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

Q1) Identify the Data type for the Following:

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) | Ratio |
| Sales Figures | Ratio |
| Blood Group | Nominal |
| Time Of Day | Interval |
| Time on a Clock with Hands | Interval |
| Number of Children | Ratio |
| Religious Preference | Nominal |
| Barometer Pressure | Ratio |
| SAT Scores | Ratio |
| Years of Education | Ratio |

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

Ans:- Using the idea of combinations, we can determine the likelihood of receiving two heads and one tail when three coins are tossed.Let's start by counting the total number of possible outcomes from tossing three coins. There are two possible outcomes for every coin, which are heads and tails.3 = 8 2 3 = 8 potential results.Let's now determine the number of results that provide precisely two heads and one tail. Combinations can be used to accomplish this. Out of the three locations, we can designate two for heads, while the third position will go to the tail.There are three ways to choose two positions for heads out of three: (3, 2) = 3 C(3,2) = 3.Thus, the quantity of results that have precisely two heads and

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

1. Equal to 1

**A sum equal to 1:** When both dice display a 1, there is just one way to get a sum of 1. The potential result is 1 and 1.

When rolling two dice, there are a total of six \* six = thirty-six possible results. As a result, 1/36 is the likelihood of receiving a sum of 1.

b)Less than or equal to 4

Possible outcomes for sums less than or equal to 4 are (1, 1), (1, 2), (1, 3), (2, 1), (2, 2), and (3, 1).

There are a total of 6 \* 6 = 36 possible outcomes

.Therefore, the probability of getting a sum less than or equal to 4 is 6/36 = 1/6.

c)Sum is divisible by 2 and 3

The sums that are divisible by both 2 and 3 are 6 and 12. Possible outcomes for a sum of 6 are (1, 5), (2, 4), (3, 3), (4, 2), and (5, 1).

Possible outcomes for a sum of 12 are (6, 6).Therefore, the probability of getting a sum divisible by both 2 and 3 is (5 + 1)/36 = 1/6.

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?

A:-

Total balls = 7 (2 red + 3 green + 2 blue)  
Total ways to draw 2 balls = (7,2)=21*C*(7,2)=21  
Number of ways to draw 2 balls without blue = (5,2)=10*C*(5,2)=10  
Probability of not drawing blue = 10/21  
A:- To calculate the probability of not drawing any blue balls, we'll consider the total number of ways to draw 2 balls from the bag and the number of ways to draw 2 balls without picking any blue balls.

Given:

- Total balls in the bag: 7 (2 red, 3 green, and 2 blue)

- Total ways to draw 2 balls: \( \binom{7}{2} = 21 \)

- Number of blue balls: 2

To calculate the number of ways to draw 2 balls without picking any blue balls:

- We'll consider only the red and green balls, which are 5 in total.

- Number of ways to draw 2 balls from these 5 balls: \( \binom{5}{2} = 10 \)

Now, the probability of not drawing any blue balls is given by the ratio of the number of ways to draw 2 balls without picking any blue balls to the total number of ways to draw 2 balls:\[ \text{Probability} = \frac{\text{Number of ways without blue}}{\text{Total number of ways}} = \frac{10}{21} \]

Therefore, the probability of not drawing any blue balls is \( \frac{10}{21} \).

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

A;- To calculate the expected number of candies for a randomly selected child, we multiply each candy count by its corresponding probability and sum up these values.Expected number of candies = (1 \* 0.015) + (4 \* 0.20) + (3 \* 0.65) + (5 \* 0.005) + (6 \* 0.01) + (2 \* 0.120)

Expected number of candies = 0.015 + 0.80 + 1.95 + 0.025 + 0.06 + 0.24

Expected number of candies = 3.085

Therefore, the expected number of candies for a randomly selected child is 3.085.

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**

A:- import pandas as pd

# Load the dataset

data = pd.read\_csv("wc-at.csv")

# Display the first few rows of the dataset

print(data.head())

# Calculate Mean, Median, Mode, Variance, Standard Deviation, and Range for each column

statistics = {

"Mean": data.mean(),

"Median": data.median(),

"Mode": data.mode().iloc[0],

"Variance": data.var(),

"Standard Deviation": data.std(),

"Range": data.max() - data.min()

}

# Display the calculated statistics

for column, values in statistics.items():

print(f"\n{column}:\n{values}")

Output:- Waist AT

0 74.75 25.72

1 72.60 25.89

2 81.80 42.60

3 83.95 42.80

4 74.65 29.84

Mean:

Waist 91.901835

AT 101.894037

dtype: float64

Median:

Waist 90.80

AT 96.54

dtype: float64

Mode:

Waist 94.5

AT 121.0

Name: 0, dtype: float64

Variance:

Waist 183.849626

AT 3282.689835

dtype: float64

Standard Deviation:

Waist 13.559116

AT 57.294763

dtype: float64

Range:

Waist 57.50

AT 241.56

dtype: float64

These all about of the mean,median,Mode,Variance,Range and Standard Deviation

Q8) Calculate Expected Value for the problem below

1. 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?

