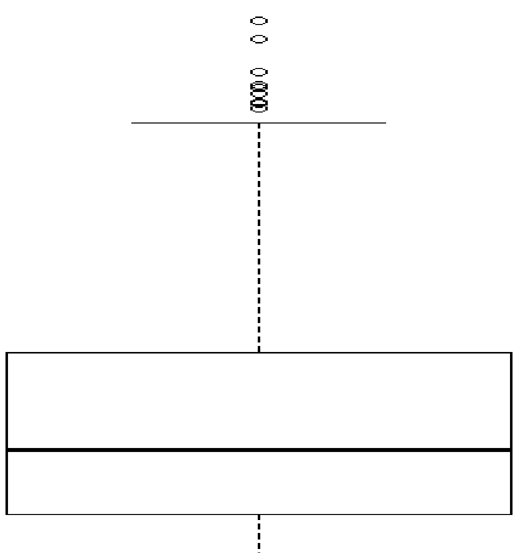
```
32.412576910246905
32.73451818
<Axes: xlabel='WT', ylabel='Count'>
```



Q10) Draw inferences about the following boxplot & histogram



In the above Histogram we can clearly see that it is Right skewed in which most of the ChickWeight lies towards the lower side of the data.



In above, box plot we can see that there are outliers which shows that there are over weighted ChickWeights.

Q11) Suppose we want to estimate the average weight of an adult male in Mexico. We draw a random sample of 2,000 men from a population of 3,000,000 men and weigh them. We find that the average person in our sample weighs 200 pounds, and the standard deviation of the sample is 30 pounds. Calculate 94%,98%,96% confidence interval?

Ans:

Population = 3000000

Sample mean = 200

Sample std. dev. = 30

n = 2000

df = 1999

std. error = Sample std/sqrt(n) = 30/sqrt(2000) => **0.67082**

#Since population std. dev. is not given so sample std. dev. is used to calculate std. error.

Now,

t_94% = 1.960 (from t-distribution table)

t_98% = 2.326 (from t-distribution table)

t_96% = 1.960 (from t-distribution table)

94% confidence interval when sample dev. is 30 and sample mean is 200

Lower limit = Sample mean – (t_94%*std. error) = 200-(1.960*0.67082) => **198.6852**

Upper limit = Sample mean + (t_94%*std. error) = 200+(1.960*0.67082) => **201.3148**

Conclusion: 94% of the time average weight of an adult male would be between 198.6852 and 201.3148.

98% confidence interval when sample dev. is 30 and sample mean is 200

Lower limit = Sample mean – (t_98%*std. error) = 200-(2.326 *0.67082) => **198.4397**

Upper limit = Sample mean + (t_98%*std. error) = 200+(2.326*0.67082) => **201.5603**

Conclusion: 94% of the time average weight of an adult male would be between 198.4397 and 201.5603.

96% confidence interval when sample dev. is 30 and sample mean is 200

Lower limit = Sample mean – (t_96%*std. error) = 200-(1.960*0.67082) => **198.6852**

Upper limit = Sample mean + (t_96%*std. error) = 200+(1.960*0.67082) => **201.3148**

Conclusion: 96% of the time average weight of an adult male would be between 198.6852 and 201.3148.

Note: If there are little variation then that will be because of the value taken from the t-distribution table which does not have exact match value in the t-distribution table.

Q12) Below are the scores obtained by a student in tests

34,36,36,38,38,39,39,40,40,41,41,41,41,42,42,45,49,56

```
[12]
# q12
import numpy as np
import pandas as pd

scores = [34,36,36,38,38,39,39,40,40,41,41,41,41,42,42,45,49,56]
scores.sort()
df = pd.Series(scores)
df
```

1) Find mean, median, variance, standard deviation.

```
print(df.mean())
print(df.median())
print(df.var())
print(df.std())

41.0
40.5
25.529411764705884
5.05266382858645
```

2) What can we say about the student marks?

