**CSL227**

Lab Practical Report

A logo of a university

Description automatically generated

**Faculty Name**: **Dr. Shaveta Arora** **Student name**: Arjun Bhardwaj

**Roll No**.: 21csu211

**Semester:** 8

**Group:** AA

Department of Computer Science and Engineering

The NorthCap University, Gurugram- 122001, India

Session 2024-25

**Student Name:**

**Student Roll No.:**

**INDEX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | **Experiment Title** | **Date of Experiment** | **Date of Submission** | **Sign student** | **CO Covered** | **Sign Faculty** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

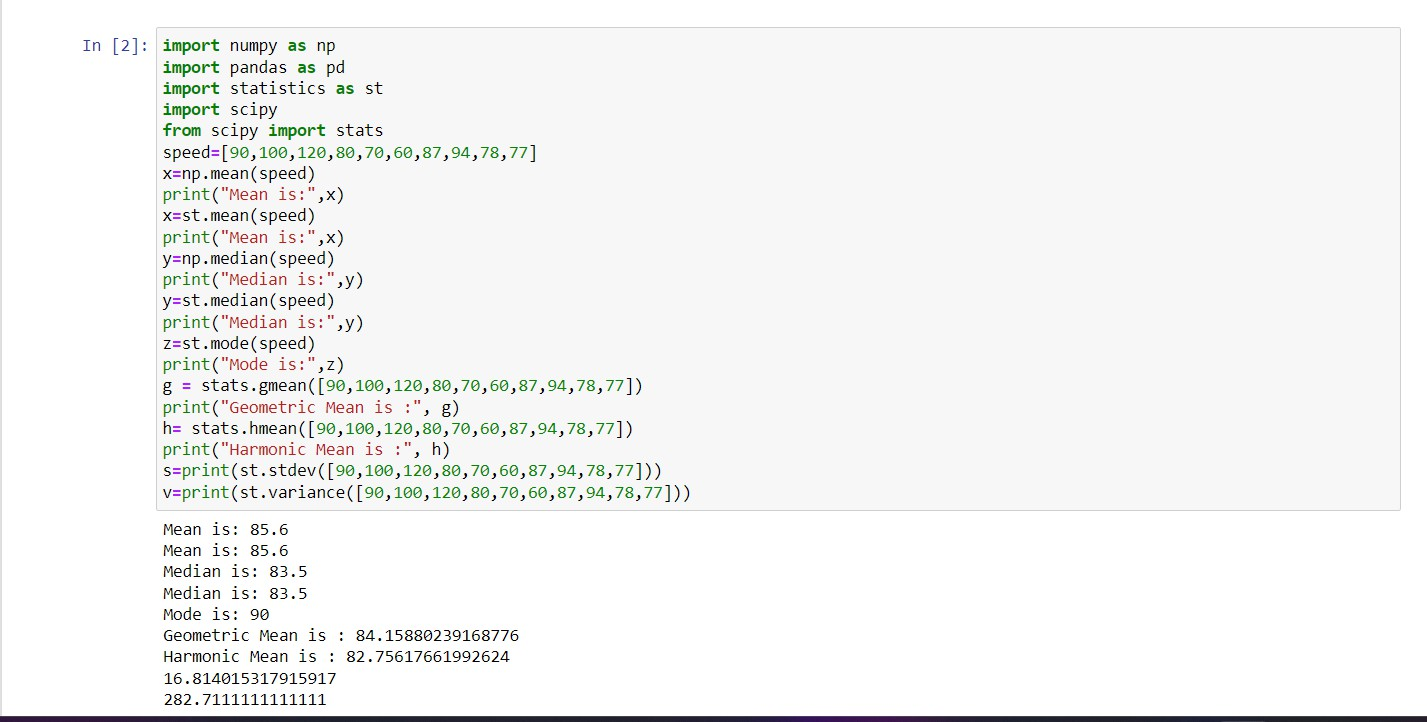
**EXPERIMENT NO. 1**

|  |
| --- |
| **Student Name and Roll Number:** Arjun Bhardwaj, 21csu211 |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks/Grade:** |

|  |
| --- |
| **Objective(s):**   * Familiarization with statistics libraries in Python |
| **Outcome:** Revision of the libraries NumPy, Pandas, Matplotlib, Seaborn and exploration of Statistics and SciPy. Statistics |
| **Problem Statement:** Introduction to statistics using Python |
| **Background Study:**  Python has libraries like Statistics and SciPy. Statistics which contain functions for several descriptive and inferential statistics tasks which can be of help to the students. |
| **Question Bank:**  1.What is difference between differential and inferential statistics?  Ans:  Descriptive Statistics is that branch of statistics which is concerned with describing the population under study.  It explains the data, which is already known, to summarize sample. Inferential Statistics  Inferential Statistics is a type of statistics, that focuses on drawing conclusions about the population, on the basis of sample analysis and observation.  It attempts to reach the conclusion to learn about the population, that extends beyond the data available  2. What is the relevance of statistics in everyday life?  Ans: Statistics is an integral part of our lives. It is used in the workplace and everyday life. In the workplace, statistics are often used to analyse what works best for a company‟s marketing strategy or how to distribute work among employees. In daily life, statistics can be used to analyse what food you should buy at the grocery store or how much money you spend on purchasing each week.  Statistics are everywhere, and they help us make sense of the world around us.  3. What are some of the common tasks you can accomplish using the statistics libraries you have studied in Python?  Python statistics libraries are comprehensive, popular, and widely used tools that will assist you in working with data.  Some common tasks that can be accomplished using these libraries are:  Descriptive statistics: With libraries like NumPy and Pandas, you can easily calculate various descriptive statistics such as mean, median, mode, variance, standard deviation, and quartiles. Hypothesis testing: Libraries like SciPy provide various statistical tests.  Data visualization: Matplotlib and Seaborn libraries can be used to create various types of plots and charts to visualize data, such as histogram, scatter plot, line charts, box plots.  4. What is data normalization, and why is it necessary in statistical modelling? How can Python libraries help normalize data?  Data normalization scales data to a standard range (e.g., 0–1 or mean=0, std=1), ensuring features contribute equally in statistical models.  It’s necessary to:  Improve model accuracy  Speed up convergence in algorithms  Prevent bias due to varying feature scales  Python libraries like scikit-learn provide tools like MinMaxScaler and StandardScaler for easy normalization.  5. What is the purpose of setting a random seed using numpy.random.seed() or random.seed()?  Setting a random seed (e.g., with numpy.random.seed() or random.seed()) ensures reproducibility—the same random numbers are generated every time the code runs, which is crucial for debugging, testing, and consistent results.2  6. How can you shuffle the elements of a list randomly in Python using random.shuffle() and numpy.random.permutation()? What are the differences between these methods?  random.shuffle(list) shuffles the list in place (modifies the original list).  numpy.random.permutation(list) returns a new shuffled array and leaves the original list unchanged.  Key diffeence:  shuffle() → in-place  permutation() → returns a new object  Use permutation() if you need the original data preserved.  7. How do you check the data type of elements in a NumPy array and convert an integer array to a float array?  import numpy as np  arr = np.array([1, 2, 3])  print(arr.dtype) # int64  float\_arr = arr.astype(float)  print(float\_arr.dtype) # float64  8. How can you cast a NumPy array of strings to integers? What errors might occur during this process?  import numpy as np  arr = np.array(["1", "2", "3"])  int\_arr = arr.astype(int)  9. How can you check for and ensure type compatibility between two arrays before performing arithmetic operations?  import numpy as np  a = np.array([1, 2, 3])  b = np.array([1.0, 2.0, 3.0])  # Check types  print(a.dtype, b.dtype)  # Convert for compatibility  a = a.astype(float)  # Now safe to operate  result = a + b  10. How do you select specific rows and columns from a DataFrame using loc [] and iloc[]? Explain the difference between them.  import pandas as pd  df = pd.DataFrame({"A": [1, 2], "B": [3, 4]}, index=["x", "y"])  df.loc["x", "A"] # label-based: value at row 'x', column 'A'  df.iloc[0, 0] # index-based: value at first row, first column |
|  |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**



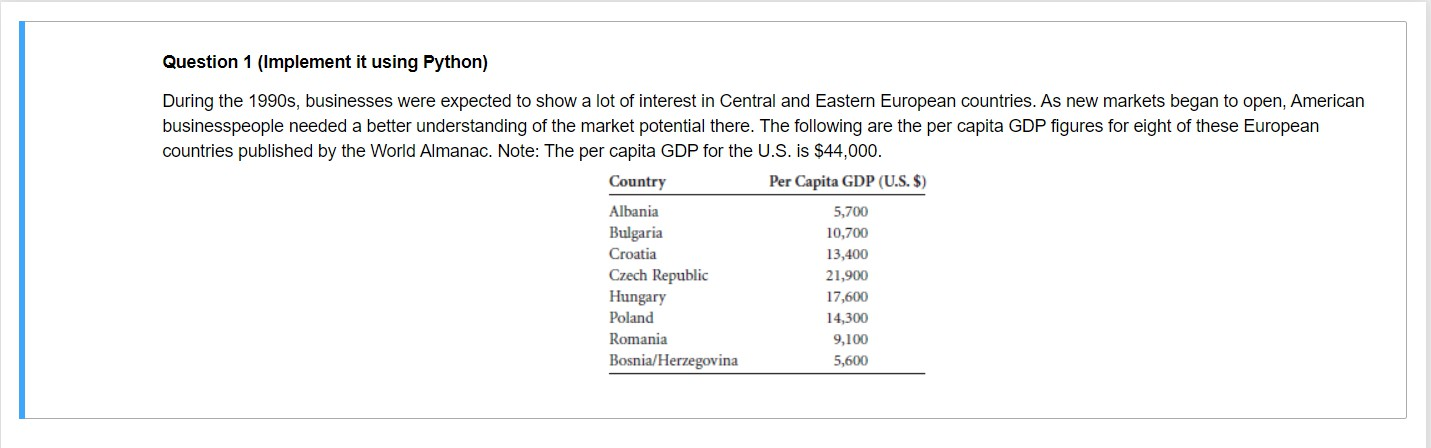
**EXPERIMENT NO. 2**

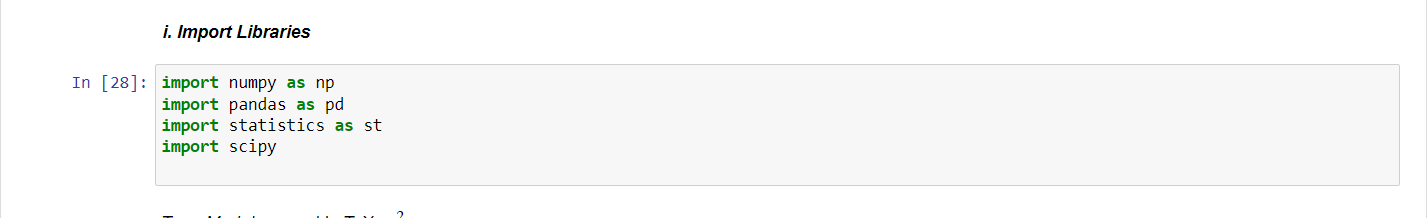
|  |
| --- |
| **Student Name and Roll Number:** Arjun Bhardwaj 21csu211 |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks/Grade:** |

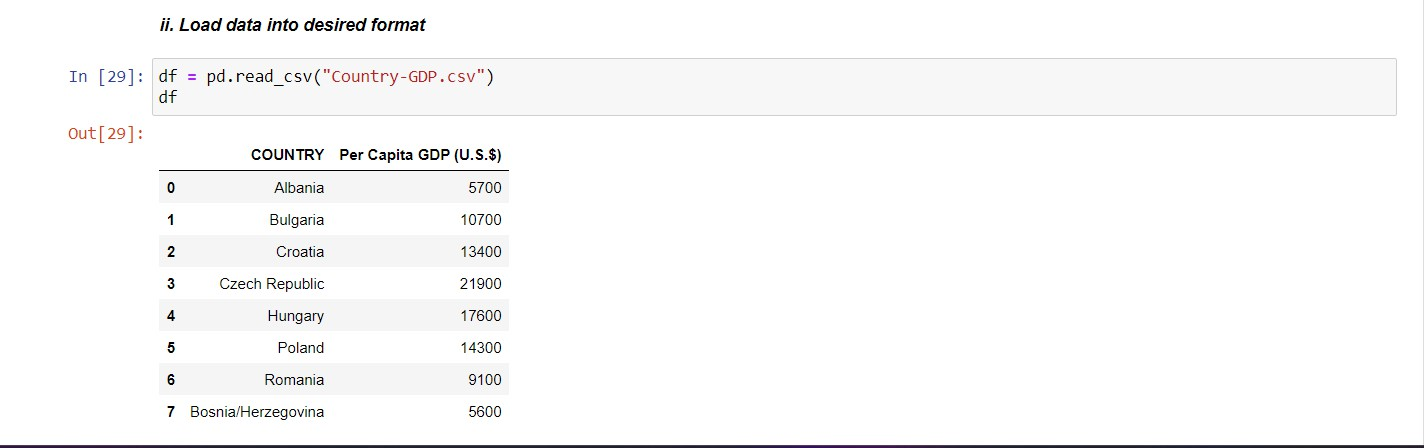
|  |
| --- |
| **Objective(s):**  Find mean, median, mode, standard deviation for a given dataset |
| **Outcome:** Measure of central tendency, measure of variation and measure of position |
| **Problem Statement:** Demonstration of descriptive statistical measures |
| **Background Study:**  Descriptive statistics are brief descriptive coefficients that summarize a given data set, which can be either a representation of the entire or a sample of a population. Descriptive statistics are broken down into measures of central tendency and measures of variability (spread). Measures of central tendency include the mean, median and mode, while measures of variability include standard deviation, variance, minimum and maximum variables. |
| **Question Bank:**  1.What is difference between mean, mode median?  Ans: Mean-The average taken for a set of numbers is called a mean.  Mode- The number that occurs the most in a given list of numbers is called a mode. Median- The middle value in the data set is called the Median.  2. What is advantage of median over mean?  Ans: The median is less affected by outliers and skewed data than the mean and is usually the preferred measure of central tendency when the distribution is not symmetrical.  3.Is mean affected by presence of outlier in data?  Yes  4. How to find the deviation of data in given dataset?    Step 1: Find the mean.  Step 2: For each data point, find the square of its distance to the mean. Step 3: Sum the values from Step 2.  Step 4: Divide by the number of data points. Step 5: Take the square root.  5. How would missing values affect the calculation of the mean, median, and mode? How can Python handle these missing values during computation?  df.mean(skipna=True)  df.median(skipna=True)  df.mode(dropna=True)  df.dropna()  6. How can pandas.DataFrame.describe() be used to quickly generate summary statistics including mean and median?  import pandas as pd  df = pd.DataFrame({"A": [1, 2, 3, 4, 5]})  print(df.describe())  7. In which scenarios would you prefer the median over the mean to describe the central tendency of a dataset?  Prefer the median when the dataset has outliers or is skewed, as it is less affected by extreme values than the mean.  Example In income data, median gives a better sense of the "typical" value than the mean.  8. How would you apply mean, median, and mode in real-world data analysis tasks such as customer transaction data or website traffic analysis?  Mean: Find average transaction value or website visits, but can be skewed by outliers.  Median: Use for typical transaction value or traffic, unaffected by extremes.  Mode: Identify most frequent transaction or popular page visits.  Example:  Transactions: Mean for average spend, median for typical value, mode for common amounts.  Traffic: Mean for average visits, median for typical traffic, mode for peak times.  9. In an online store, daily sales vary significantly. Would you use the mean or median to summarize daily sales, and why?  Use the \*\*median\*\* to summarize daily sales, as it is less affected by extreme values (e.g., a few high-sales days) and provides a better representation of typical daily sales.  10. For analysing monthly temperature data, would you use the mean, median, or mode to represent the central temperature trend? How would seasonal extremes affect your choice?    Use the mean to represent the central temperature trend, as it reflects overall trends. However, seasonal extremes (e.g., hot summer or cold winter days) can skew the mean, so if outliers are significant, the median might be a better choice for a more accurate reflection of typical temperatures. |