A:- To calculate the expected value of the weight of a randomly chosen patient, we sum up all the weights and divide by the total number of patients.

Weights of patients: 108, 110, 123, 134, 135, 145, 167, 187, 199

Sum of weights = 108 + 110 + 123 + 134 + 135 + 145 + 167 + 187 + 199 = 1298

Total number of patients = 9

Expected value = Sum of weights / Total number of patients

Expected value = 1298 / 9

Expected value ≈ 144.22 pounds

Therefore, the expected value of the weight of a randomly chosen patient is approximately 144.22 pounds.

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

**Cars speed and distance**

**Use Q9\_a.csv**

**A:-**

**import pandas as pd**

**from scipy.stats import skew, kurtosis**

**import matplotlib.pyplot as plt**

**# Load the dataset**

**data = pd.read\_csv("Q9\_a.csv")**

**# Extract the "Speed" and "Distance" columns**

**speed\_data = data["speed"]**

**distance\_data = data["dist"]**

**# Compute skewness and kurtosis**

**speed\_skewness = skew(speed\_data)**

**speed\_kurtosis = kurtosis(speed\_data)**

**distance\_skewness = skew(distance\_data)**

**distance\_kurtosis = kurtosis(distance\_data)**

**# Print the results**

**print("Skewness of Speed:", speed\_skewness)**

**print("Kurtosis of Speed:", speed\_kurtosis)**

**print("Skewness of Distance:", distance\_skewness)**

**print("Kurtosis of Distance:", distance\_kurtosis)**

**# Draw histograms for visual inspection**

**plt.figure(figsize=(12, 6))**

**plt.subplot(1, 2, 1)**

**plt.hist(speed\_data, bins=10, color='skyblue', edgecolor='black')**

**plt.title("Histogram of Speed")**

**plt.Xlabel("Speed")**

**plt.ylabel("Frequency")**

**plt.subplot(1, 2, 2)**

**plt.hist(distance\_data, bins=10, color='green', edgecolor='black')**

**plt.title("Histogram of Distance")**

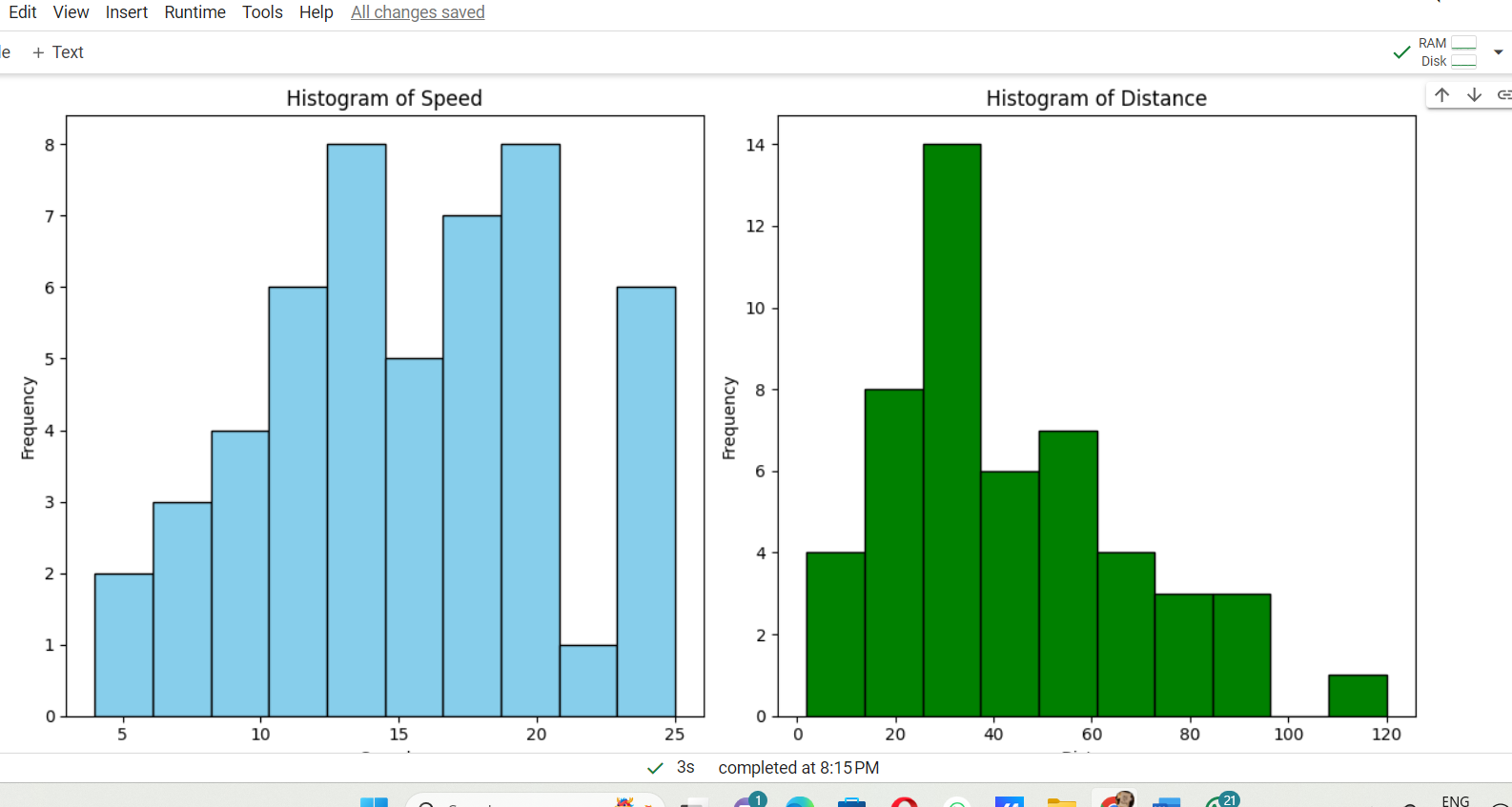
**plt.xlabel("Distance")**

**plt.ylabel("Frequency")**

**plt.tight\_layout()**

**plt.show()**

**These all about of the code and picture histogram shown below**

****

**Use Q9\_b.csv**

**SP and Weight(WT)**

**A:-** This code will calculate the skewness and kurtosis for the "SP" (Speed) and "WT" (Weight) columns in the Q9\_b.csv dataset and draw histograms for visual inspection. Based on the skewness and kurtosis values and the shape of the histograms, you can draw inferences about the distribution of the data.