Ans: we can say that the average marks obtained by the students in the tests is 41 with marks deviation of 5.

Q13) What is the nature of skewness when mean, median of data are equal?

Ans: When the mean and median of a dataset are equal, it indicates zero skewness, implying that the dataset is symmetric. This means that the data is not skewed towards either the right or the left. The concentration of the dataset lies to the center which forms bell shaped distribution.

Q14) What is the nature of skewness when mean > median?

Ans: When the mean is greater than the median, it suggests positive skewness, indicating that the dataset is right-skewed. In other words, the tail of the distribution extends towards the right side, while the majority of the data points are concentrated towards the left side.

Q15) What is the nature of skewness when median > mean?

Ans: When the median is greater than the mean, it suggests negative skewness, indicating that the dataset is left-skewed. In other words, the tail of the distribution extends towards the left side, while the majority of the data points are concentrated towards the right side.

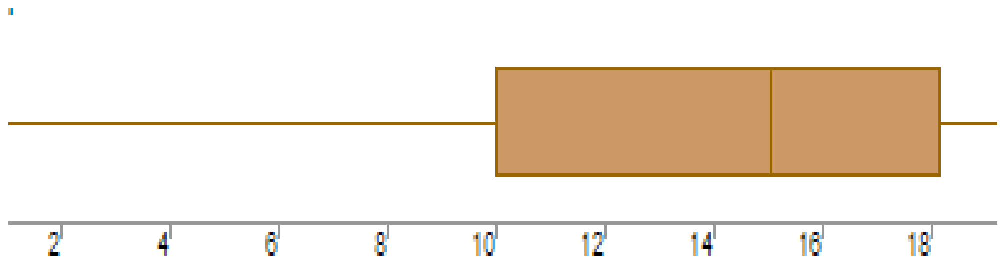
Q16) What does positive kurtosis value indicates for a data?

Ans: The positive kurtosis indicates that there are more extreme values in the data set, both positive and negative due to which it forms higher peak in the distribution compare to normal distribution in which peak is flat.

Q17) What does negative kurtosis value indicates for a data?

Ans: The negative kurtosis indicates that there are fewer extreme values in the data set, both positive and negative. The peak is the point where the distribution is higher. However, in negative distribution the peak is lower than the normal distribution in which peak is flat.

Q18) Answer the below questions using the below boxplot visualization.



What can we say about the distribution of the data?

Ans: The data is negatively skewed where the median value is close to the top of the box and we can see a longer whisker on the left side and small whisker on the right side. Concentration of data is more towards right side of distribution.

What is nature of skewness of the data?

Ans: The nature of the skewness of the data is Left skewed where we can see the concentration of the data is more towards the right side of the distribution.

What will be the IQR of the data (approximately)?

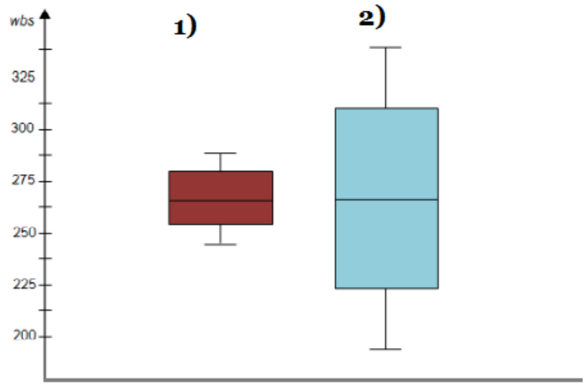
Ans:

Q3 = 18

Q1 = 10

IQR = Q3-Q1 = 18-10 => **8**(Approximately)

Q19) Comment on the below Boxplot visualizations?



Draw an Inference from the distribution of data for Boxplot 1 with respect Boxplot 2.

Ans: In the above two Boxplot it seems to have a symmetric distribution. They differ each other by Four parameters only i.e., Minimum, Q1, Q3, and Maximum value the Median value is same for both the Boxplots.