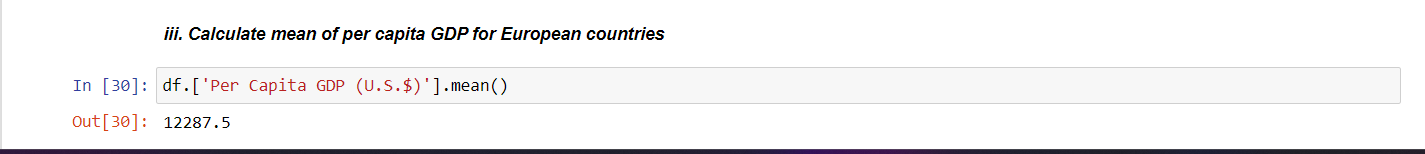
**Student Work Area**

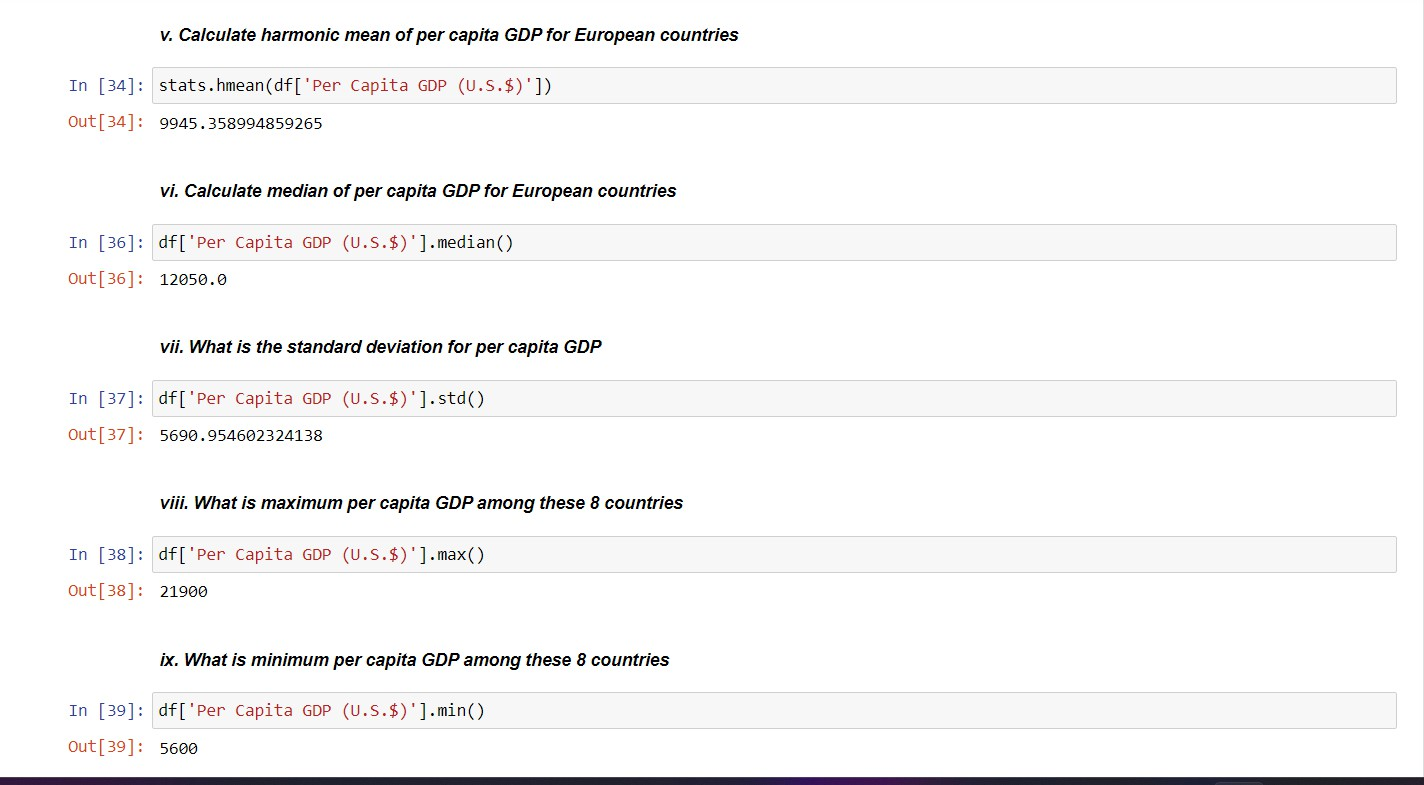
**Algorithm/Flowchart/Code/Sample Outputs**

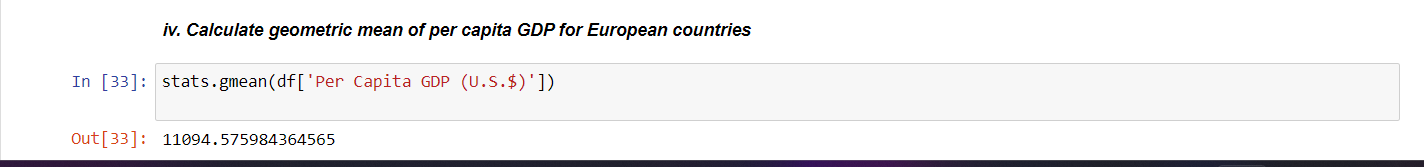


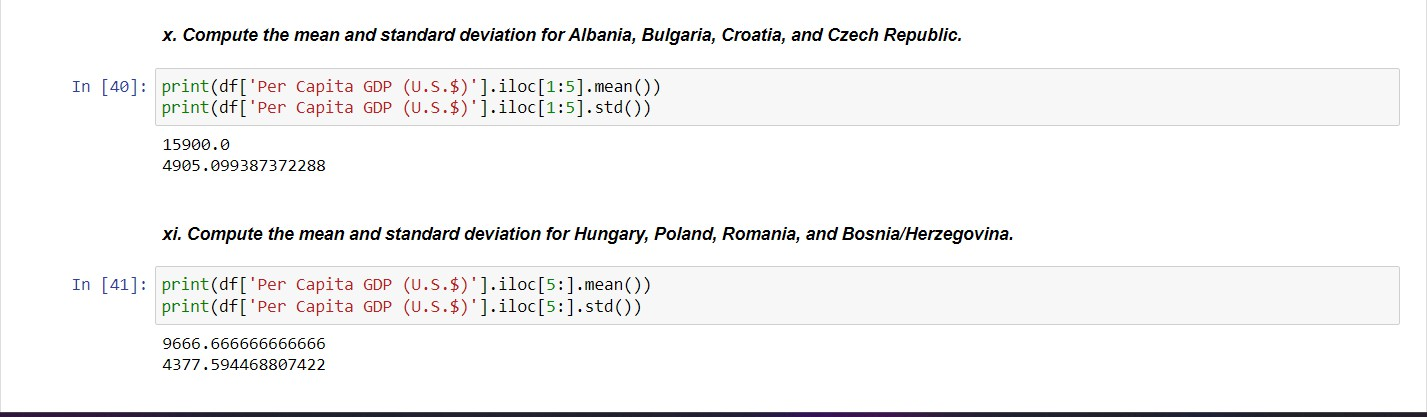


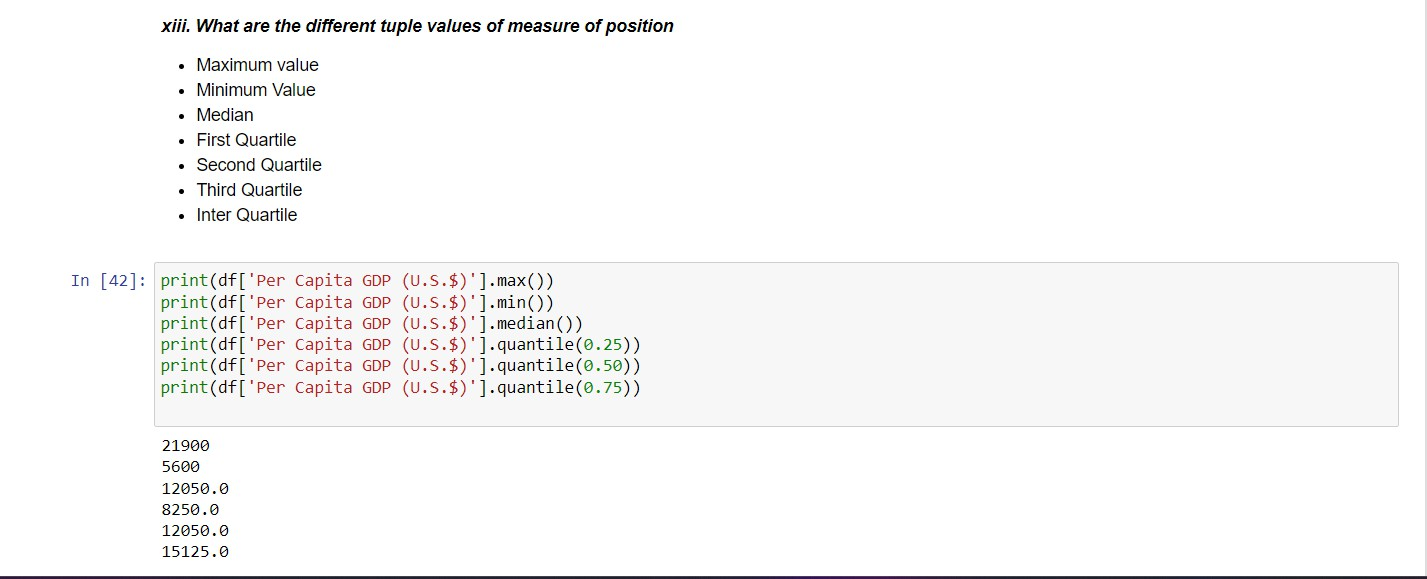














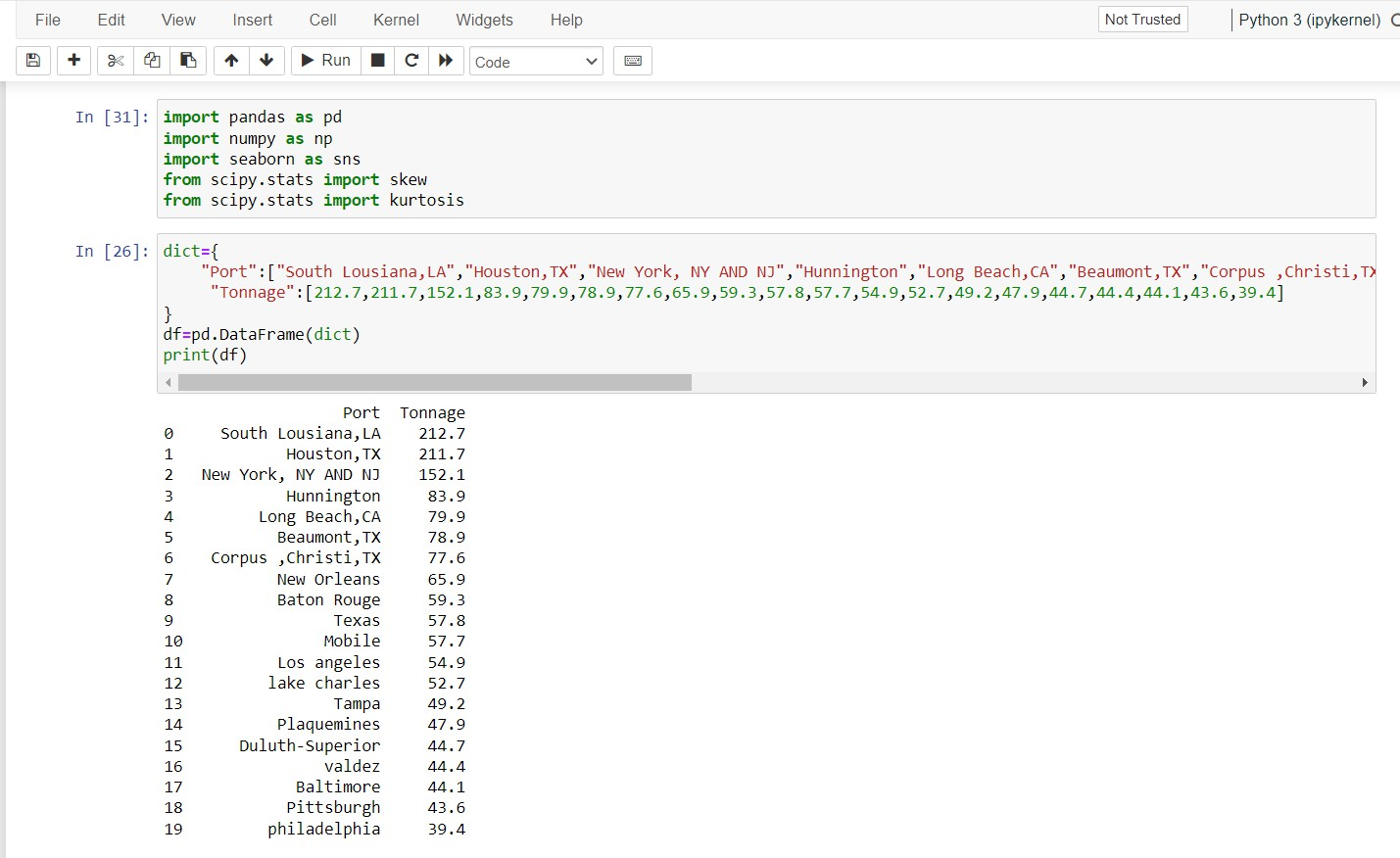
**EXPERIMENT NO. 3**

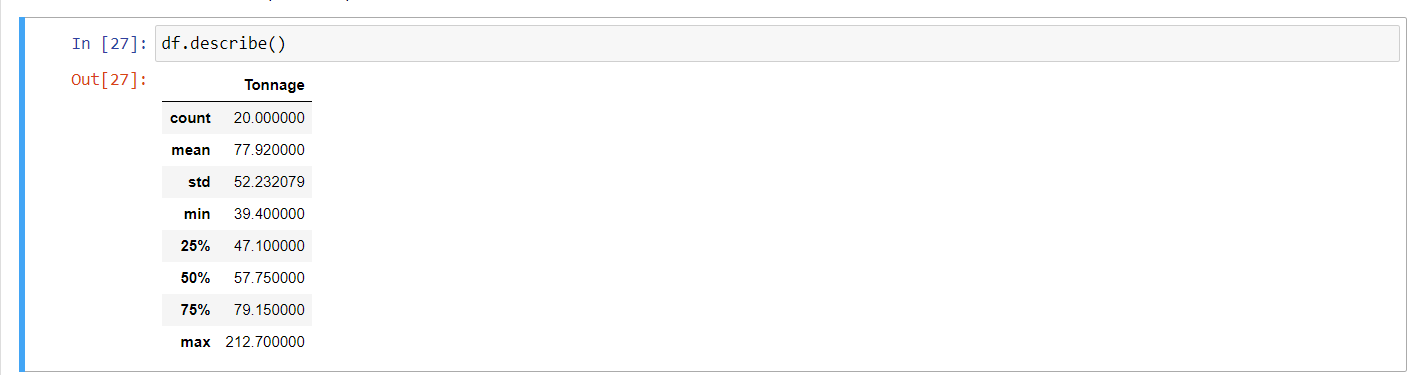
|  |
| --- |
| **Student Name and Roll Number:** Arjun Bhardwaj 21csu211 |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks/Grade:** |

