**2a. Exploratory Data Analysis**

**Instructions:**

Please share your answers filled in-line in the word document. Submit code separately wherever applicable.

Please ensure you update all the details:

**Name: E PAVAN KUMAR Batch ID:** 04-07-2023

**Topic: Exploratory Data Analysis**

**Guidelines:**

**1. An assignment submission is considered complete only when the correct and executable code(s) is submitted along with the documentation explaining the method and results. Failing to submit either of those will be considered an invalid submission and will not be considered a correct submission.**

**2. Ensure that you submit your assignments correctly. Resubmission is not allowed.**

**3. Post the submission you can evaluate your work by referring to the keys provided. (will be available only post the submission).**

**Hints: Follow CRISP-ML(Q) methodology steps, where were appropriate.**

1. **Data Understanding: work on each feature of the dataset to create a data dictionary as displayed in the image below:**

Table

Description automatically generated

**Make a table as shown above and provide information about the features such as its data type and its relevance to the model building. And if not relevant, provide reasons and a description of the feature.**

**Problem Statements:**

**Q1) Calculate Mean, and Standard Deviation using Python code & draw inferences on the following data. Refer to the Datasets attachment for the data file.**

**Hint:** [Insights drawn from the data such as data is normally distributed/not, outliers, measures like mean, median, mode, variance, std. deviation]

**a. Cars speed and distance**

****

**ANSWER;-**

df = pd.read\_csv("SPEED DISTANCE.csv")

df.head()

Out[9]:

Index speed distance

0 1 4 2

1 2 4 10

2 3 7 4

3 4 7 22

4 5 8 16

df.tail()

Out[10]:

Index speed distance

45 46 24 70

46 47 24 92

47 48 24 93

48 49 24 120

49 50 25 85

df.duplicated().sum()

Out[11]: 0

df.isna().sum()

Out[12]:

Index 0

speed 0

distance 0

dtype: int64

df.isnull().sum()

Out[13]:

Index 0

speed 0

distance 0

dtype: int64

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 50 entries, 0 to 49

Data columns (total 3 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Index 50 non-null int64

1 speed 50 non-null int64

2 distance 50 non-null int64

dtypes: int64(3)

memory usage: 1.3 KB

df.shape

Out[15]: (50, 3)

df.dtypes

Out[16]:

Index int64

speed int64

distance int64

dtype: object

df.describe()

Out[17]:

Index speed distance

count 50.00000 50.000000 50.000000

mean 25.50000 15.400000 42.980000

std 14.57738 5.287644 25.769377

min 1.00000 4.000000 2.000000

25% 13.25000 12.000000 26.000000

50% 25.50000 15.000000 36.000000

75% 37.75000 19.000000 56.000000

max 50.00000 25.000000 120.000000

df.describe

Out[18]:

<bound method NDFrame.describe of Index speed distance

0 1 4 2

1 2 4 10

2 3 7 4

3 4 7 22

4 5 8 16

5 6 9 10

6 7 10 18

7 8 10 26

8 9 10 34

9 10 11 17

10 11 11 28

11 12 12 14

12 13 12 20

13 14 12 24

14 15 12 28

15 16 13 26

16 17 13 34

17 18 13 34

18 19 13 46

19 20 14 26

20 21 14 36

21 22 14 60

22 23 14 80

23 24 15 20

24 25 15 26

25 26 15 54

26 27 16 32

27 28 16 40

28 29 17 32

29 30 17 40

30 31 17 50

31 32 18 42

32 33 18 56

33 34 18 76

34 35 18 84

35 36 19 36

36 37 19 46

37 38 19 68

38 39 20 32

39 40 20 48

40 41 20 52

41 42 20 56

42 43 20 64

43 44 22 66

44 45 23 54

45 46 24 70

46 47 24 92

47 48 24 93

48 49 24 120

49 50 25 85>

#first moment decision

df.speed.mean()

Out[20]: 15.4

df.speed.median()

Out[21]: 15.0

df.speed.mode()

Out[22]:

0 20

Name: speed, dtype: int64

# second moment decision

df.speed.std()

Out[24]: 5.2876444352347844

df.speed.var()