Q20) Calculate probability from the given dataset for the below cases

Data _set: Cars.csv

Calculate the probability of MPG of Cars for the below cases.

MPG <- Cars$MPG

a. P(MPG>38)
b. P(MPG<40)
c. P (20<MPG<50)

```
# Since, the nature of 'MPG' column is Continuous, so we have to go with the z-score calculation
# Z_score can be calculated as,
# z_score = (datapoint - Mean) / (Std. Dev.),
# if condition is, to calculate probability for less than any value(suppose, X) then probability value(P(X)) will be (value) from z-score table for above z_score calculated value but
# if condition is, to calculate probability for greater than any value(X), then probability value(P(X)) will be (1-value) from z-score table for above z_score calculated value

mean = np.mean(df.MPG)
std = np.std(df.MPG)
q_a = 1 - stats.norm.cdf(38,mean,std)
print("P(MPG>38): ",q_a)
q_b = stats.norm.cdf(40,mean,std)
print("P(MPG<40): ",q_b)
q_c = stats.norm.cdf(50,mean,std) - stats.norm.cdf(20,mean,std)
print("P(20<MPG<50): ",q_c)
```

```
P(MPG>38):  0.3466923853688819
P(MPG<40):  0.7306083416219191
P(20<MPG<50):  0.9009686820346053
```

Q21) Check whether the data follows normal distribution

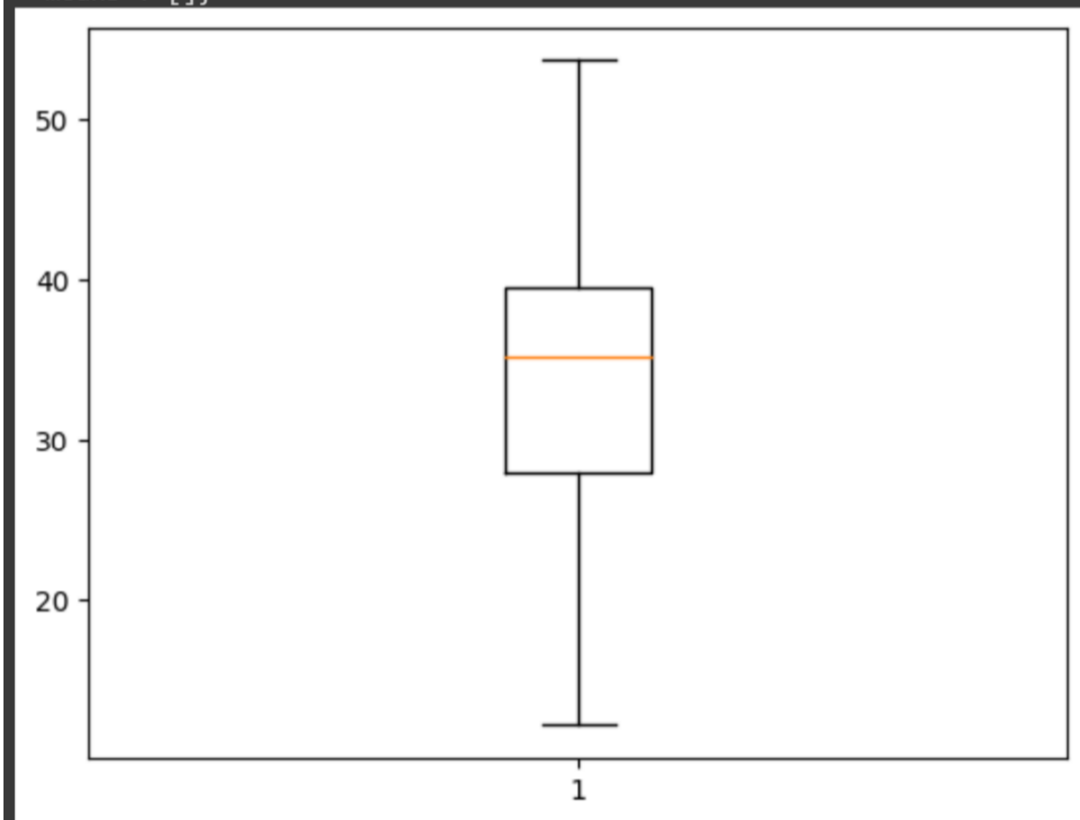
a) Check whether the MPG of Cars follows Normal Distribution
   Dataset: Cars.csv