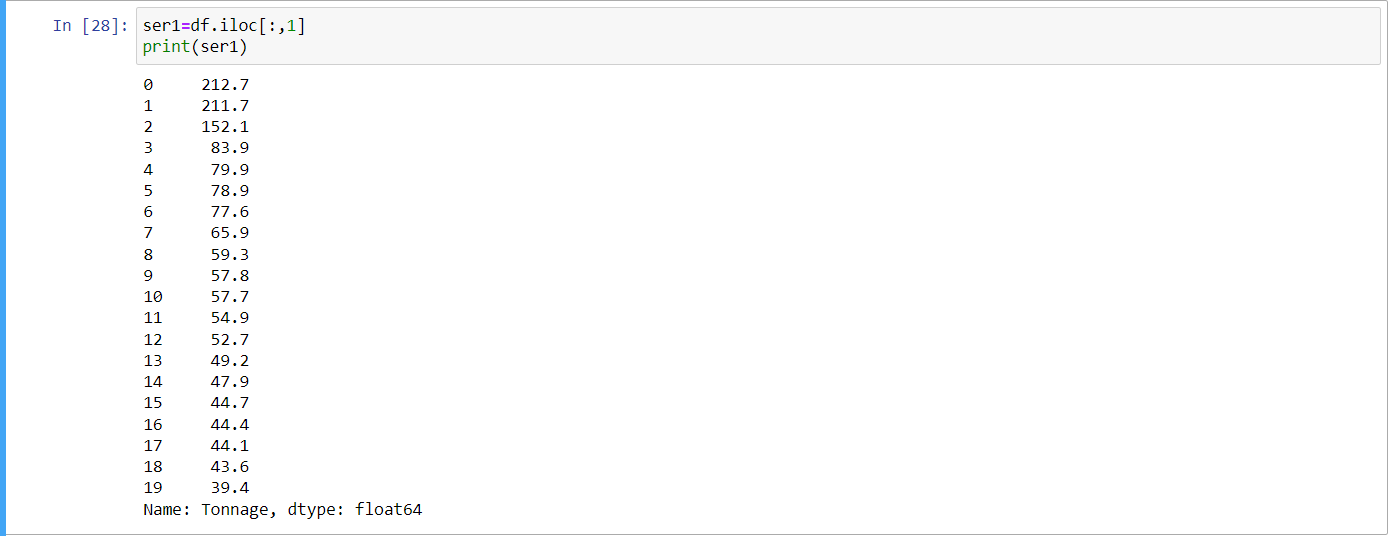
|  |
| --- |
| **Objective(s):**   * Find skewness and kurtosis for a given dataset |
| **Outcome:** Understand skewness and kurtosis and their implementation |
| **Problem Statement:** Demonstrate measures of shape using Python |
| **Background Study:**  Descriptive statistics are brief descriptive coefficients that summarize a given data set, which can be either a representation of the entire or a sample of a population. Descriptive statistics are broken down into measures of central tendency and measures of variability (spread). Measures of central tendency include the mean, median and mode, while measures of variability include standard deviation, variance, minimum and maximum variables, and kurtosis and skewness. |
| **Question Bank:**   1. What is the difference between skewness and kurtosis?   Skewness and kurtosis are both measures of the shape of a probability distribution.  Skewness measures the degree of asymmetry of a distribution. A distribution can be positively skewed, meaning that it has a long tail on the right-hand side and most of the observations are c clustered on the left-hand side of the distribution. Alternatively, a distribution can be negatively skewed, meaning that it has a long tail on the left-hand side and most of the observations are clustered on the right-hand side of the distribution. A perfectly symmetrical distribution has zero skewness.  Kurtosis, on the other hand, measures the degree of peakedness of a distribution. A distribution with high kurtosis has a sharp peak and fat tails, indicating that it has more extreme values than a normal distribution. A distribution with low kurtosis is flatter and more spread out than a normal distribution.   1. How to compute coefficient of skewness?   Ans: By multiplying the difference between the mean and median, multiplied by three.   1. What are mesokurtic, platykurtic and leptokurtic distributions?   Ans: Mesokurtosis: An excess kurtosis of 0. Normal distributions are mesokurtic.  Platykurtosis: A negative excess kurtosis. Platykurtic distributions are thin-tailed, meaning that they have few outliers.  Leptokurtosis: A positive excess kurtosis.   1. How would you interpret positive and negative skewness?   Positive skewness (right-skewed): The tail is on the right, and most values are concentrated on the left. The mean is greater than the median.  Negative skewness (left-skewed): The tail is on the left, and most values are concentrated on the right. The mean is less than the median.   1. What does it mean when a distribution has high or low kurtosis?   High kurtosis: The distribution has heavy tails and a sharp peak, meaning more extreme values (outliers) than a normal distribution.  Low kurtosis: The distribution has light tails and a flatter peak, indicating fewer extreme values and a more uniform spread.   1. What does it mean when a dataset has a skewness value of 0? Does this always indicate a perfectly symmetric distribution?   A skewness value of 0 means the dataset is perfectly symmetric around the mean. However, this does not always indicate a perfectly symmetric distribution, as it only suggests no skew; the distribution could still be flat or have outliers that affect its shape.   1. How does the presence of skewness in your data affect the results of statistical tests?   Skewness can distort statistical test results, especially those assuming normality (e.g., t-tests, ANOVA). Positively skewed data can inflate means, while negatively skewed data can lead to underestimations. Skewness can affect test power, leading to incorrect conclusions, so transformation (e.g., log) or non-parametric tests may be used to mitigate its impact.   1. How can you compute skewness and kurtosis in Python using libraries like scipy.stats and pandas?   Skewness measures the asymmetry of a distribution. Positive skew indicates a longer right tail, while negative skew indicates a longer left tail. A skewness of 0 suggests a symmetric distribution.  Kurtosis measures the tailedness of a distribution. High kurtosis indicates heavy tails (more outliers), while low kurtosis indicates light tails (fewer outliers). A kurtosis of 0 indicates a normal distribution’s tail characteristics.  In Python, you can compute these values using  scipy.stats.skew() and scipy.stats.kurtosis() for skewness and kurtosis, respectively.  Pandas Series.skew() and Series.kurt() methods can also compute these values for a dataset in a pandas Series.   1. How can the measurement of kurtosis help risk managers understand the probability of extreme events?   Kurtosis helps risk managers by indicating the likelihood of extreme events (outliers) in a dataset:  High kurtosis suggests a higher probability of extreme events or tail risks, as the distribution has heavy tails, meaning rare, extreme occurrences are more likely than in a normal distribution.  Low kurtosis indicates fewer extreme events, and the data is more concentrated around the mean.  Risk managers use this information to assess potential for unexpected risks (e.g., financial losses or market crashes) and plan accordingly.   1. Why might **skewness** be important when analyzing stock returns or other financial data?   Skewness is important in financial data analysis because:  Positive skew in stock returns suggests that while most returns are small, there are occasional large gains, which might signal growth opportunities.  Negative skew indicates more frequent small gains but occasional large losses, suggesting higher risk and potential for catastrophic events.  Understanding skewness helps investors manage risk exposure and adjust their strategies based on the likelihood of extreme returns, both positive and negative. |

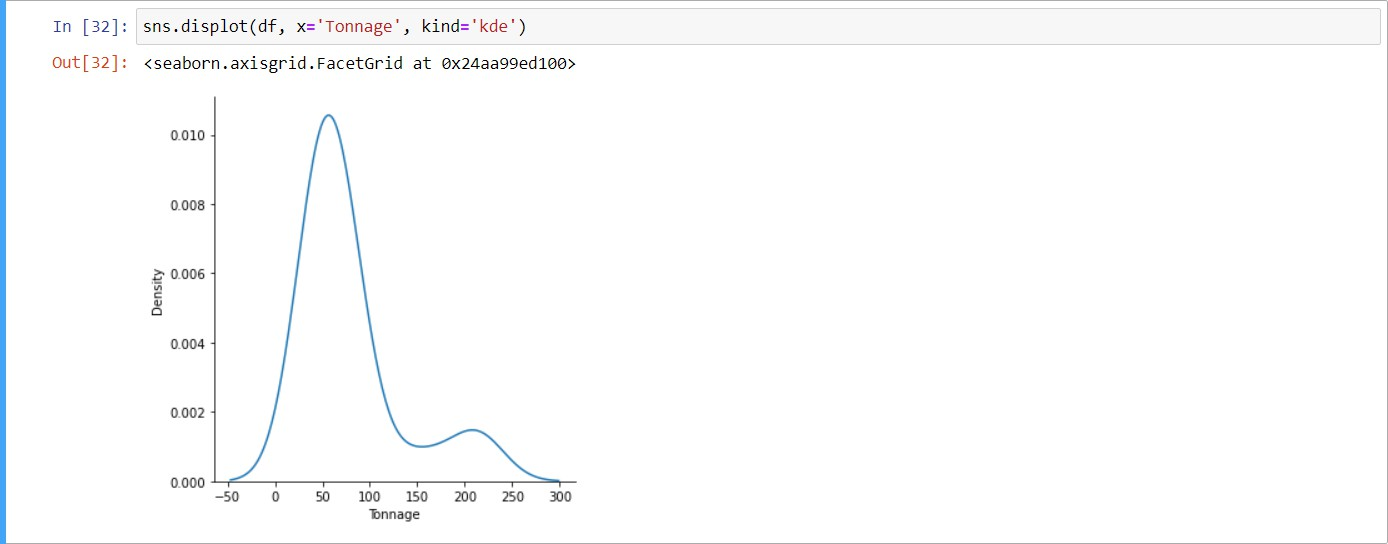
**Student Work Area**

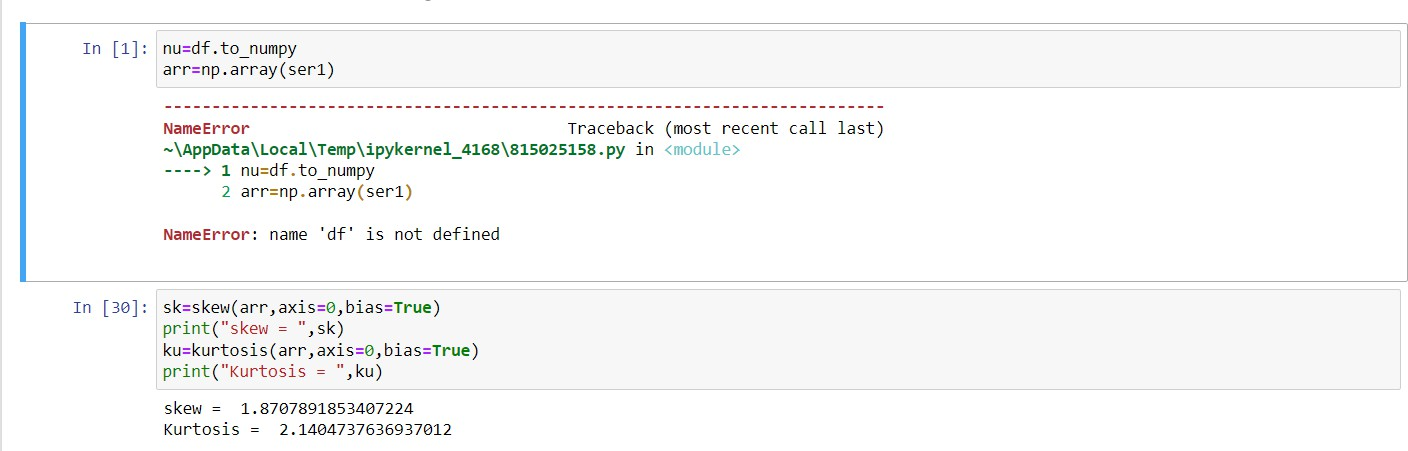
**Algorithm/Flowchart/Code/Sample Outputs**