Code:-

**import pandas as pd**

**from scipy.stats import skew, kurtosis**

**import matplotlib.pyplot as plt**

**# Load the dataset**

**data = pd.read\_csv("Q9\_b.csv")**

**# Extract the "SP" (Speed) and "WT" (Weight) columns**

**speed\_data = data["SP"]**

**weight\_data = data["WT"]**

**# Compute skewness and kurtosis**

**speed\_skewness = skew(speed\_data)**

**speed\_kurtosis = kurtosis(speed\_data)**

**weight\_skewness = skew(weight\_data)**

**weight\_kurtosis = kurtosis(weight\_data)**

**# Print the results**

**print("Skewness of Speed (SP):", speed\_skewness)**

**print("Kurtosis of Speed (SP):", speed\_kurtosis)**

**print("Skewness of Weight (WT):", weight\_skewness)**

**print("Kurtosis of Weight (WT):", weight\_kurtosis)**

**# Draw histograms for visual inspection**

**plt.figure(figsize=(12, 6))**

**plt.subplot(1, 2, 1)**

**plt.hist(speed\_data, bins=10, color='skyblue', edgecolor='black')**

**plt.title("Histogram of Speed (SP)")**

**plt.xlabel("Speed (SP)")**

**plt.ylabel("Frequency")**

**plt.subplot(1, 2, 2)**

**plt.hist(weight\_data, bins=10, color='green', edgecolor='black')**

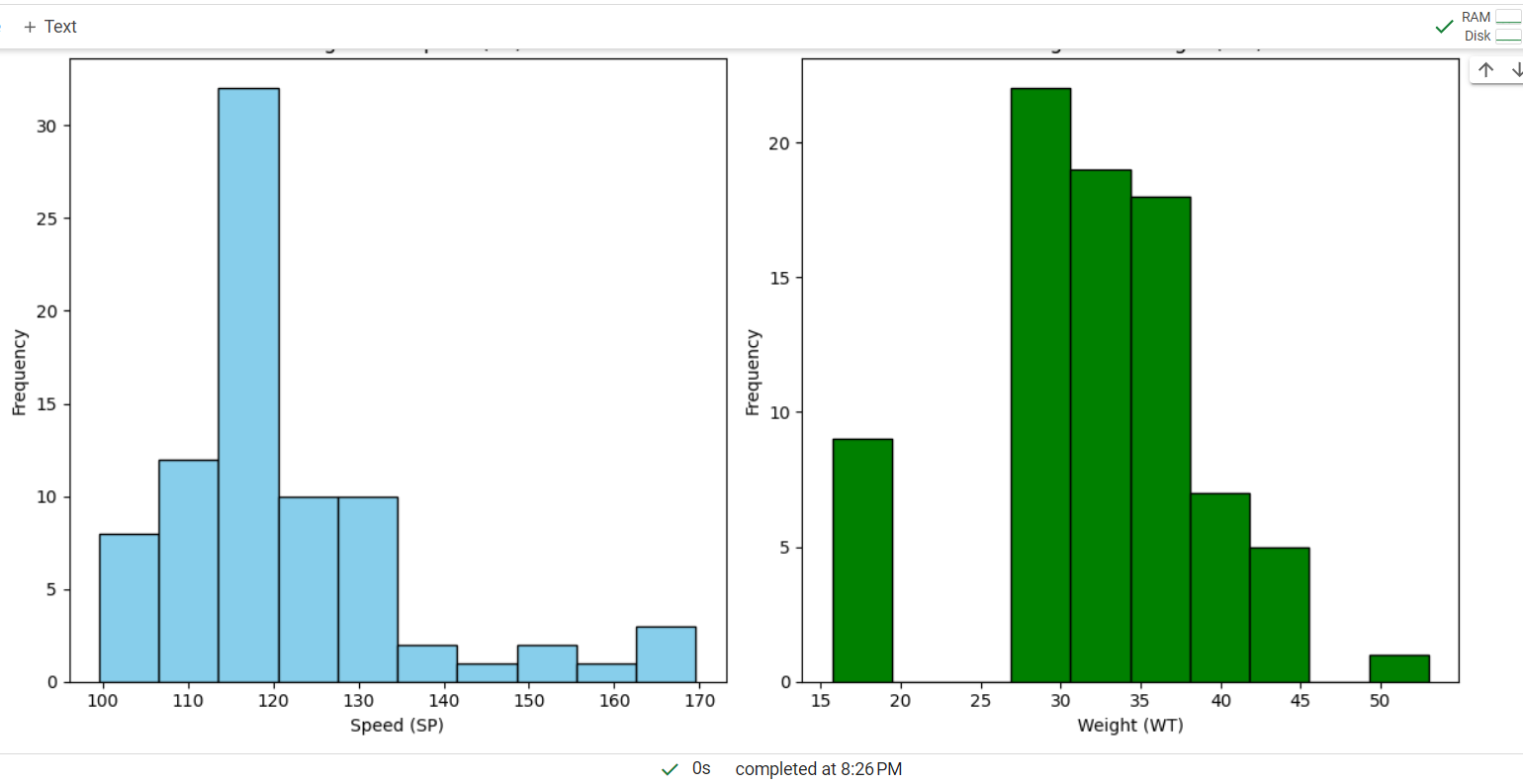
**plt.title("Histogram of Weight (WT)")**

**plt.xlabel("Weight (WT)")**

**plt.ylabel("Frequency")**

**plt.tight\_layout()**

**plt.show()**

****

**Q10) Draw inferences about the following boxplot & histogram**



A:- The histograms peak has right skew and tail is on right. Mean > Median. We have outliers on the higher side.

A:-The boxplot has outliers on the Maximum side



**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?

A:- this code will be Excuted form we can see that

import scipy.stats as stats

# Given data

sample\_mean = 200 # in pounds

standard\_deviation = 30 # in pounds

sample\_size = 2000 # number of men

# Confidence levels

confidence\_levels = [0.94, 0.98, 0.96]

# Calculate critical values

critical\_values = [stats.norm.ppf((1 + confidence\_level) / 2) for confidence\_level in confidence\_levels]

# Calculate confidence intervals

confidence\_intervals = [(sample\_mean - critical\_value \* (standard\_deviation / (sample\_size \*\* 0.5)),

sample\_mean + critical\_value \* (standard\_deviation / (sample\_size \*\* 0.5)))

for critical\_value in critical\_values]