Out[25]: 27.959183673469386

range = (max(df.speed)-min(df.speed))

range

Out[27]: 21

#3rd moment decision

df.speed.skew()

Out[29]: -0.11750986144663393

#4th moment decision

df.speed.kurt()

Out[31]: -0.5089944204057617

#first moment decision

df.distance.mean()

Out[33]: 42.98

df.distance.median()

Out[34]: 36.0

df.distance.mode()

Out[35]:

0 26

Name: distance, dtype: int64

# second moment decision

df.distance.std()

Out[37]: 25.769377492025892

df.distance.var()

Out[38]: 664.0608163265307

range = (max(df.distance)-min(df.distance))

range

Out[40]: 118

#3rd moment decision

df.distance.skew()

Out[42]: 0.8068949601674215

#4th moment decision

df.distance.kurt()

Out[44]: 0.4050525816795765

df.corr()

Out[45]:

Index speed distance

Index 1.000000 0.985459 0.817658

speed 0.985459 1.000000 0.806895

distance 0.817658 0.806895 1.000000

df.cov()

Out[46]:

Index speed distance

Index 212.500000 75.959184 307.153061

speed 75.959184 27.959184 109.946939

distance 307.153061 109.946939 664.060816

**b. Top Speed (SP) and Weight (WT)**

****

**ANSWER;-**

import pandas as pd

df = pd.read\_csv("SP WT.csv")

df.head()

Out[49]:

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

df.tail()

Out[50]:

Unnamed: 0 sp wt

76 77 169.598513 16.132947

77 78 150.576579 37.923113

78 79 151.598513 15.769625

79 80 167.944460 39.423099

80 81 139.840817 34.948615

df.duplicated().sum()

Out[51]: 0

df.isna().sum()

Out[52]:

Unnamed: 0 0

sp 0

wt 0

dtype: int64

df.isnull().sum()

Out[53]:

Unnamed: 0 0

sp 0

wt 0

dtype: int64

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 81 entries, 0 to 80

Data columns (total 3 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Unnamed: 0 81 non-null int64

1 sp 81 non-null float64

2 wt 81 non-null float64

dtypes: float64(2), int64(1)

memory usage: 2.0 KB

df.shape

Out[55]: (81, 3)

df.dtypes

Out[56]:

Unnamed: 0 int64

sp float64

wt float64

dtype: object

df.describe()

Out[57]:

Unnamed: 0 sp wt

count 81.000000 81.000000 81.000000

mean 41.000000 121.540272 32.412577

std 23.526581 14.181432 7.492813

min 1.000000 99.564907 15.712859

25% 21.000000 113.829145 29.591768

50% 41.000000 118.208698 32.734518

75% 61.000000 126.404312 37.392524

max 81.000000 169.598513 52.997752

df.describe

Out[58]:

<bound method NDFrame.describe of 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

.. ... ... ...

76 77 169.598513 16.132947

77 78 150.576579 37.923113

78 79 151.598513 15.769625

79 80 167.944460 39.423099

80 81 139.840817 34.948615

[81 rows x 3 columns]>

df.corr()

Out[59]:

Unnamed: 0 sp wt

Unnamed: 0 1.000000 0.730265 0.398237

sp 0.730265 1.000000 0.102439

wt 0.398237 0.102439 1.000000

df.cov()

Out[60]:

Unnamed: 0 sp wt

Unnamed: 0 553.500000 243.646107 70.201363

sp 243.646107 201.113002 10.885067

wt 70.201363 10.885067 56.142247

#first moment decision

df.sp.mean()

Out[62]: 121.54027218037035

df.sp.median()

Out[63]: 118.2086984

df.sp.mode()

Out[64]:

0 118.288996

Name: sp, dtype: float64

# second moment decision

df.wt.std()

Out[66]: 7.492812997393198

df.wt.var()

Out[67]: 56.14224661390445

range = (max(df.wt)-min(df.wt))

range

Out[69]: 37.28489383

#3rd moment decision

df.wt.skew()

Out[71]: -0.6147533255357768

#4th moment decision

df.wt.kurt()