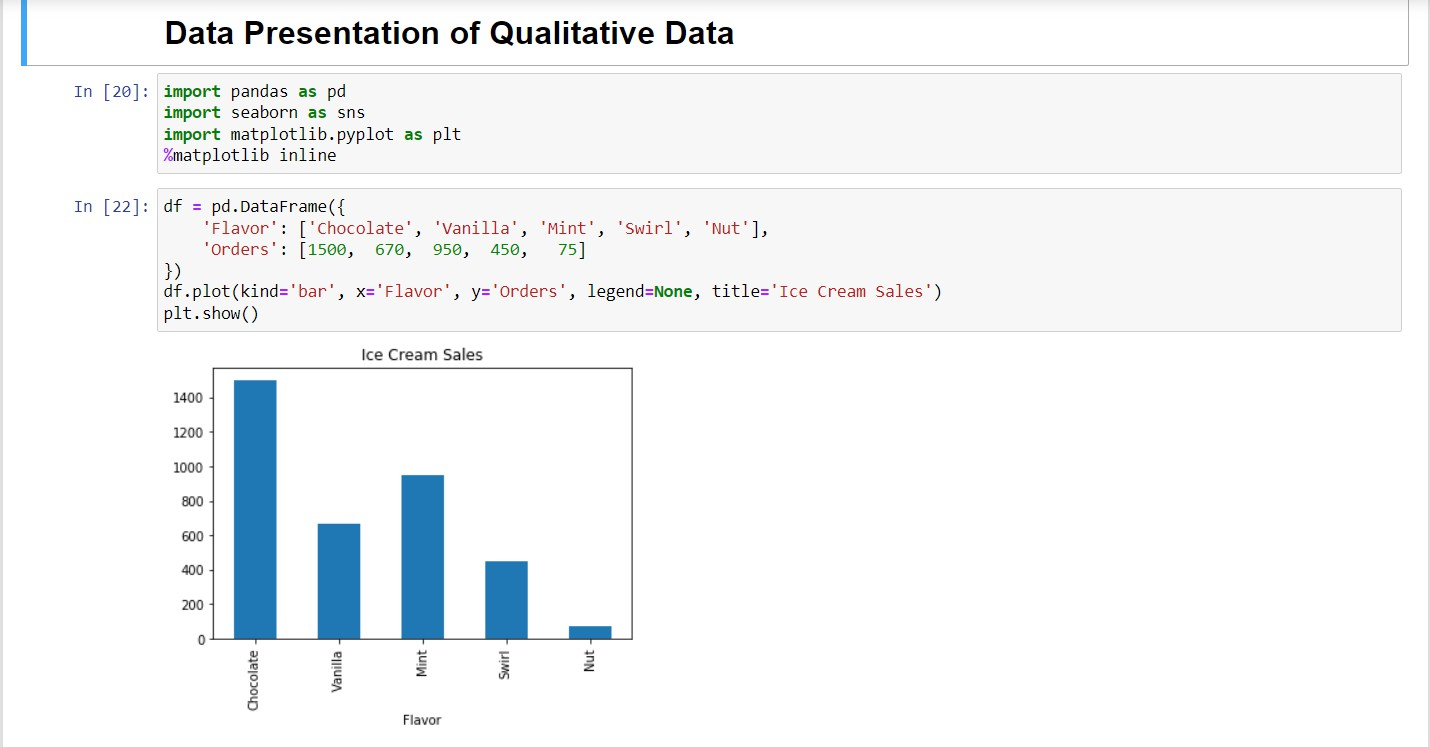
**EXPERIMENT NO. 4**

|  |
| --- |
| **Student Name and Roll Number:** Arjun Bhardwaj 21csu211 |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks/Grade:** |

|  |
| --- |
| **Objective(s):** Data visualization for qualitative and quantitative data |
| **Outcome:** Students will be familiarized with the use of python libraries such as matplotlib and seaborn |
| **Problem Statement:** Write a python program to use visualize qualitative and quantitative data |
| **Background Study:**  Pareto Chart, Pie chart, bar charts and boxplot are some examples of visualization for qualitative data  stem-and-leaf chart, count plot, histogram, frequency polygon and scatter plot are some examples of visualization for qualitative data |
| **Question Bank:**   1. What kind of data is generated using boxplot?   Ans: Box plots are used to show distributions of numeric data values.A box plot also displays the five-number summary of a set of data. The five-number summary is the minimum, first quartile, median, third quartile, and maximum   1. Can we plot boxplot in both matplotlib and seaborn library?   **Yes**   1. Can we identify outlier from stem-and-leaf plot?   Yes   1. What is kernel density estimation?   Yes   1. How do we decide upon the number of bins in histogram?   1. Count the number of data points.  2. Calculate the number of bins by taking the square root of the number of data points and round up   1. Which plot is used for analysis of bivariate data?   Scatter plot   1. How a stem-and-leaf plot is helpful in visualizing the distribution of data? How does it differ from a histogram?   A stem-and-leaf plot shows individual data points while visualizing the distribution, offering more detail than a histogram. A histogram groups data into bins and shows frequencies, making it better for large datasets.   1. When would you prefer using a **bar chart** over a **pie chart** to visualize categorical data? Explain with examples.   Use a **bar chart** when comparing **multiple categories** or showing **trends over time**, as it handles more data effectively. A **pie chart** is better for displaying **parts of a whole** with fewer categories.  Given a stem-and-leaf plot, how would you determine the median, range, and spread of the data? Provide an example.  To determine the median, range, and spread from a stem-and-leaf plot:  Median: Find the middle value by ordering the data and selecting the middle number.  Range: Subtract the smallest value (leaf with the smallest stem) from the largest value (leaf with the largest stem).  Spread: Look at the distribution of values to understand the variability or how spread out the data is.  Example:  Stem-and-leaf plot:  4 | 1 2 3  5 | 0 6 7  6 | 1 3 9  Ordered data: 41, 42, 43, 50, 56, 57, 61, 63, 6  Median: Middle value = 56  Range: 69 - 41 = 28  Spread: The data varies from 41 to 69, indicating a spread of 28.   1. How does a **count plot** work in Python using libraries like seaborn? What type of data would be appropriate for visualizing using a count plot?   **A count plot in Python, using Seaborn, visualizes the count of observations in categorical data. It creates a bar plot where the height of each bar represents the number of occurrences for each category.**  **Appropriate data:**   * **Categorical data such as gender, region, product types, etc.** * **Data with distinct categories where you want to show frequency distribution.**   **Example:**  **import seaborn as sns**  **import matplotlib.pyplot as plt**  **data = ['Male', 'Female', 'Female', 'Male', 'Male']**  **sns.countplot(x=data)**  **plt.show()** |

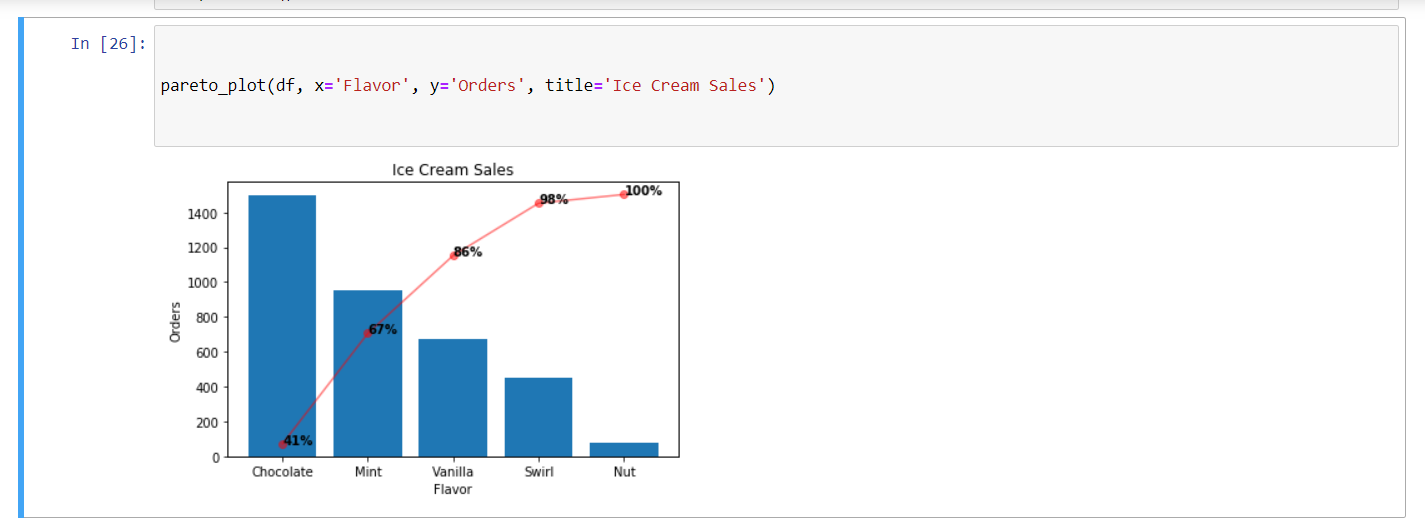
**Student Work Area**

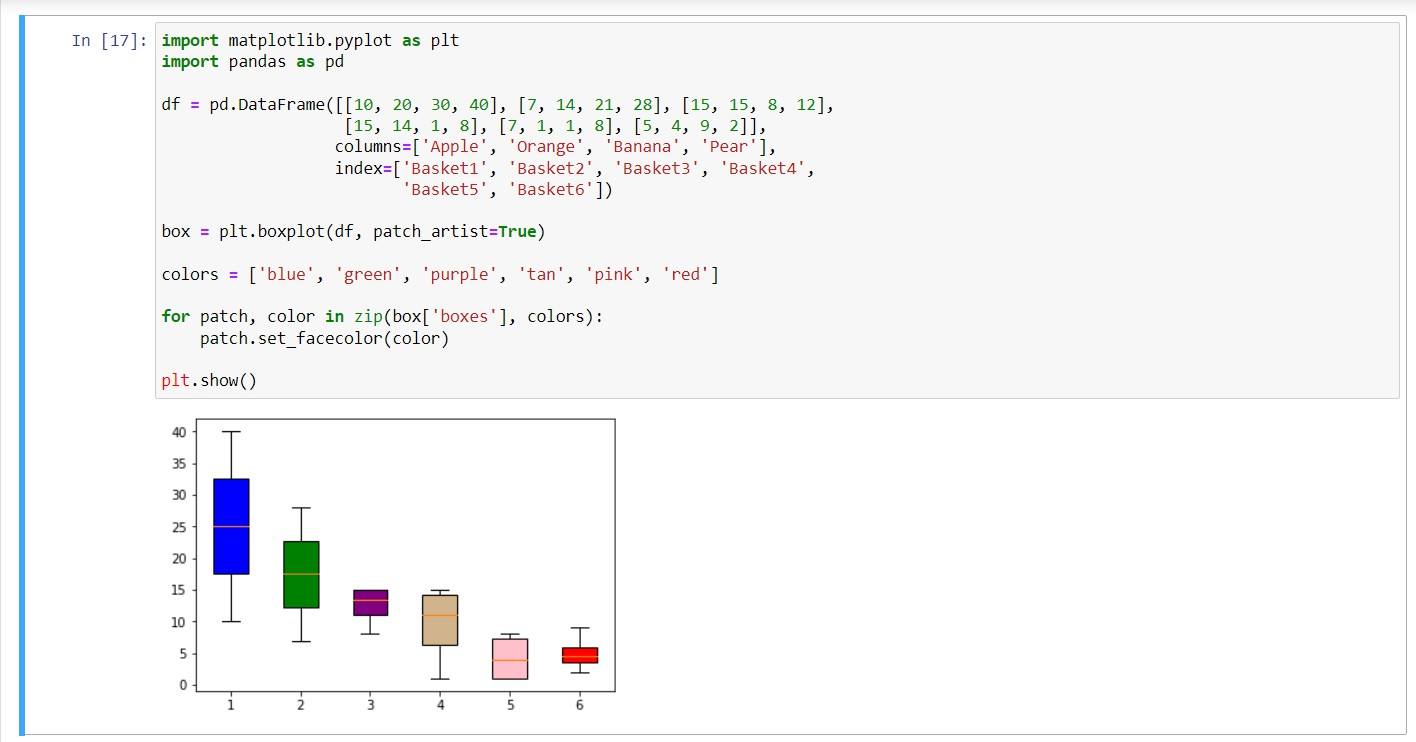
**Algorithm/Flowchart/Code/Sample Outputs**

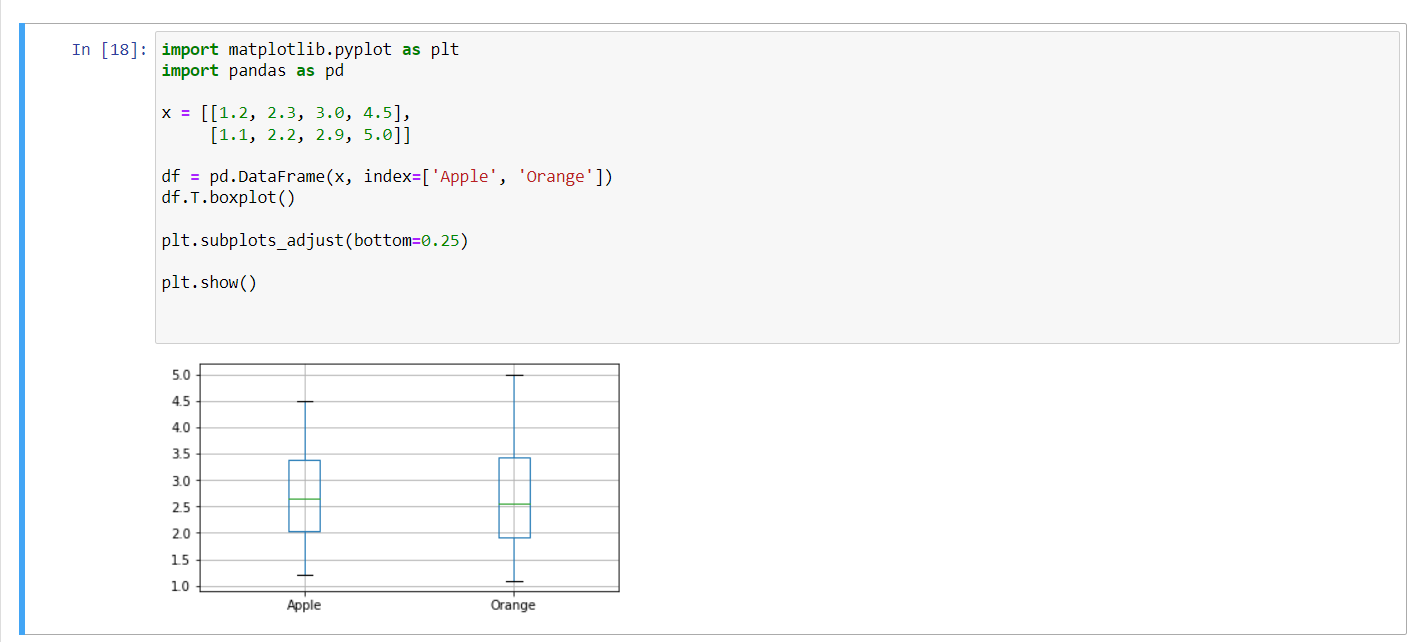
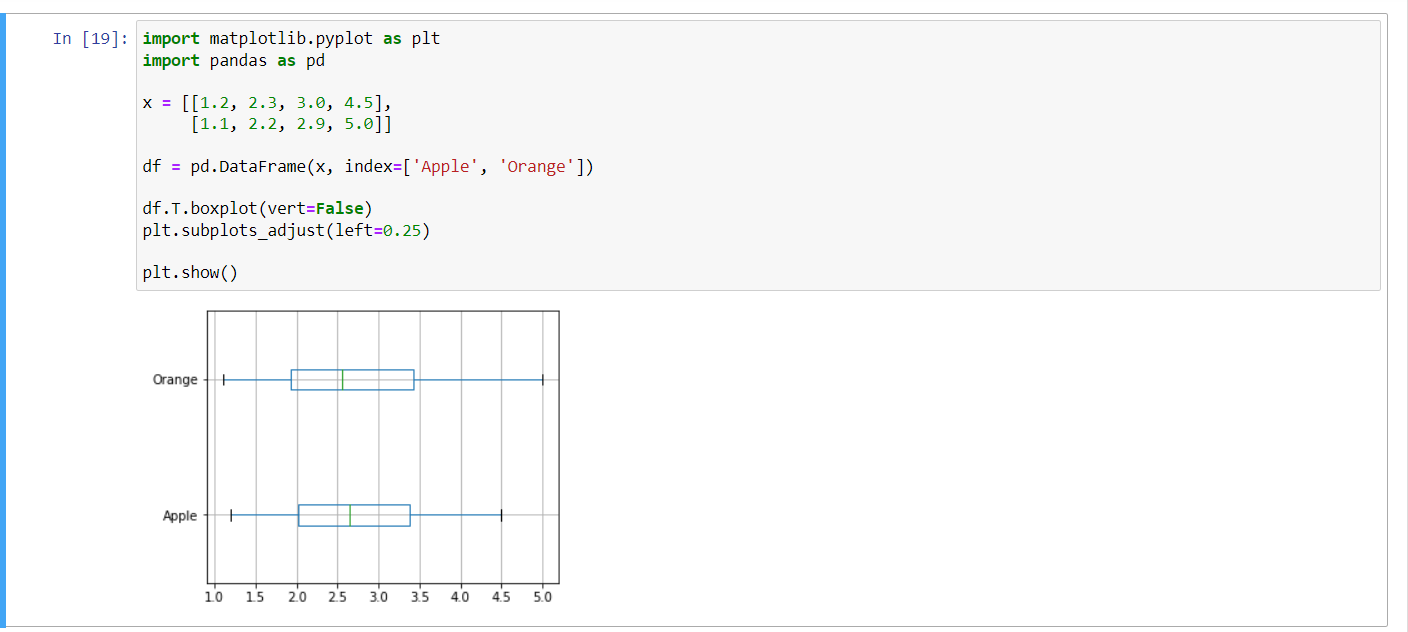