Ans: No, It doesn't follows Normal Distribution since it's median value is higher than the mean value of MPG which shows the Negative skewness of the data which is skewed towards left and concentration is little more towards right of the distribution and median value is near to the top of the box which is shown below.

```
[28]  df.MPG.median()

      35.15272697

[29]  df.MPG.mean()

      34.42207572802469
```

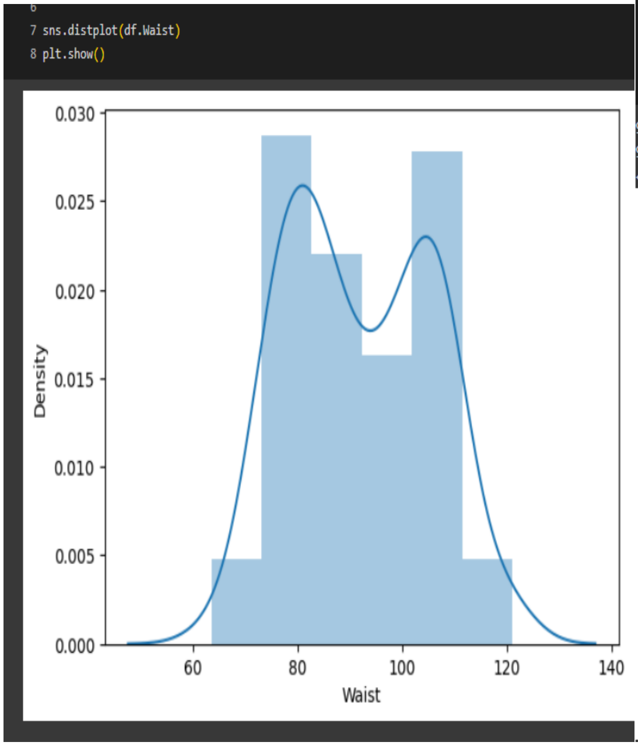
```
1 plt.boxplot(df.MPG)
```

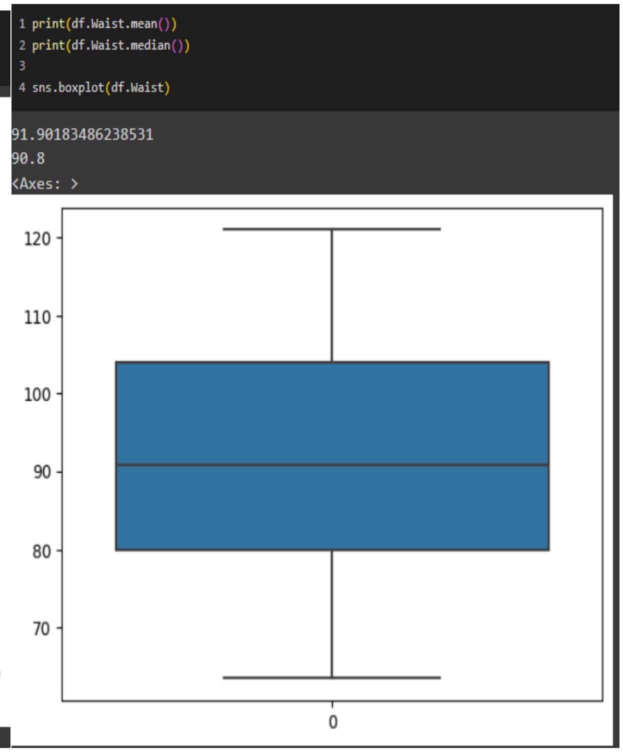
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7ff7fc5ff9d0>,
  <matplotlib.lines.Line2D at 0x7ff7fc5ff1f0>],
 'caps': [<matplotlib.lines.Line2D at 0x7ff7fc5ffe20>,
  <matplotlib.lines.Line2D at 0x7ff7fc5ff400>],