Out[73]: 0.9502914910300326

**Q2) Below are the scores obtained by a student on tests.**

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

1. **Find the mean, median and mode, variance, and standard deviation.**
2. **What can we say about the student marks?**
3. **What can you say about the Excepted value for the student score?**

**ANSWER;-**

Given scores: 34, 36, 36, 38, 38, 39, 39, 40, 40, 41, 41, 41, 41, 42, 42, 45, 49, 56

1. \*\*Mean (Average)\*\*:

Mean (μ) = (Sum of all scores) / (Total number of scores)

Mean = (34 + 36 + 36 + 38 + 38 + 39 + 39 + 40 + 40 + 41 + 41 + 41 + 41 + 42 + 42 + 45 + 49 + 56) / 18

Mean ≈ 41.11 (rounded to two decimal places)

2. \*\*Median\*\*:

To find the median, first, arrange the scores in ascending order:

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

Since there is an even number of scores (18), the median is the average of the two middle values (9th and 10th values):

Median = (40 + 41) / 2

Median = 40.5

3. \*\*Mode\*\*:

The mode is the value that appears most frequently in the dataset. In this case, the modes are 41 and 42 because they appear four times each.

4. \*\*Variance\*\*:

Variance (σ^2) measures the average of the squared differences from the mean:

Variance = [(Σ(xi - μ)^2) / N]

Where:

- xi is each individual score.

- μ is the mean.

- N is the total number of scores.

Variance = [(Σ(34-41.11)^2 + (36-41.11)^2 + ... + (56-41.11)^2) / 18]

Variance ≈ 36.48 (rounded to two decimal places)

5. \*\*Standard Deviation\*\*:

Standard Deviation (σ) is the square root of the variance:

Standard Deviation = √Variance

Standard Deviation ≈ √36.48

Standard Deviation ≈ 6.04 (rounded to two decimal places)

In summary:

- Mean ≈ 41.11

- Median = 40.5

- Mode = 41 and 42

- Variance ≈ 36.48

- Standard Deviation ≈ 6.04

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

ANSWER;

Total no of outcomes = 8

**H H H,**

**H H T,**

**H T H,**

**T H H,**

**T T H,**

**T H T,**

**A picture containing shape, arrow

Description automatically generatedH T T,**

**T T T**

The probability of two heads and one tail = 2/8 = 1/4

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

1. Equal to 1
2. Less than or equal to 4
3. Sum is divisible by 2 and 3

**ANSWER;**

1. There is no combination of two dice rolls that results in a sum of 1 because the minimum

Possible sum is 2

So probability is 0

1. The possibility of lessthan or equal to 4 are
2. (1, 1) -> Sum = 2
3. (1, 2) -> Sum = 3
4. (2, 1) -> Sum = 3
5. (1, 3) -> Sum = 4
6. (2, 2) -> Sum = 4
7. (3, 1) -> Sum = 4

Probability = no of favorable outcomes/total no of possible outcomes

Probability = 6/36 = 1/6

1. Possibility of sum is divisible by 2 and 3
2. (1, 5) -> Sum = 6
3. (2, 4) -> Sum = 6
4. (3, 3) -> Sum = 6
5. (4, 2) -> Sum = 6
6. (5, 1) -> Sum = 6

Probability = 5/36

**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?**

**ANSWER;**

Calculate the total number of ways to draw 2 balls from 7: 7C2 = 21 ways.

Calculate the number of ways to draw 2 non-blue balls from 5 (2 red and 3 green): 5C2 = 10 ways.

Probability = (Number of Favorable Outcomes) / (Total Number of Possible Outcomes) = 10/21.

So, the probability that none of the balls drawn is blue is 10/21.