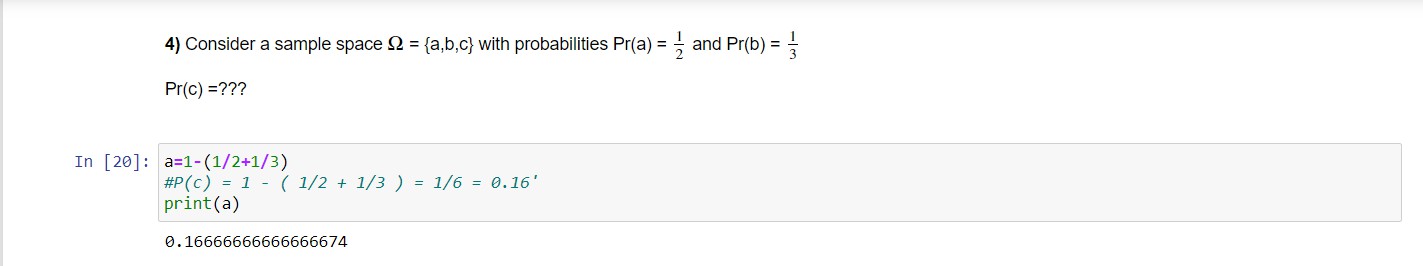
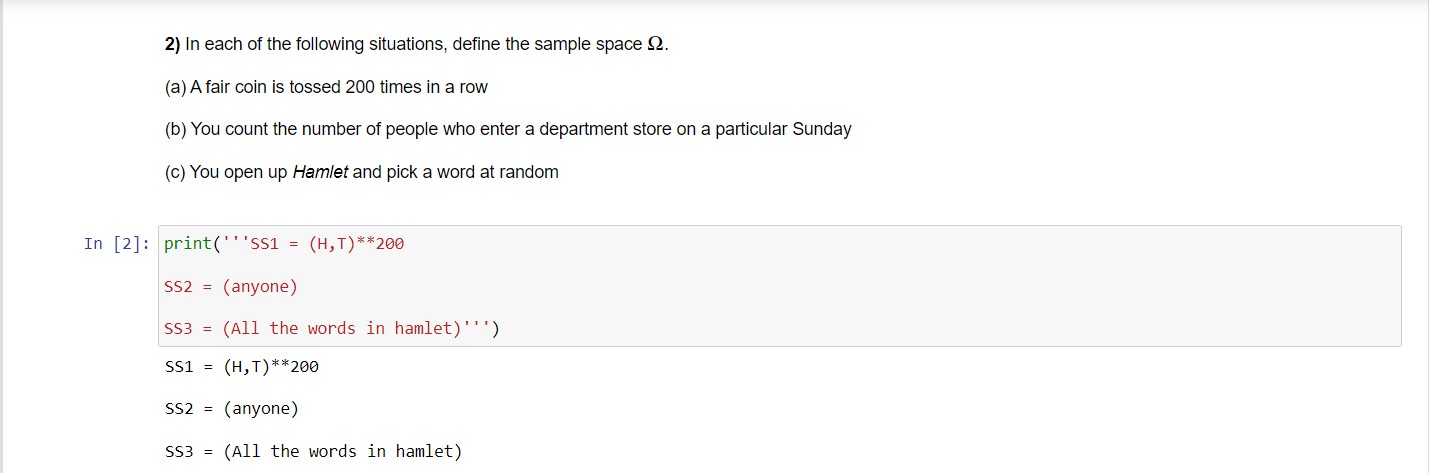
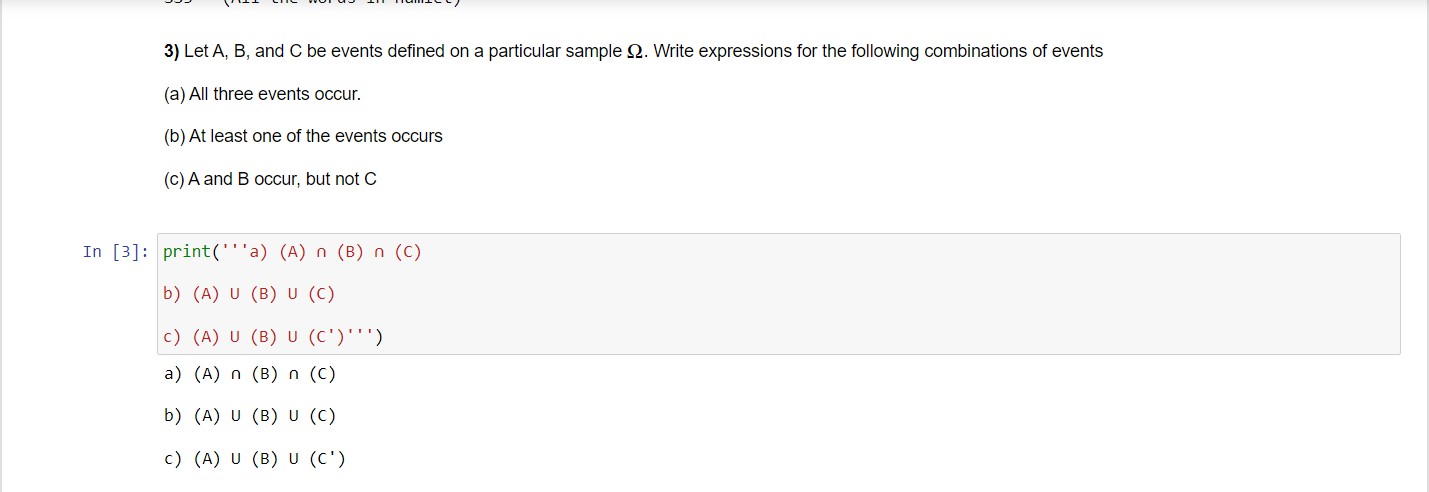
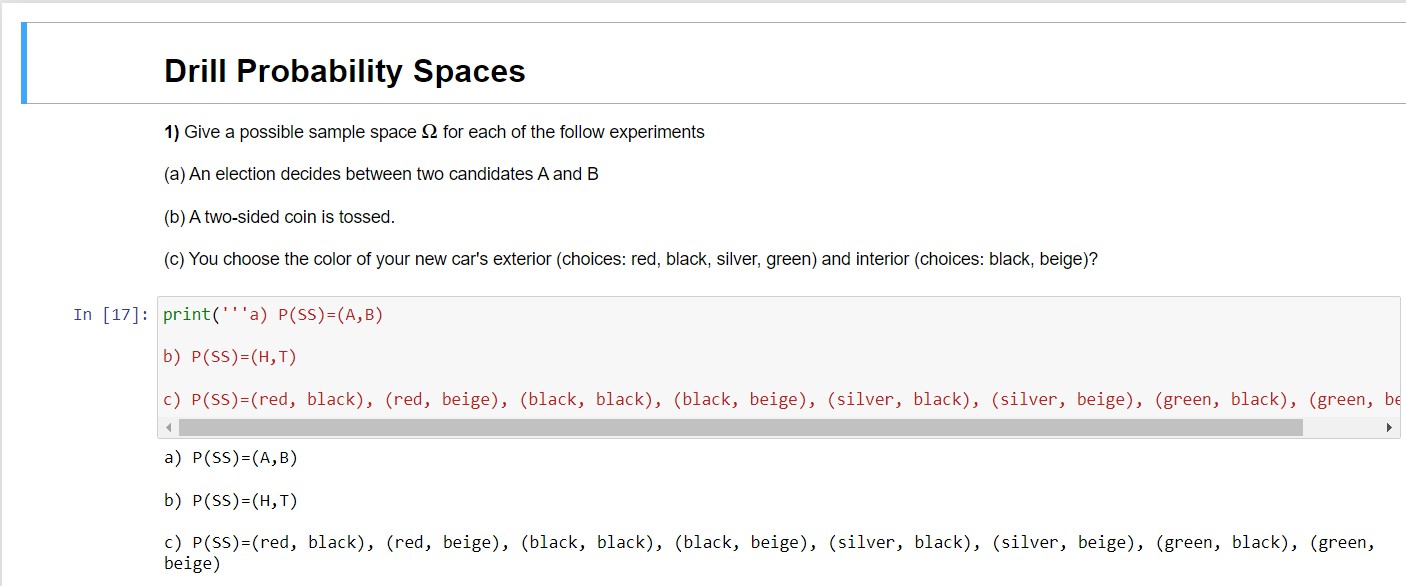
**EXPERIMENT NO. 5**

|  |
| --- |
| **Student Name and Roll Number:** Arjun Bhardwaj 21csu211 |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks/Grade:** |

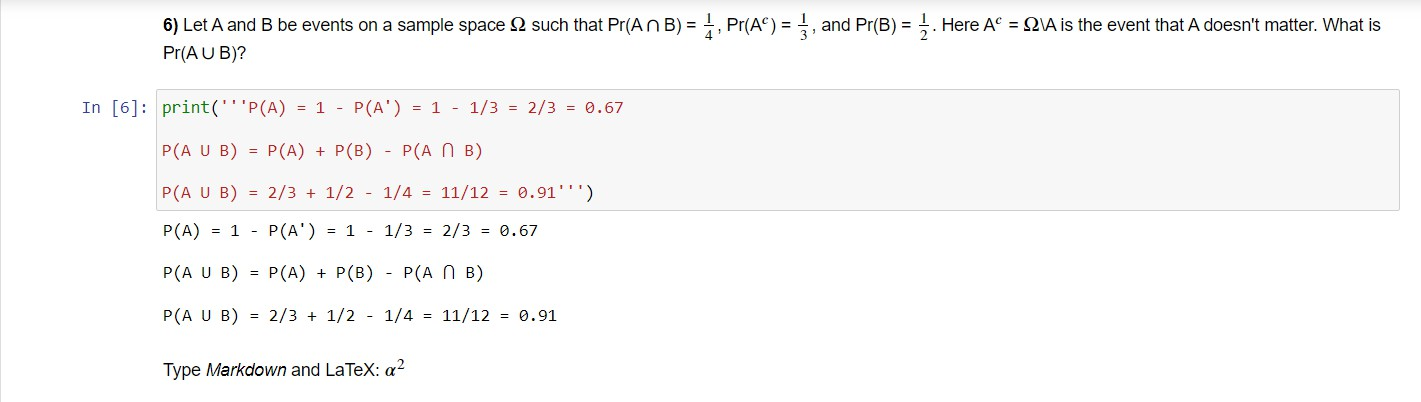
|  |
| --- |
| **Objective(s):** Write python program to implement Probability |
|  |
| **Problem Statement:** Basics of probability using Python |
| **Background Study:**  Various types of events, conditional probability, permutation and combination.  **Mutually Exclusive Events:** Events that cannot happen simultaneously.  Example: Drawing a heart and a spade from a single card draw.  **Independent Events:** Events where the outcome of one does not affect the other.  Example: Flipping a coin and rolling a die simultaneously.  **Dependent Events:** Events where the outcome of one affects the probability of the other.  Example: Drawing two cards from a deck without replacement.  **Complementary Events:** Two events where one event occurs if and only if the other does not.  Example: Getting heads vs. tails in a coin flip.  **Conditional Probability**  The probability of an event A occurring given that event B has already occurred is called conditional probability.  **Formula:**  P(A∣B) =P(A∩B)/ P(B) (if P(B)>0)  Example: Suppose the probability of rain is higher given that the sky is cloudy. The probability of rain (event A) given clouds (event B) is a conditional probability.  **Permutations:** The number of ways to arrange a set of items where the order matters.  **Formula:**  P(n,r)=n!/(n−r)!  where n is the total number of items, and r is the number of items to arrange.  Example: How many different ways can 3 books be arranged on a shelf from a set of 5 books?  **4. Combinations**  The number of ways to select items from a set where the order does not matter.  **Formula:**  C(n,r)=n!/r!(n−r)!  ​  Example: How many ways can you choose 3 players from a team of 10 for a relay race? |
| **Question Bank:**   1. What is the difference between mutually exclusive events and independent events? Provide examples of each.   Mutually exclusive events: Two events cannot occur at the same time.  Example: Rolling a die and getting a 2 or a 5. You can't get both at once.  Independent events: The occurrence of one event does not affect the probability of the other.  Example: Tossing a coin and rolling a die. The result of the coin toss does not affect the die roll.   1. If two events A and B are dependent, what is the formula for the conditional probability of A given B? How does this change if the events are independent?      1. What is the formula for calculating permutations when there are repeated elements?      1. What are complementary events, and how do you calculate the probability of the complement of an event?   Complementary events are two events that represent all possible outcomes of an experiment, such that one event occurs if and only if the other does not. In other words, if event A happens, then its complement A' does not happen, and vice versa.  The probability of the complement of an event A is calculated as:     1. Provide an example of a real-world scenario where you would use **permutations** and a different scenario where you would use **combinations**. How do the outcomes differ in these cases?   Permutations: Arranging books on a shelf, where order matters.  Example: 3 books, 3! = 6 ways to arrange.  Combinations: Choosing a team from 5 people, where order doesn't matter.  Example: 3 players from 5=10 ways to choose.   1. In how many ways can you select 3 representatives from a class of 15 students?      1. Can two events, such as getting an even number or an odd number on a die, be mutually exclusive? Explain why or why not.   Yes, the events of getting an even number or an odd number on a die are mutually exclusive because they cannot occur at the same time. A die roll can either result in an even number (2, 4, 6) or an odd number (1, 3, 5), but not both in a single roll.   1. What is the **intersection** of two events, and how would you describe it with an example from a deck of cards?   The intersection of two events refers to the outcomes that are common to both events, i.e., when both events occur simultaneously.  Example with a deck of cards  Event A: Drawing a red card (hearts or diamonds).  Event B: Drawing a face card (Jack, Queen, King).  The intersection of A and B would be the cards that are both red and face cards: the Jack of Hearts, Queen of Hearts, King of Hearts, Jack of Diamonds, Queen of Diamonds, and King of Diamonds.  So, the intersection would be these 6 cards.   1. What is the **union** of two events, and how would you represent it with an example from tossing a coin?   The union of two events refers to all outcomes that belong to either of the events or both.  Example with tossing a coin:  Event A: Getting Heads.  Event B: Getting Tails.  The union of A and B includes all possible outcomes: Heads or Tails. So, the union of these events is {Heads, Tails}, which represents all possible outcomes of a coin toss.   1. Can you provide an example of dependent events in the case of drawing cards from a deck without replacement?   Example of dependent events in drawing cards from a deck without replacement:  Event A: Drawing a king on the first draw.  Event B: Drawing a king on the second draw.  If you draw a king on the first draw, the probability of drawing a king on the second draw is affected because there is one less king and one less card in the deck. Hence, the events are dependent since the outcome of the first draw influences the second. |
|  |