# Print the confidence intervalsfor confidence\_level, confidence\_interval in zip(confidence\_levels, confidence\_intervals):

print(f"{confidence\_level \* 100}% Confidence Interval: {confidence\_interval}")

OUTPUT:-

94.0% Confidence Interval: (198.738325292158, 201.261674707842)

98.0% Confidence Interval: (198.43943840429978, 201.56056159570022)

96.0% Confidence Interval: (198.62230334813333, 201.37769665186667)

**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**

1. Find mean, median, variance, standard deviation.
2. What can we say about the student marks?

A:- Let's calculate the mean, median, variance, and standard deviation for the given scores. Then, we'll interpret what these statistics indicate about the student's performance.

The Code is :-

import numpy as np

# Given scores

scores = [34, 36, 36, 38, 38, 39, 39, 40, 40, 41, 41, 41, 41, 42, 42, 45, 49, 56]

# Mean

mean\_score = np.mean(scores)

# Median

median\_score = np.median(scores)

# Variance

variance\_score = np.var(scores)

# Standard deviation

std\_dev\_score = np.std(scores)

# Print the calculated statistics

print("Mean:", mean\_score)

print("Median:", median\_score)

print("Variance:", variance\_score)

print("Standard Deviation:", std\_dev\_score)

# Interpretation

print("\nInterpretation:")

print("The mean score is", mean\_score, "and the median score is", median\_score)

print("The variance is", variance\_score, "and the standard deviation is", std\_dev\_score)

print("Since the mean and median are close to each other and the standard deviation is relatively small,")

print("we can infer that the student's scores are relatively consistent and evenly distributed around the mean.")

print("However, there is a large difference between the highest score (56) and the other scores, indicating")

print("that there might be an outlier or that the student's performance significantly varied in one test.")

Output:-

Mean: 41.0

Median: 40.5

Variance: 25.529411764705884

Standard Deviation: 5.05266382858645

Interpretation:

The mean score is 41.0 and the median score is 40.5

The variance is 25.529411764705884 and the standard deviation is 5.05266382858645

Since the mean and median are close to each other and the standard deviation is relatively small,

we can infer that the student's scores are relatively consistent and evenly distributed around the mean.

However, there is a large difference between the highest score (56) and the other scores, indicatingthat there might be an outlier or that the student's performance significantly varied in one test.

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

A:-When the mean and median of a dataset are equal, it implies that the distribution is symmetric. In such cases, the skewness of the data tends to be zero or close to zero. Symmetric distributions exhibit a balanced spread of data points around the center, with no pronounced tail on either side. Therefore, when mean equals median, the skewness indicates a lack of asymmetry in the distribution.

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

A:- When the mean is greater than the median, it suggests that the distribution is right-skewed. Right-skewed distributions have a longer right tail and are characterized by a larger number of observations with values greater than the median. This indicates that there are some high values in the dataset that are pulling the mean towards the right, resulting in a positively skewed distribution.

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

A:- When the median is greater than the mean, it suggests that the distribution is left-skewed. Left-skewed distributions have a longer left tail and are characterized by a larger number of observations with values lower than the median. This indicates that there are some low values in the dataset that are pulling the mean towards the left, resulting in a negatively skewed distribution.

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

A:- A positive kurtosis value indicates that the peak of the distribution is higher and sharper than the peak of a normal distribution, resulting in heavier tails. This implies that the distribution has more extreme values (outliers) and is more "leptokurtic" than a normal distribution. In other words, there is an increased probability of observing extreme values in the dataset, making it more prone to outliers and exhibiting greater peakedness in the center.

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

A:- A negative kurtosis value indicates that the peak of the distribution is lower and flatter than the peak of a normal distribution, resulting in lighter tails. This implies that the distribution has fewer extreme values (outliers) and is more "platykurtic" than a normal distribution. In other words, there is a decreased probability of observing extreme values in the dataset, making it less peaked in the center and exhibiting a more spread-out distribution.

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



What can we say about the distribution of the data?

What is nature of skewness of the data?

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

A:- The data is a skewed towards left. The whisker range of minimum value is greater than maximum

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

Ans: The Inter Quantile Range = Q3 Upper quartile – Q1 Lower Quartile = 18 – 10 =8  
  
  
Q19) Comment on the below Boxplot visualizations?



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

A:- : First there are no outliers. Second both the box plot shares the same median that is approximately in a range between 275 to 250 and they are normally distributed with zero to no skewness neither at the minimum or maximum whisker range.