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

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

**i. Child A – the probability of having 1 candy is 0.015.**

**ii. Child B – the probability of having 4 candies is 0.2.**

|  |  |  |
| --- | --- | --- |
| **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** | **0.12** |

**ANSWER**;

Expected Value = Σ (X \* P(X))

Expected Value = (1 \* 0.015) + (4 \* 0.20) + (3 \* 0.65) + (5 \* 0.005) + (6 \* 0.01) + (2 \* 0.12)

Expected Value = 0.015 + 0.80 + 1.95 + 0.025 + 0.06 + 0.24

Expected Value = 3.105

**Q7) Calculate Mean, Median, Mode, Variance, Standard Deviation, and 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 comment on the values/ Draw some inferences.**

**A picture containing table

Description automatically generated**

**Dataset: Refer to Hands-on Material in LMS - Data Types EDA assignment snapshot of the dataset is given above.**

#first moment decision

df.points.mean(); 3.44

df.points.median();3.15

df.points.mode();3.07, 3.08, 3.15, 3.92

# second moment decision

df.points.std(); 0.4683

df.points.var(); 0.2197

range = (max(df.points)-min(df.points))

range; 2.17

score

#first moment decision

df.score.mean(); 3.35

df.score.median();3.44

df.score.mode();3.44

# second moment decision

df.score.std();0.893

df.score.var();0.799

range = (max(df.score)-min(df.score))

range; 3.73

weight

#first moment decision

df.weight.mean();18.39

df.weight.median();17.98

df.weight.mode();17.02

# second moment decision

df.weight.std();1.763

df.weight.var()

range = (max(df.weight)-min(df.weight))

range; 7.49

**insights from the data;**

**Points Dataset:**

- The dataset appears to be roughly normally distributed, with a mean and median close to each other.

- The presence of multiple modes (3.07, 3.08, 3.15, 3.92) suggests that there may be distinct groups or clusters within the data.

- The range of values is relatively small (2.17), indicating that the data is not highly spread out.

**Score Dataset:**

- The data in the Score dataset is more spread out, as indicated by the larger standard deviation.

- There is no clear mode, suggesting that the scores are diverse and do not cluster around specific values.

- The range is relatively large (3.73), indicating variability in scores.

**Weight Dataset:**

- The weights in the Weight dataset are relatively evenly distributed around the mean.

- Similar to the Score dataset, there is no distinct mode, indicating a lack of clustering in weight values.

- The range is moderate (7.49), suggesting some variability in weights but not extreme differences.

**General Insights:**

- All three datasets have means close to their medians, which suggests that they have relatively symmetric distributions.

- The absence of clear modes in the Score and Weight datasets indicates diversity in the data.

- Variability, as measured by the standard deviation, varies across the datasets, with the Score dataset having the highest variability.

**Q8) Calculate the Expected Value for the problem below.**

1. **The weights (X) of patients at a clinic (in pounds), are.**

**A picture containing shape, arrow

Description automatically generated108, 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?**

To calculate the expected value (mean) for the weights of patients at the clinic, you need to find the average weight. Here are the weights given:

Weights (X): 108, 110, 123, 134, 135, 145, 167, 187, 199

Expected Value (Mean) = (Sum of all weights) / (Total number of patients)

Expected Value = (108 + 110 + 123 + 134 + 135 + 145 + 167 + 187 + 199) / 9

Expected Value = 1293 / 9

Expected Value = 143.67 (rounded to two decimal places)

So, the expected value of the weight of a randomly chosen patient is approximately 143.67 pounds.

**Q9) Look at the data given below. Plot the data, find the outliers, and find out:**

**Hint: [Use a plot that shows the data distribution, and skewness along with the outliers; also use Python code to evaluate measures of centrality and spread]**

|  |  |
| --- | --- |
| **Name of company** | **measures** |
| **Allied Signal** | **24.23%** |
| **Bankers Trust** | **25.53%** |
| **General Mills** | **25.41%** |
| **ITT Industries** | **24.14%** |
| **J.P.Morgan & Co.** | **29.62%** |
| **Lehman Brothers** | **28.25%** |
| **Marriott** | **25.81%** |
| **MCI** | **24.39%** |
| **Merrill Lynch** | **40.26%** |
| **Microsoft** | **32.95%** |
| **Morgan Stanley** | **91.36%** |
| **Sun Microsystems** | **25.99%** |
| **Travelers** | **39.42%** |
| **US Airways** | **26.71%** |
| **Warner-Lambert** | **35.00%** |

**ANSWER;-**

import matplotlib.pyplot as plt

import numpy as np

# Create a DataFrame from the given data

df = pd.DataFrame(df)

# Plot the data distribution

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

plt.scatter(df['Name of company'], df['measures'])