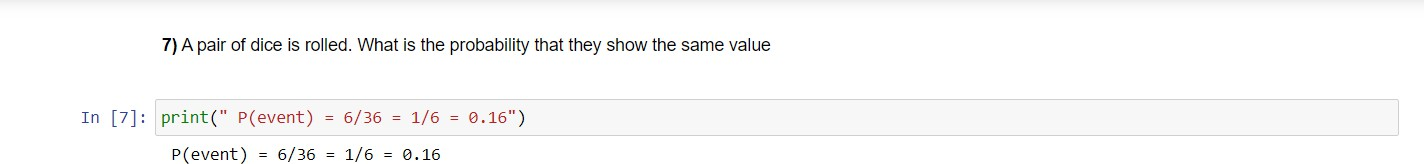
**Student Work Area**

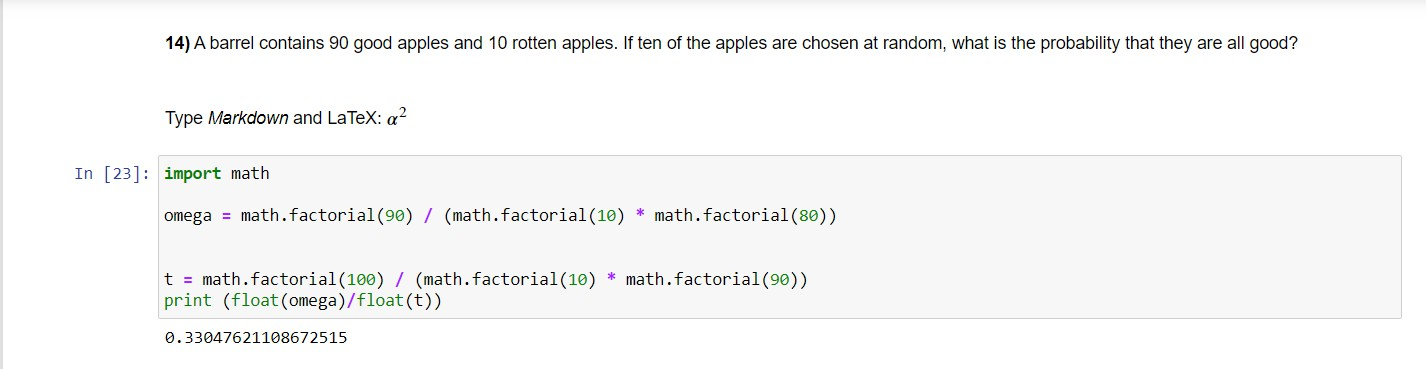
**Algorithm/Flowchart/Code/Sample Outputs**

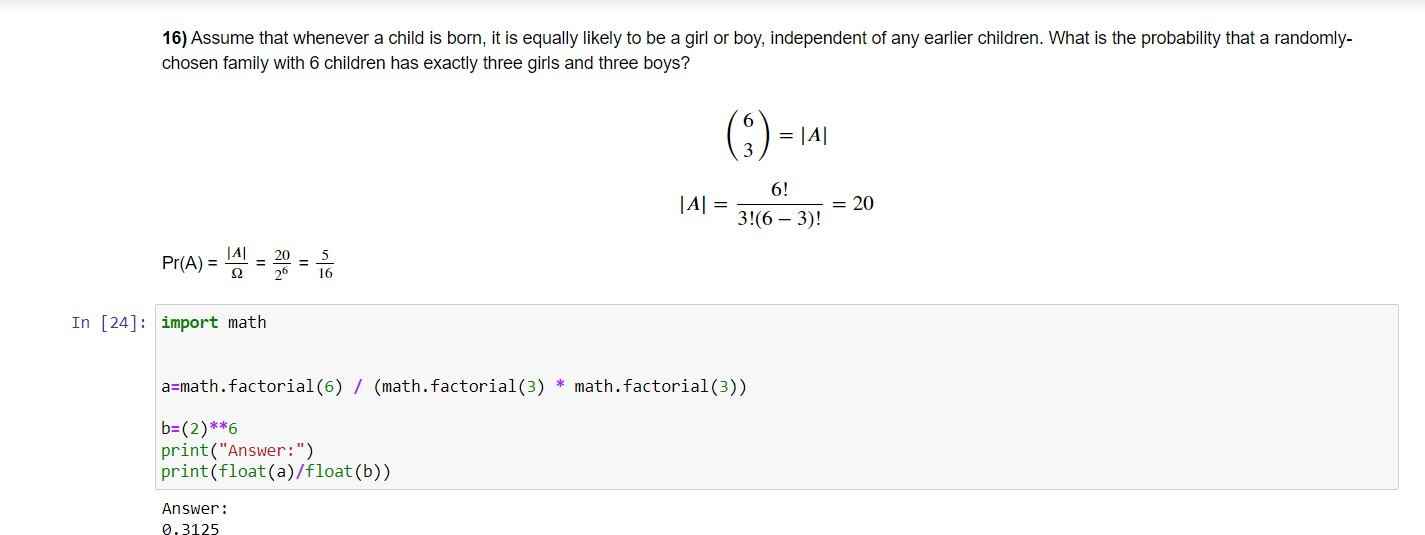












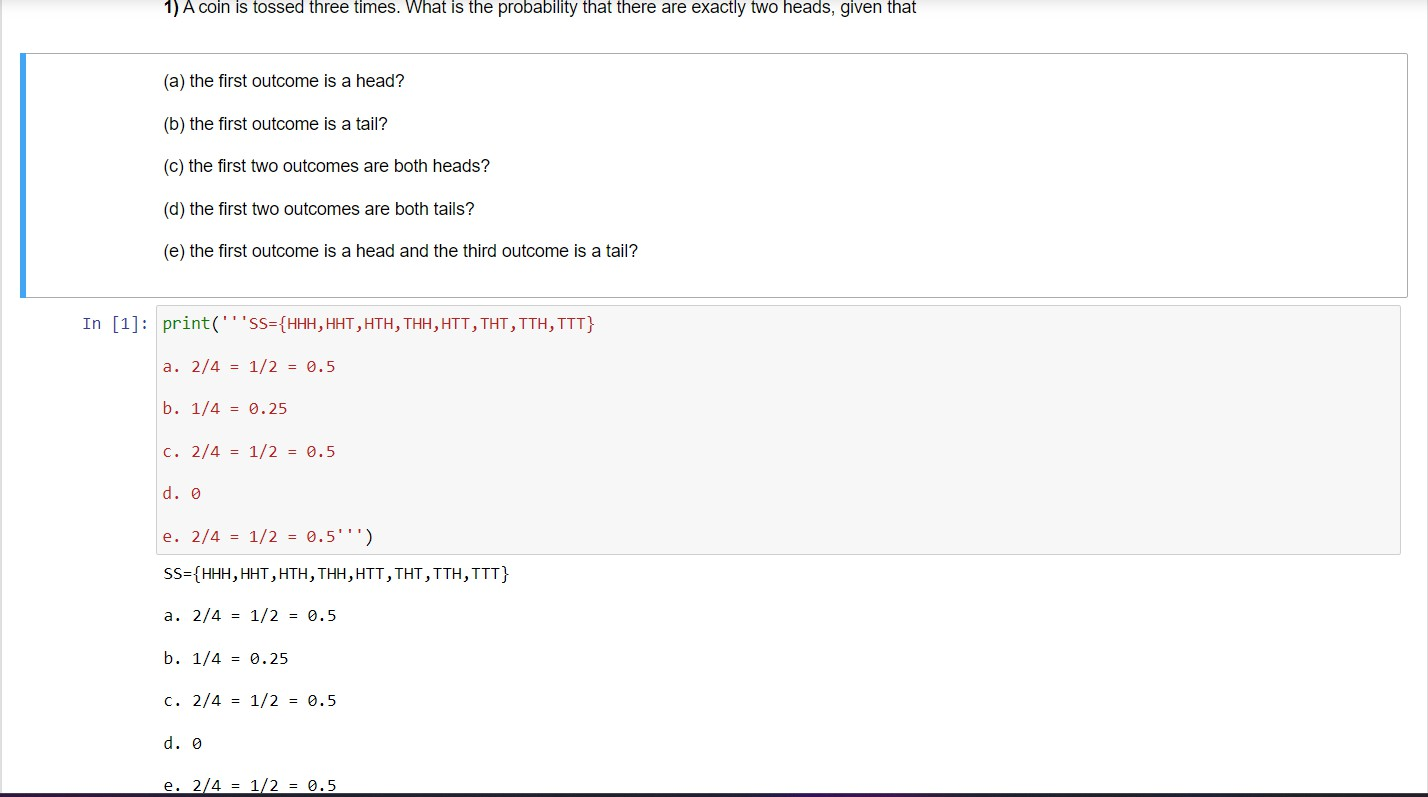
**EXPERIMENT NO. 6**

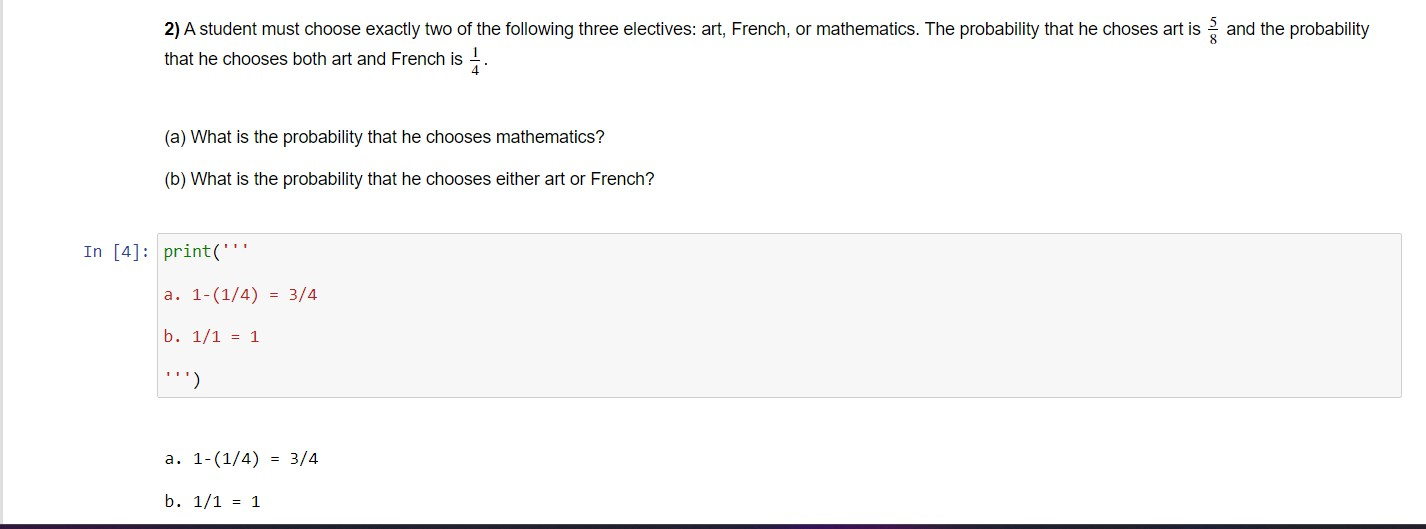
|  |
| --- |
| **Student Name and Roll Number:** Arjun Bhardwaj 21csu211 |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks/Grade:** |

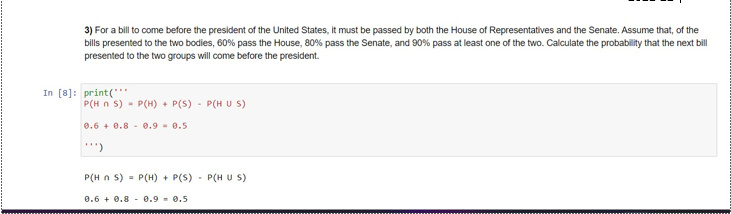
|  |
| --- |
| **Objective(s):** Write python program to implement Law of Probability |
|  |
| **Problem Statement:** ImplementingLaws of probability using Python |
| **Background Study:**  Various laws of probability like law of total probability, addition rule of probability, addition rule of probability etc.  The laws of probability are foundational principles that govern how probabilities are calculated and interpreted in various scenarios. They provide a framework for understanding how different events relate to each other, how likely outcomes are, and how to manipulate probabilities when considering different events. Here’s a brief overview of the key laws of probability:  **1. The Law of Total Probability**  This law allows us to find the probability of an event by considering all possible ways the event can occur, broken down by mutually exclusive events. It is particularly useful when an event is dependent on some other condition.    **Formula:**    where Bi​ are mutually exclusive events that partition the sample space.  **2. Addition Rule of Probability**  The addition rule calculates the probability of either one of two events happening. If the events are mutually exclusive, the probability of either event occurring is simply the sum of the individual probabilities. If the events are not mutually exclusive, the intersection (overlap) must be subtracted.    **Formula (for two events)**      **3. Multiplication Rule of Probability**  The multiplication rule helps calculate the probability of both of two events occurring, whether they are independent or dependent. If events are independent, the probability of their intersection is simply the product of their individual probabilities.  **Formula (for two events):**    **4. Complementary Rule**  The complementary rule is based on the fact that the probability of an event happening plus the probability of it not happening must equal 1.  **Formula: P(A′) =1−P(A)**  where A′ is the complement of event A, meaning the event that A does not occur.  **5. Bayes' Theorem**  Bayes' Theorem allows us to update our beliefs about the probability of an event based on new evidence. It is a powerful tool for conditional probability.  **Formula**: **P(A|B) =P(B|A) \* P(A)/P(B)**  where P(A∣B) is the probability of A given B, P(B∣A) is the probability of B given A, and P(A)and P(B)are the probabilities of A and B independently. |
| **Question Bank: (Done in assignment)**   1. If two events A and B are independent, what is the probability of A∩B? How can you test if two events are independent using Python, based on simulated data or real-world events? 2. Given that P(A)=0.4, P(B)=0.5, and P(B∣A) =0.3, use Bayes' Theorem to calculate P(A∣B). 3. How would you use Python to simulate the complement of a random event? 4. Suppose you are given two events A and B such that the probability of A given B is 0.3, and the probability of B is 0.4. Also, you know the probability of A given the complement of B is 0.5, and the probability of the complement of B is 0.6. What is the probability of event A using the Law of Total Probability? 5. In a population, 2% of people have a certain disease. A test for the disease is 95% accurate in detecting the disease (i.e., it correctly identifies 95% of sick people) and 90% accurate in detecting non-sick people (i.e., it correctly identifies 90% of healthy people). If a person tests positive, what is the probability that they actually have the disease? 6. In a deck of 52 cards, what is the probability of drawing either a heart or a queen? 7. A student is either in a morning or an afternoon class. The probability of passing the morning class is 0.8, and the probability of passing the afternoon class is 0.7. If the student is equally likely to be in either class, what is the total probability that the student will pass? 8. The probability of rain tomorrow is 0.4, and the probability of snow is 0.2. What is the probability of either rain or snow tomorrow (assuming rain and snow cannot occur simultaneously)? 9. A lottery ticket has a 1 in 1000 chance of winning. If a person buys two tickets, what is the probability of winning on at least one of the tickets? 10. A person has a 25% chance of developing a health issue due to poor lifestyle choices. What is the probability that the person will not develop the health issue? |