 'boxes': [<matplotlib.lines.Line2D at 0x7ff7fc5fece0>],
 'medians': [<matplotlib.lines.Line2D at 0x7ff7fc5ffac0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7ff7fc5fe0b0>],
 'means': []}
```



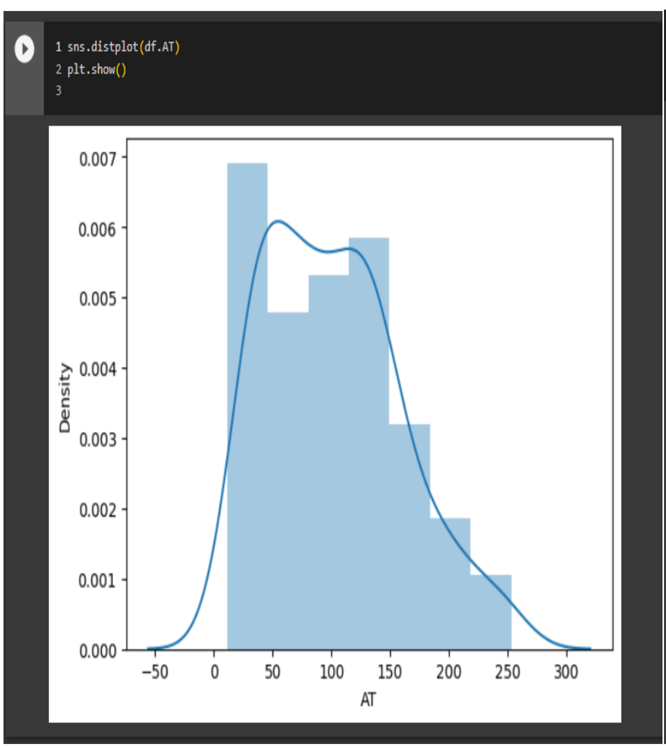
b) Check Whether the Adipose Tissue (AT) and Waist Circumference (Waist) from wc-at data set follows Normal Distribution
Dataset: wc-at.csv

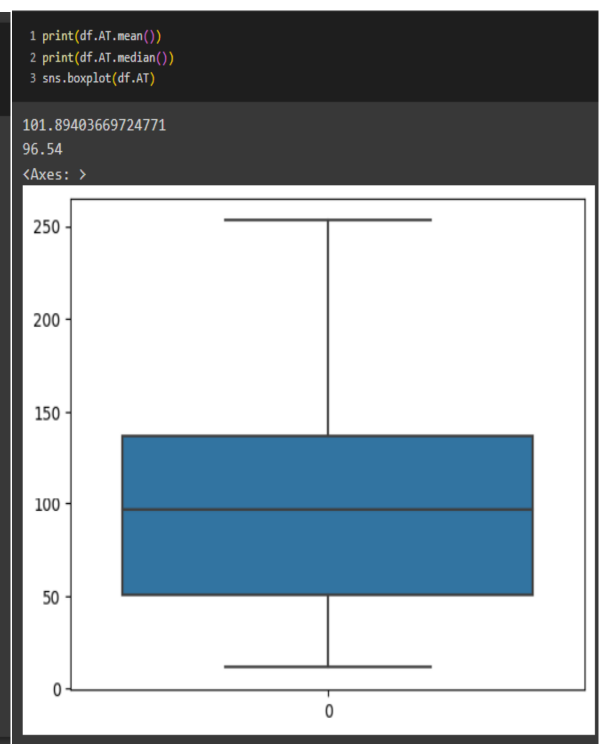
The column Waist Circumference seems to have the Normal Distribution which we can see from the below graph.





The column AT is right skewed that means AT column is not normally distributed which we can see from below graph.





Q22) Calculate the Z scores of 90% confidence interval,94% confidence interval, 60% confidence interval.