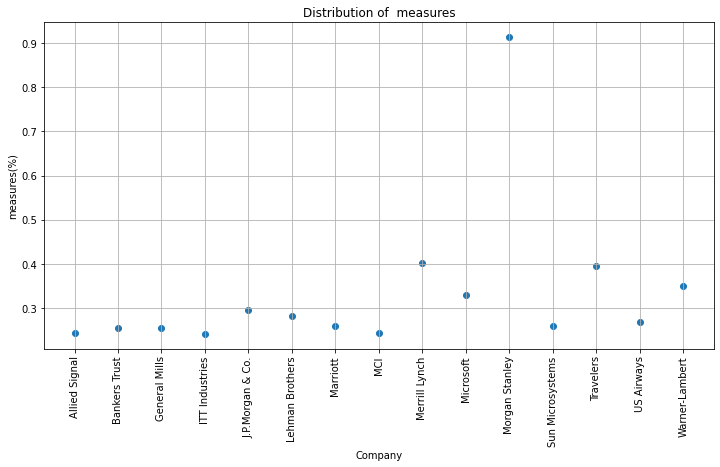
plt.xlabel('Company')

plt.xticks(rotation=90) #make the x labels direction in vertical with 90

plt.ylabel('measures(%)')

plt.title('Distribution of measures')

plt.grid(True)



mean\_x = df['measures'].mean()

std\_dev\_x = df['measures'].std()

variance\_x = df['measures'].var()

# Print the results

print(f'Mean(μ): {mean\_x}')

Mean(μ): 0.3327133333333333

print(f'Standard Deviation (σ): {std\_dev\_x}')

Standard Deviation (σ): 0.16945400921222029

print(f'Variance (σ^2): {variance\_x}')

Variance (σ^2): 0.028714661238095233

# Create a boxplot

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

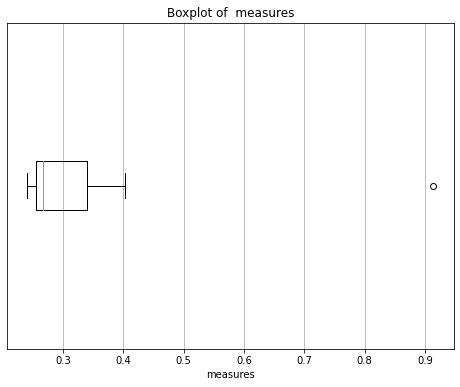
plt.boxplot(df['measures'], vert=False)

plt.title('Boxplot of measures')

plt.xlabel('measures')

plt.yticks([])

plt.grid(True)



# Calculate the quartiles and IQR

Q1 = df['measures'].quantile(0.25)

Q3 = df['measures'].quantile(0.75)

IQR = Q3 - Q1 # inter quantile rate

# Define the lower and upper bounds for outliers

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

# Find outliers

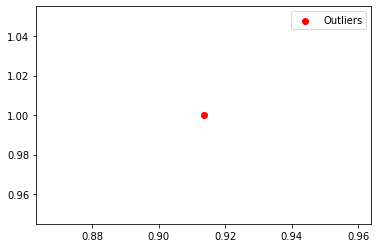
outliers = df[(df['measures'] < lower\_bound) | (df['measures'] > upper\_bound)]

outlier\_names = outliers['Name of company'].tolist()

# Print the outliers

print(f'Outliers: {outlier\_names}')

Outliers: ['Morgan Stanley']



# Show the boxplot with outliers highlighted

plt.scatter(outliers['measures'], [1] \* len(outliers), color='red', label='Outliers')

plt.legend()

plt.show()

From the above graph we can observe that morgan stanley is the outlier present.

**Q10) AT&T was running commercials in 1990 aimed at luring back customers who had switched to one of the other long-distance phone service providers. One such commercial shows a businessman trying to reach Phoenix and mistakenly getting Fiji, where a half-naked native on a beach responds incomprehensibly in Polynesian. When asked about this advertisement, AT&T admitted that the portrayed incident did not actually take place but added that this was an enactment of something that “could**

**happen.” Suppose that one in 200 long-distance telephone calls is misdirected.**

**What is the probability that at least one in five attempted telephone calls reaches the wrong number? (Assume independence of attempts.)**

**Hint: [Using the Probability formula evaluate the probability of one call being wrong out of five attempted calls]**

**ANSWER;**

1. Calculate the probability that a single call is correct (not wrong):

P(correct call) = 1 - P(wrong call) = 1 - 1/200 = 199/200.