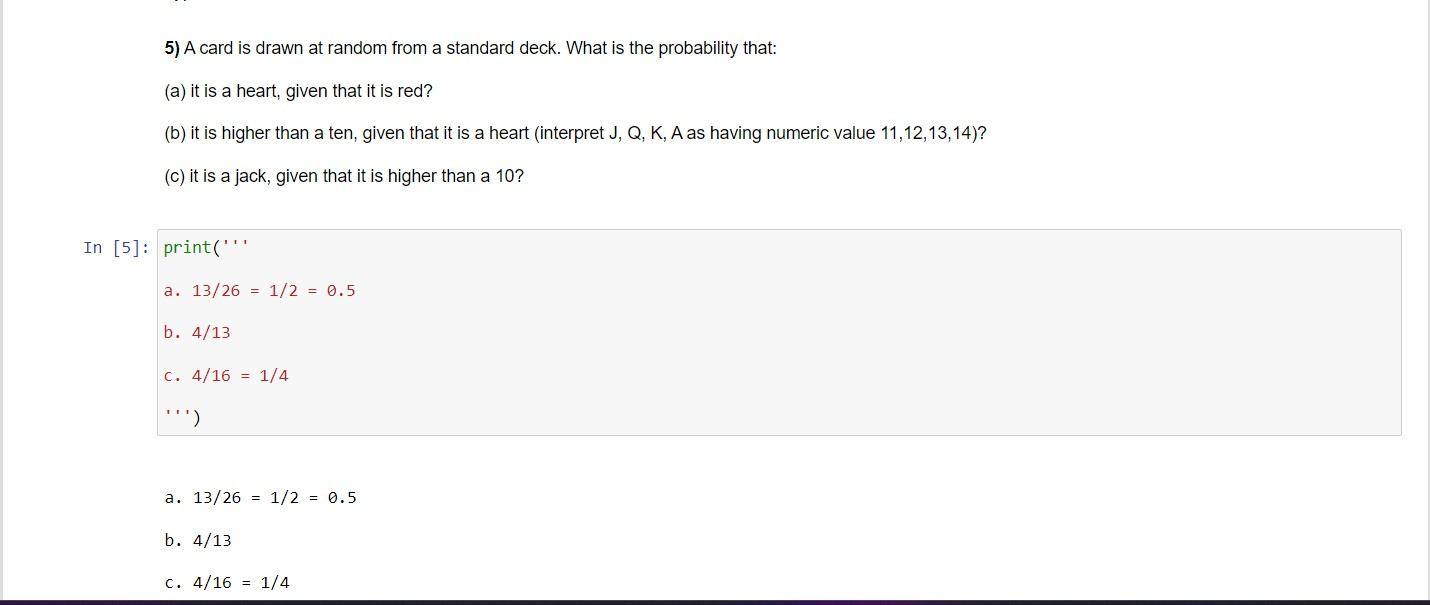
**Student Work Area**

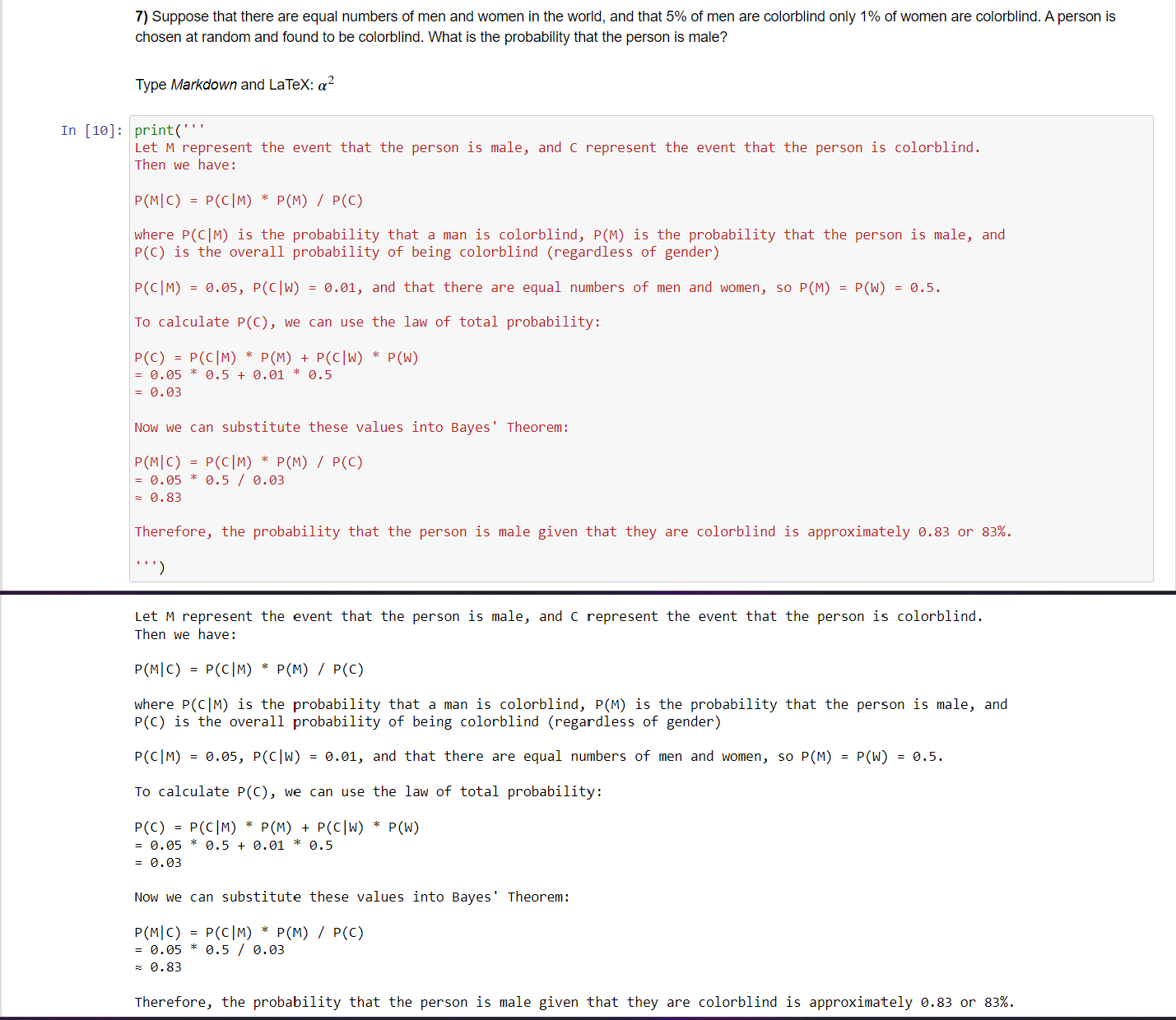
**Algorithm/Flowchart/Code/Sample Outputs**







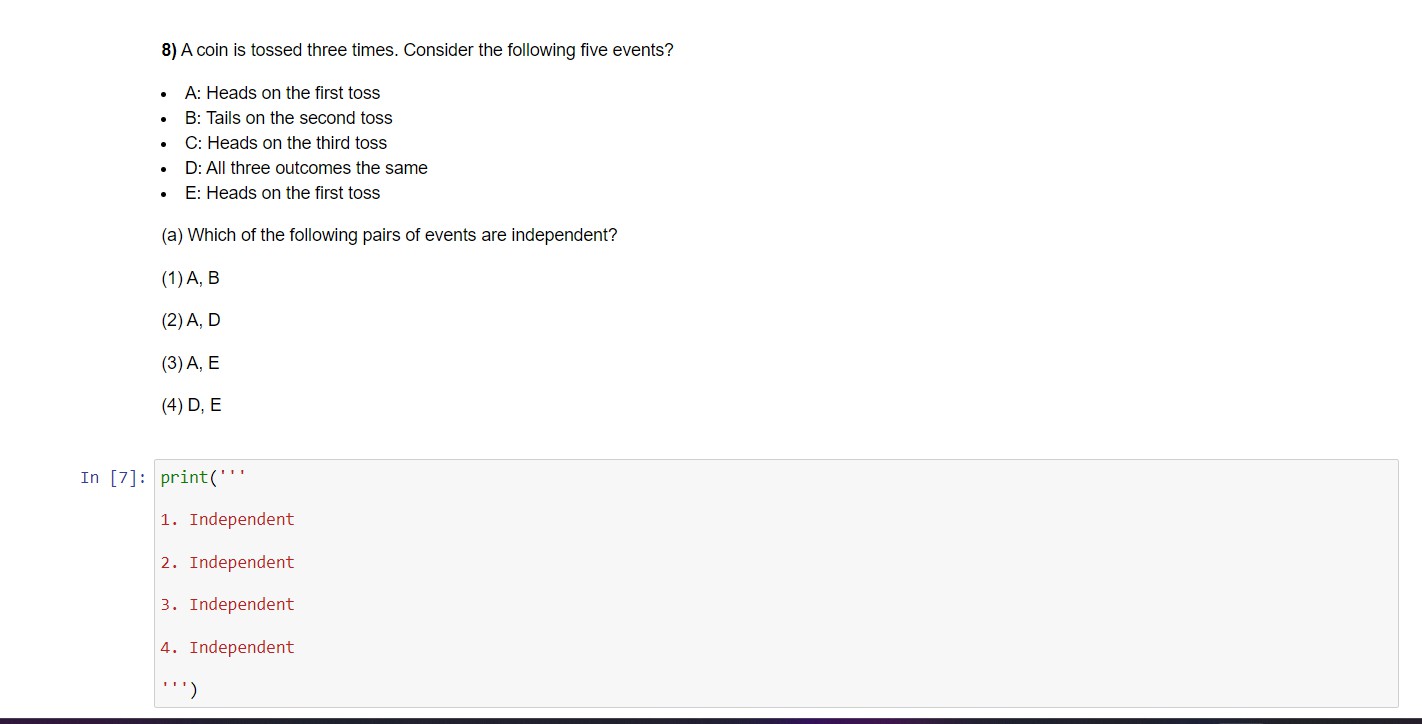






A screenshot of a computer

AI-generated content may be incorrect.



**EXPERIMENT NO. 7**

|  |
| --- |
| **Student Name and Roll Number:** Arjun Bhardwaj 21csu211 |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks/Grade:** |

|  |
| --- |
| **Objective(s):** Write python program to implement Probability Mass Function for Discrete Random Variable using Python |
|  |
| **Problem Statement:** Implementation of Probability Mass Function for Discrete Random Variable using Python |
| **Background Study:**  The **Probability Mass Function (PMF)** describes the probability distribution for discrete random variables. It assigns a probability to each possible value that the random variable can take. The PMF is defined as:  P(X=x) =p(x)  Where:   * X is a discrete random variable. * x represents possible outcomes of X. * p(x) is the probability of the outcome x.   The sum of all probabilities in a PMF equals 1:  ∑p(x)=1  **Example: Tossing a Fair Die**  Let X be the result of rolling a six-sided die. The PMF for this scenario would be:  P(X=x) =1/ 6 for x∈ {1,2,3,4,5,6} |
| **Question Bank:**   1. What types of random variables are described by a PMF? Give examples of real-life scenarios where PMFs are used.   Ans: A random variable is a rule that assigns a numerical value to each outcome in a sample space, or it can be defined as a variable whose value is unknown or a function that gives numerical values to each of an experiment‟s outcomes. There are two types of random variables, i.e. discrete and continuous random variables.  A discrete variable is a variable whose value can be obtained by counting since it contains a possible number of values that we can count.  A continuous variable is a variable whose value is obtained by measuring.  Examples of discrete random variable the number of outcomes in a rolling die, the number of outcomes in drawing a jack of spades from a deck of cards and so on.  Examples of continuous random variable height, weight, the amount of sugar in an orange, the time required to run a mile.   1. What is a Probability Mass Function (PMF)? How does it differ from a Probability Density Function (PDF)?   A Probability Mass Function (PMF) gives the probability of a discrete random variable taking a specific value.    A Probability Density Function (PDF) is for continuous random variables, where the probability at a specific point is 0, but the area under the curve gives the probability of an interval.  PMF: Discrete, sums to 1.  PDF: Continuous, area under the curve gives probability.   1. Why must the sum of all probabilities in a PMF equal 1?   The sum of all probabilities in a Probability Mass Function (PMF) must equal 1 because it represents the certainty that one of the possible outcomes will occur. In other words, for any random variable, the total probability across all possible values must sum to 100% (or 1), as one of these outcomes is guaranteed to happen.   1. What are the key properties of a **Probability Mass Function (PMF)**?      1. How can you compute and visualize the **PMF** of a discrete random variable using Python’s scipy.stats library?   import numpy as np  import matplotlib.pyplot as plt  from scipy.stats import binom  # Parameters for binomial distribution (n = 10 trials, p = 0.5 probability)  n = 10  p = 0.5  # Values for which to compute the PMF  x = np.arange(0, n+1)  # Compute PMF  pmf = binom.pmf(x, n, p)  # Plot PMF  plt.stem(x, pmf, basefmt=" ", use\_line\_collection=True)  plt.xlabel('x')  plt.ylabel('P(X = x)')  plt.title('Probability Mass Function (PMF) - Binomial Distribution')  plt.show()   1. How would you use Python’s matplotlib library to plot the **PMF** of a discrete random variable? What kind of plot should you use to represent a **PMF**?   To plot the PMF of a discrete random variable using matplotlib:  Stem Plot: Ideal for PMF, where each possible outcome has a vertical line representing its probability.  Bar Plot: Also suitable, where each bar's height shows the probability of a specific outcome.  Both plots help visualize the probability of each outcome for discrete variables, with stem plots being more commonly used for PMF.   1. How can you apply the **PMF** in real-life scenarios, such as modeling the number of customers arriving at a store within an hour, or the number of calls received by a call center?   The PMF can be applied in real-life scenarios where the outcomes are discrete and countable, such as:  Customers arriving at a store: You can use a Poisson distribution to model the number of customers arriving in a fixed time period, where the PMF gives the probability of exactly k customers arriving in an hour.  Calls received by a call center: Similarly, the Poisson distribution can model the number of calls per minute or hour. The PMF will show the likelihood of receiving k calls within a given time frame.  In both cases, the PMF provides probabilities for specific outcomes (e.g., 3 customers, 5 calls) and helps in planning resources, staffing, or optimizing operations based on the expected frequency of these events.   1. How would you interpret the **PMF** of a discrete random variable, and what information does it provide about the likelihood of different outcomes?   The PMF of a discrete random variable gives the probability of each possible outcome occurring. Here's how to interpret it:  Values on the x-axis: Represent the possible outcomes (e.g., number of customers, dice rolls, etc.).  Values on the y-axis: Show the probability of each outcome. The higher the value, the more likely that outcome is.  The PMF provides insight into:  Which outcomes are most likely.  The distribution of probabilities across all possible outcomes.  The shape of the distribution, helping to understand the behavior of the variable.  For example, in a dice roll, the PMF would show that each outcome (1 through 6) has a probability of 1/6 meaning each result is equally likely.   1. If you roll a fair 6-sided die, what is the **PMF** for the random variable representing the outcome of the roll?      1. Given that the probability of drawing a red ball from a bag containing 3 red balls and 2 blue balls is 0.6, what is the **PMF** of this random variable? |