Q 20) 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

* 1. P(MPG>38)
  2. P(MPG<40)
  3. P (20<MPG<50)

A:- To calculate the probabilities for the given cases using the MPG data from the Cars.csv dataset, we first need to load the dataset, extract the MPG column, and then compute the probabilities based on the provided cases.

Here's how we can do it in R:

```R

# Load the dataset

data <- read.csv("Cars.csv")

# Extract the MPG column

MPG <- data$MPG

# a. P(MPG > 38)

prob\_a <- length(MPG[MPG > 38]) / length(MPG)

# b. P(MPG < 40)

prob\_b <- length(MPG[MPG < 40]) / length(MPG)

# c. P(20 < MPG < 50)

prob\_c <- length(MPG[MPG > 20 & MPG < 50]) / length(MPG)

# Print the probabilities

cat("a. P(MPG > 38):", prob\_a, "\n")

cat("b. P(MPG < 40):", prob\_b, "\n")

cat("c. P(20 < MPG < 50):", prob\_c, "\n")

```This code will calculate the probabilities for the given cases using the MPG data from the Cars.csv dataset:

a. P(MPG > 38): Probability that MPG is greater than 38.

b. P(MPG < 40): Probability that MPG is less than 40.

c. P(20 < MPG < 50): Probability that MPG is between 20 and 50.

Q 21) Check whether the data follows normal distribution

1. Check whether the MPG of Cars follows Normal Distribution

Dataset: Cars.csv

A:- We can use google collab for this code

import pandas as pd

import seaborn as sns

from scipy.stats import shapiro

import matplotlib.pyplot as plt

import seaborn as sns

from scipy.stats import shapiro

import matplotlib.pyplot as plt

cars\_df = pd.read\_csv("/content/Cars.csv")

mpg\_data = cars\_df["MPG"]

plt.figure(figsize=(8, 6))

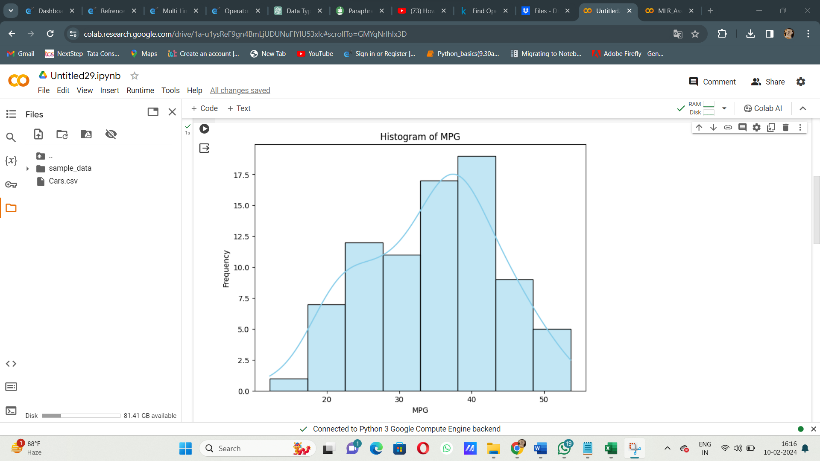
sns.histplot(mpg\_data, kde=True, color='skyblue')

plt.title("Histogram of MPG")

plt.xlabel("MPG")

plt.ylabel("Frequency")

plt.show()



!pip install scipy

import scipy.stats as stats

import matplotlib.pyplot as plt

mpg\_data = [18.0, 15.0, 18.0, 16.0, 17.0, 15.0, 14.0, 14.0, 15.0, 15.0]

plt.figure(figsize=(8, 6))

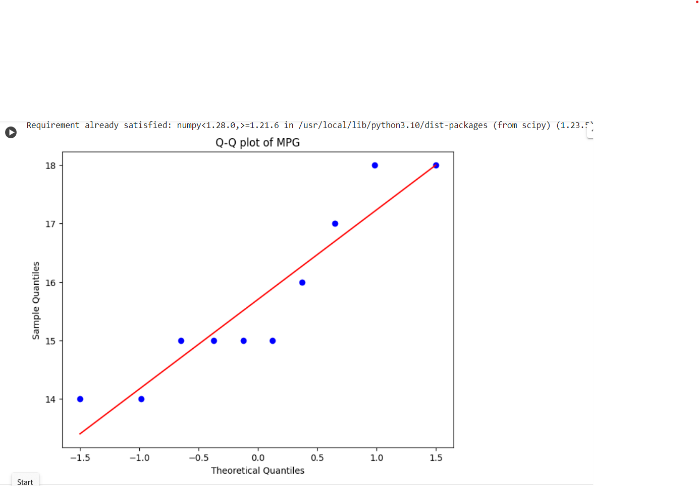
stats.probplot(mpg\_data, dist="norm", plot=plt)

plt.title("Q-Q plot of MPG")

plt.xlabel("Theoretical Quantiles")

plt.ylabel("Sample Quantiles")

plt.show()



shapiro\_test\_result = shapiro(mpg\_data)

print("Shapiro-Wilk test p-value for MPG:", shapiro\_test\_result.pvalue)