```
# Samit
# Q22

from scipy import stats
```

```
# Z score for 90% confidence interval
confidence_level90 = 0.90
# 10% probability is assigned to the tails, 5% each tail of two tailed distribution.
alpha90 = 1-confidence_level90
# PPF = Probability Point Function, Opposite of CDF, which calculates normal distribution value
# alpha90/2 means, alpha value dividing in two tails 5% each
z_score90 = stats.norm.ppf(1-(alpha90/2))
print("Z-score for 90% confidence interval: ",z_score90)

# Z score for 94% confidence interval
confidence_level94 = 0.94
alpha94 = 1-confidence_level94
z_score94 = stats.norm.ppf(1-(alpha94/2))
print("Z-score for 94% confidence interval: ",z_score94)

# Z score for 60% confidence interval
confidence_level60 = 0.60
alpha60 = 1-confidence_level60
z_score60 = stats.norm.ppf(1-(alpha60/2))
print("Z-score for 60% confidence interval: ",z_score60)
```

```
Z-score for 90% confidence interval:  1.6448536269514722
Z-score for 94% confidence interval:  1.8807936081512509
Z-score for 60% confidence interval:  0.8416212335729143
```

Q23) Calculate the t scores of 95% confidence interval, 96% confidence interval, 99% confidence interval for sample size of 25

```python
sample_size = 25

# T score for 95% confidence interval
confidence_level95 = 0.95
alpha95 = 1-confidence_level95
dof95 = sample_size-1      #dof = degree of freedom
# PPF = Probability Point Function, Opposite of CDF, which calculates normal distribution value
# PPF generally returns the exact point.
t_score95 = stats.t.ppf(1-(alpha95/2),df=dof95)
print("T-score for 95% confidence interval: ",t_score95)

# T score for 96% confidence interval
confidence_level96 = 0.96
alpha96 = 1-confidence_level96
dof96 = sample_size-1
t_score96 = stats.t.ppf(1-(alpha96/2),df=dof96)
print("T-score for 96% confidence interval: ",t_score96)


# T score for 99% confidence interval
confidence_level99 = 0.99
alpha99 = 1-confidence_level99
dof99 = sample_size-1
t_score99 = stats.t.ppf(1-(alpha99/2),df=dof99)
print("T-score for 99% confidence interval: ",t_score99)
```

```
T-score for 95% confidence interval:  2.0638985616280205
T-score for 96% confidence interval:  2.1715446760080677
T-score for 99% confidence interval:  2.796939504772804
```

Q24**)** A Government company claims that an average light bulb lasts 270 days. A researcher randomly selects 18 bulbs for testing. The sampled bulbs last an average of 260 days, with a standard deviation of 90 days. If the CEO's claim were true, what is the probability that 18 randomly selected bulbs would have an average life of no more than 260 days

Hint:

rcode → pt(tscore,df)

df → degrees of freedom

```
# Samit
# Q24

from scipy import stats
import numpy as np

sample_size = 18
sample_mean = 260
pop_mean = 270
std_dev = 90
dof = sample_size-1       #dof = degree of freedom
t_score = (sample_mean-pop_mean) / (std_dev/np.sqrt(sample_size))
prob = stats.t.cdf(t_score,df=dof)      # Here is the case of no more than so we use cdf
print("Probability of 18 bulbs, average life no more than 260 days: ",prob)
```

```
Probability of 18 bulbs, average life no more than 260 days:  0.32167253567098364
```