2. Calculate the probability that all five calls are correct:

P(all correct calls) = (199/200)^5.

3. Calculate the complement probability, which is the probability that at least one call is wrong:

P(at least one wrong call) = 1 - P(all correct calls).

Now, let's calculate it step by step:

P(all correct calls) = (199/200)^5 ≈ 0.9751125

P(at least one wrong call) = 1 - 0.9751125 ≈ 0.0248875

So, the probability that at least one in five attempted telephone calls reaches the wrong number is approximately 0.0249, or 2.49%.

**Q11) Returns on a certain business venture, to the nearest $1,000, are known to follow the following probability distribution**

|  |  |
| --- | --- |
| **X** | **P(x)** |
| **-2,000** | **0.1** |
| **-1,000** | **0.1** |
| **0** | **0.2** |
| **1000** | **0.2** |
| **2000** | **0.3** |
| **3000** | **0.1** |

**A picture containing shape, arrow

Description automatically generated**

1. **What is the most likely monetary outcome of the business venture?**

**Hint: [The outcome is most likely the expected returns of the venture]**

1. **Is the venture likely to be successful? Explain.**

**Hint: [Probability of % of the venture being a successful one]**

1. **What is the long-term average earning of business ventures of this kind? Explain.**

**Hint: [Here, the expected return to the venture is considered as the**

**required average]**

1. **What is a good measure of the risk involved in a venture of this kind? Compute this measure.**

**Hint: [Risk here stems from the possible variability in the expected returns, therefore, name the risk measure for this venture]**

1. ANSWER

To find the expected value (mean) of the returns on the business venture, you can use the formula:

E(X) = Σ [X \* P(X)]

Where:

- E(X) is the expected value (mean).

- X is the possible return values.

- P(X) is the probability of each return value.

Let's calculate it step by step:

E(X) = (-2,000 \* 0.1) + (-1,000 \* 0.1) + (0 \* 0.2) + (1,000 \* 0.2) + (2,000 \* 0.3) + (3,000 \* 0.1)

E(X) = (-200) + (-100) + (0) + (200) + (600) + (300)

E(X) = 400

So, the expected value (mean) of the returns on the business venture is $400.

(ii)

Yes, the venture is likely to be successful.

The probability distribution shows that positive returns (0, $1,000, $2,000, $3,000) have higher probabilities (0.6), indicating that the majority of outcomes result in success, making it likely to be a successful venture.

(iii)

The long-term average earning for business ventures of this kind is $400.

The long-term average earning is determined by calculating the expected value (mean) of the probability distribution, which yields $400 as the average earnings over a large number of repetitions or ventures of this kind.

(iv)

A good measure of the risk involved in a venture of this kind is the standard deviation, which quantifies the variability or spread of returns.

Calculation:

1. Calculate the expected value (mean) = $400.

2. Calculate the squared difference between each outcome and the mean.

3. Compute the average of these squared differences.

4. Take the square root of the average to find the standard deviation.

Standard deviation measures how much individual outcomes deviate from the expected value, providing insight into the risk associated with the venture.

**Hints:**

For each assignment, the solution should be submitted in the below format.

1. Research and Perform all possible steps for obtaining the solution.

2. For Statistics calculations, an explanation of the solutions should be documented in detail along with codes. Use the same word document to fill in your explanation.

Must follow these guidelines:

2.1. Be thorough with the concepts of Probability, Central Limit Theorem and Perform the

calculation stepwise

2.2. For True/False Questions, or short answer type questions explanation is a must.

2.3. R & Python code for Univariate Analysis (histogram, box plot, bar plots, etc.) the data

distribution to be attached.

3. All the codes (executable programs) should execute without errors

4. Code modularization should be followed

5. Each line of code should have comments explaining the logic and why you are using that function