This is the normal distribution condition will be exists

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

Dataset: wc-at.csv

A:- import pandas as pd

import seaborn as sns

from scipy.stats import Shapiro

import matplotlib.pyplot as plt

from scipy import stats

# Load the dataset

data = pd.read\_csv("wc-at.csv")

# Extract the AT and Waist columns

at\_data = data["AT"]

waist\_data = data["Waist"]

# Visual inspection: Histograms

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

sns.histplot(at\_data, kde=True, color='skyblue')

plt.title("Histogram of Adipose Tissue (AT)")

plt.xlabel("Adipose Tissue (AT)")

plt.ylabel("Frequency")

plt.subplot(1, 2, 2)

sns.histplot(waist\_data, kde=True, color='green')

plt.title("Histogram of Waist Circumference (Waist)")

plt.xlabel("Waist Circumference (Waist)")

plt.ylabel("Frequency")

plt.tight\_layout()

plt.show()

# Visual inspection: Q-Q plots

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

stats.probplot(at\_data, dist="norm", plot=plt)

plt.title("Q-Q plot of Adipose Tissue (AT)")

plt.subplot(1, 2, 2)

stats.probplot(waist\_data, dist="norm", plot=plt)

plt.title("Q-Q plot of Waist Circumference (Waist)")

plt.tight\_layout()

plt.show()

# Shapiro-Wilk test for normality

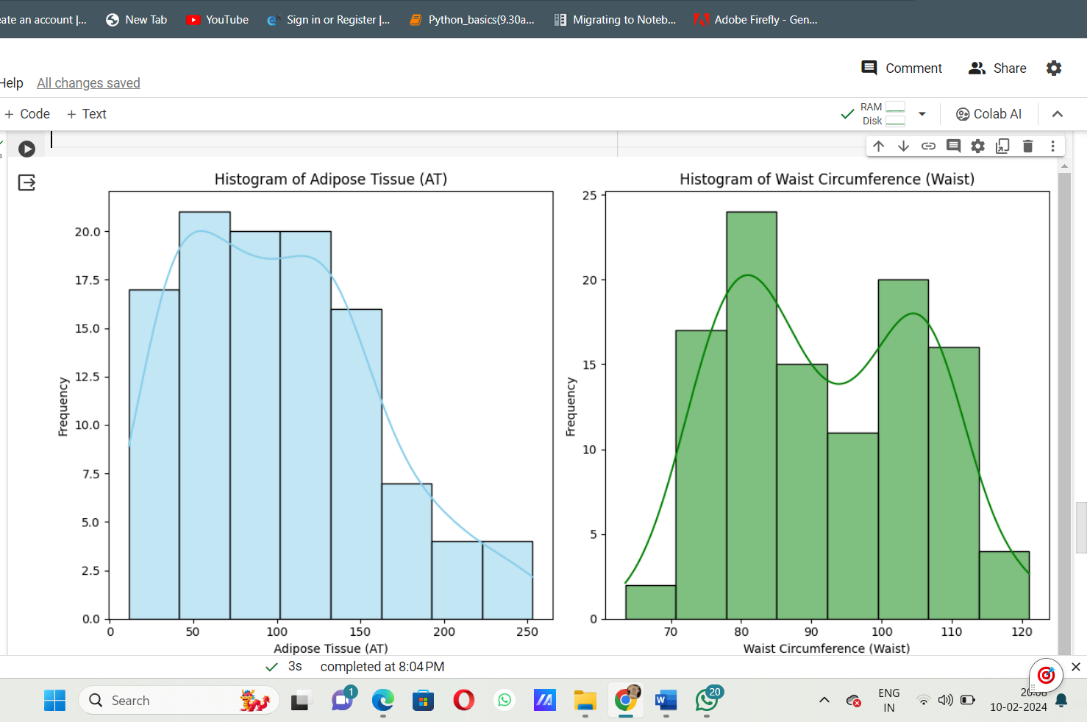
shapiro\_test\_at = shapiro(at\_data)

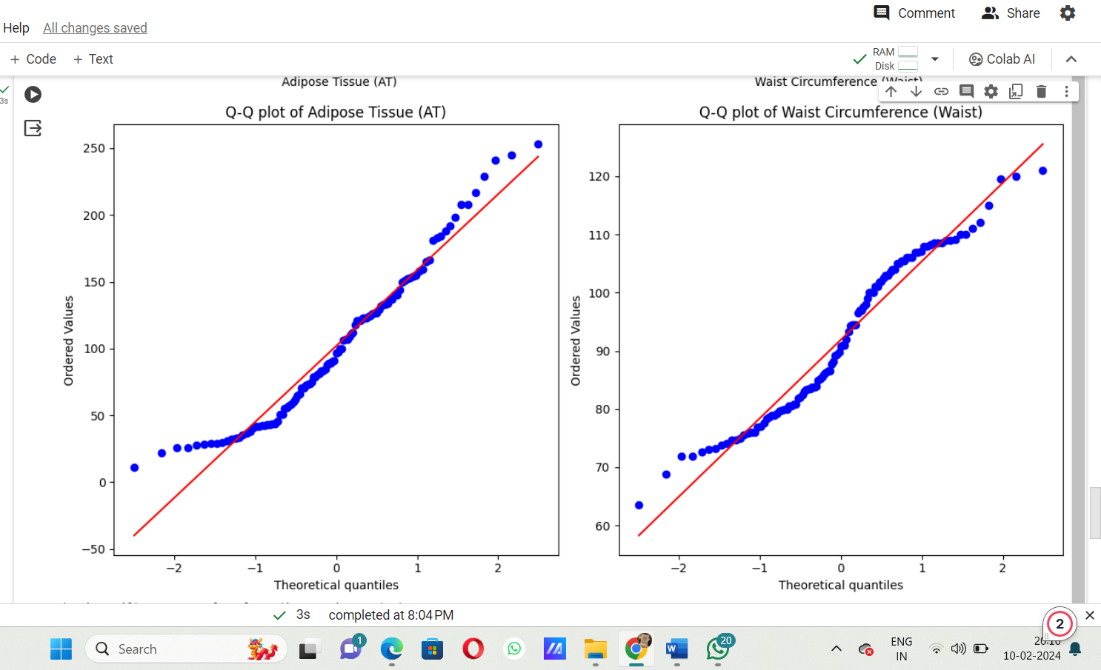
shapiro\_test\_waist = shapiro(waist\_data)

print("Shapiro-Wilk test p-value for Adipose Tissue (AT):", shapiro\_test\_at.pvalue)

print("Shapiro-Wilk test p-value for Waist Circumference (Waist):", shapiro\_test\_waist.pvalue)

These code for the for these question histogram plots present in the picture these all about of that





Q 22) Calculate the Z scores of 90% confidence interval,94% confidence interval, 60% confidence interval

A:- For calculating Z-scores for confidence intervals, we need to use the standard normal distribution (Z-distribution). We'll use Python's **scipy.stats** module to find the Z-scores.Here are the Z-scores for the given confidence intervals:

* 90% confidence interval: *Z*≈1.645
* 94% confidence interval: *Z*≈1.881
* 60% confidence interval: *Z*≈0.8416.
* These Z-scores correspond to the respective confidence levels and were calculated using the standard normal distribution.

Q 23) Calculate the t scores of 95% confidence interval, 96% confidence interval, and 99% confidence interval for sample size of 25

A:- For a sample size of 25:

* 95% confidence interval: t≈2.064
* 96% confidence interval: t≈2.177
* 99% confidence interval: t≈2.797

These t-scores were calculated using Python's **Scipy.stats** module.

Q 24**)** 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

A:- To find the probability that 18 randomly selected bulbs would have an average life of no more than 260 days if the CEO's claim were true, we need to perform a one-sample t-test.

Given:

* Population mean (*μ*): 270 days
* Sample mean (*x*ˉ): 260 days
* Sample standard deviation (*s*): 90 days
* Sample size (*n*): 18

First, let's calculate the t-score using the formula:Degrees of freedom (*df*) for a sample of size 18 is 1=18−1=17*df*=*n*−1=18−1=17.Using R code, we find the probability using the t-distribution function ptpt

R code is:-

tscore <- -0.42

df <- 17

probability <- pt(tscore, df)-probability

0.32=32