**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

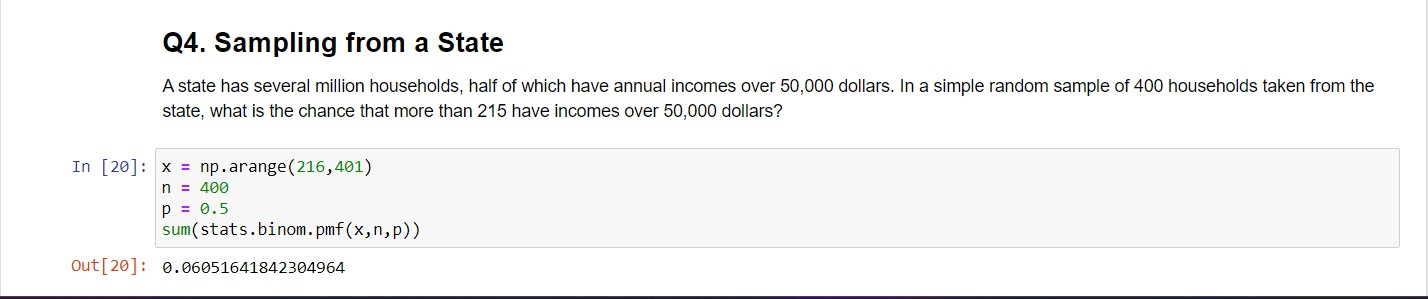


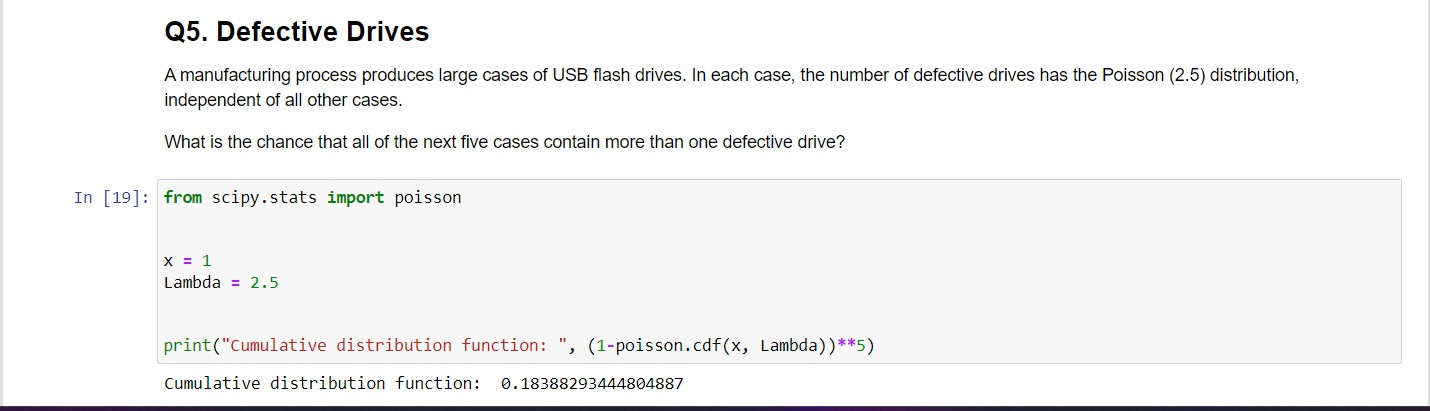


**Applied Computational Statistics**

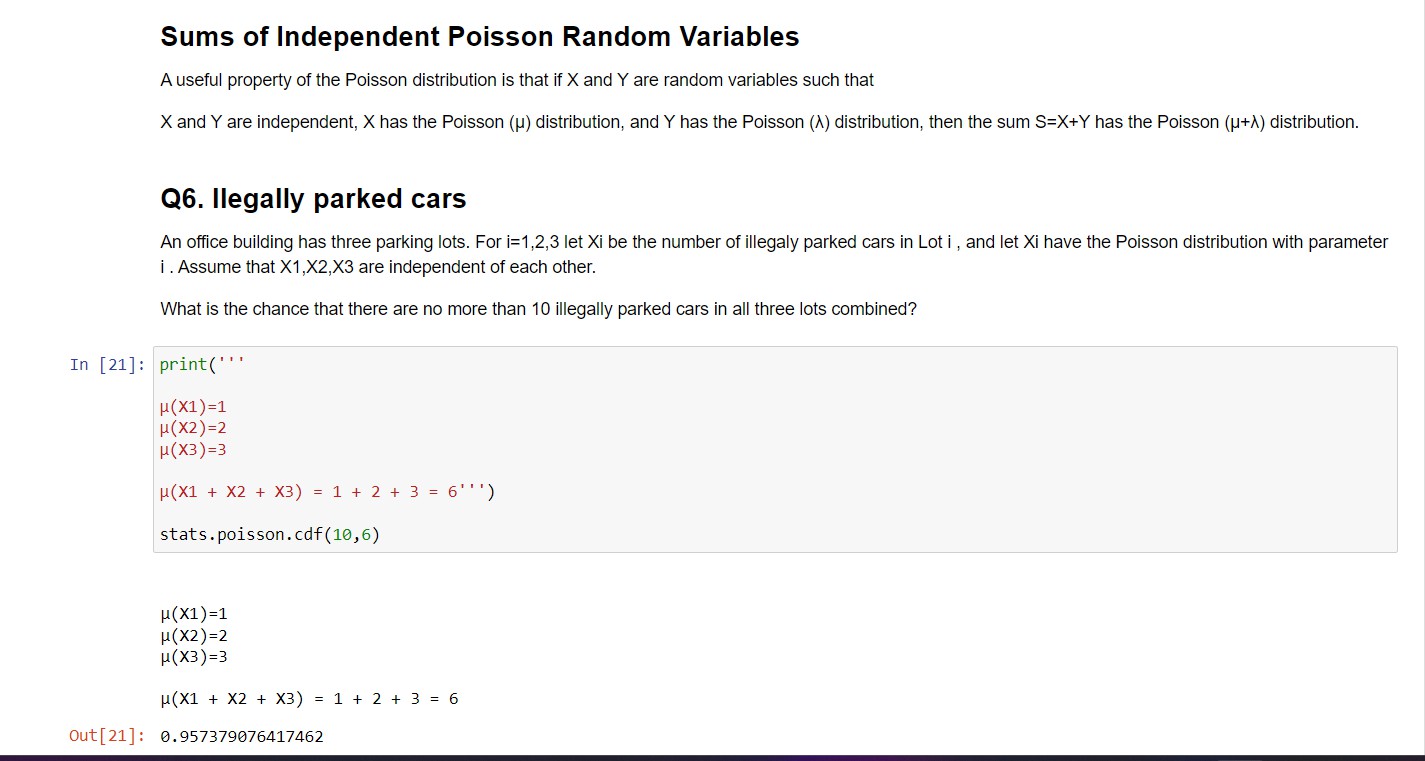
**(CSL227) Lab Manual**

**2021-22**

****

****

****



**Applied Computational Statistics**

**(CSL227) Lab Manual**

**2021-22**

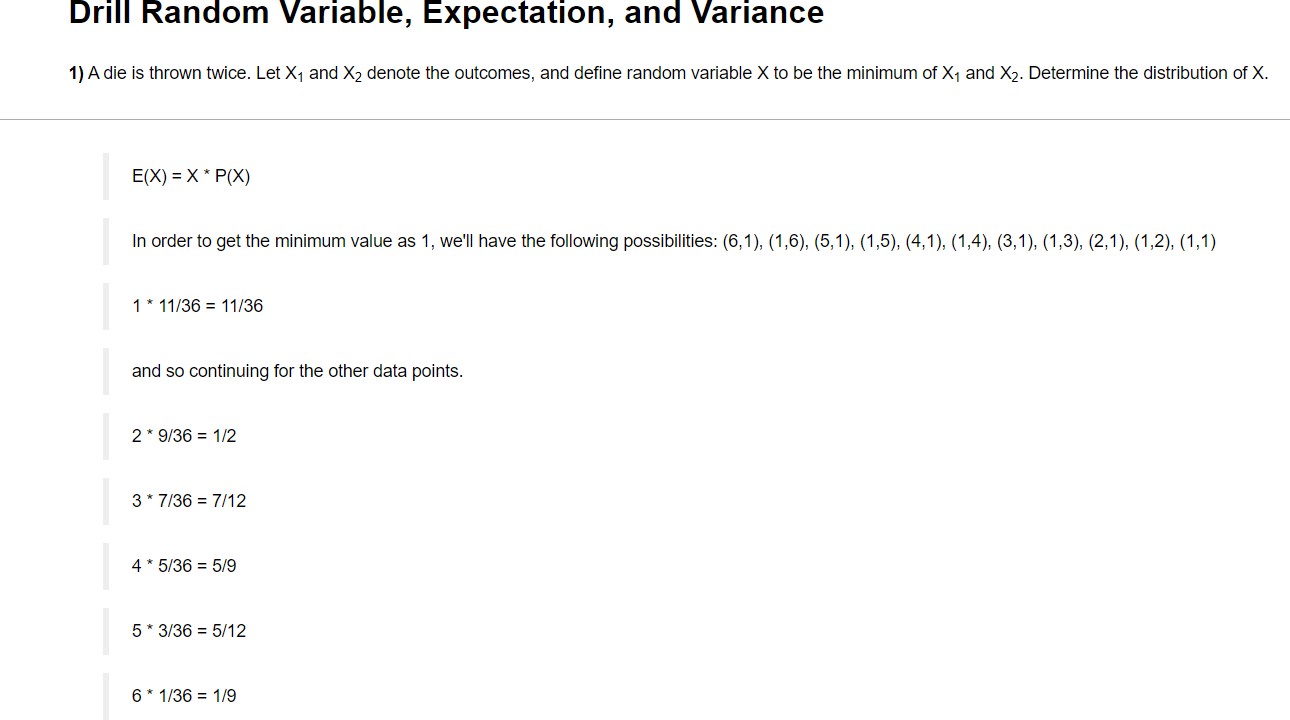
**EXPERIMENT NO. 8**

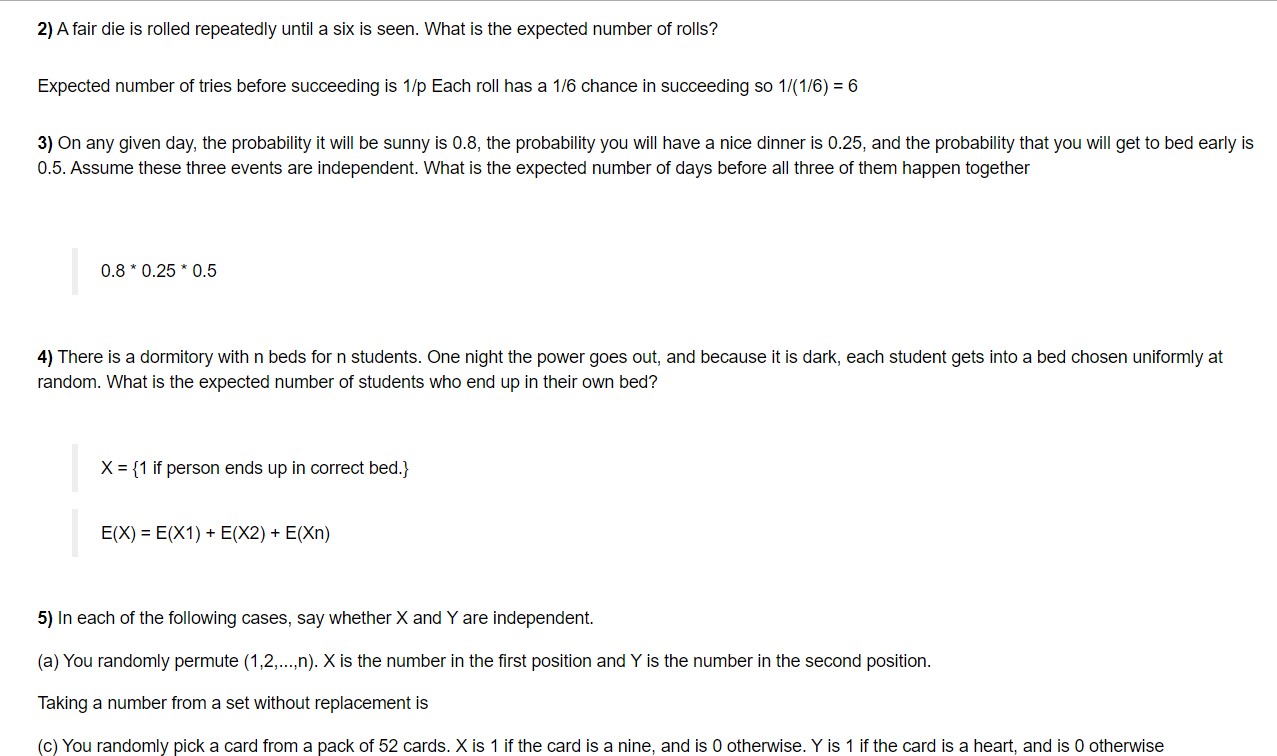
|  |
| --- |
| **Student Name and Roll Number:** |
| **Semester /Section:** |
| **Link to Code:** |
| **Date:** |
| **Faculty Signature:** |
| **Marks/Grade:** |

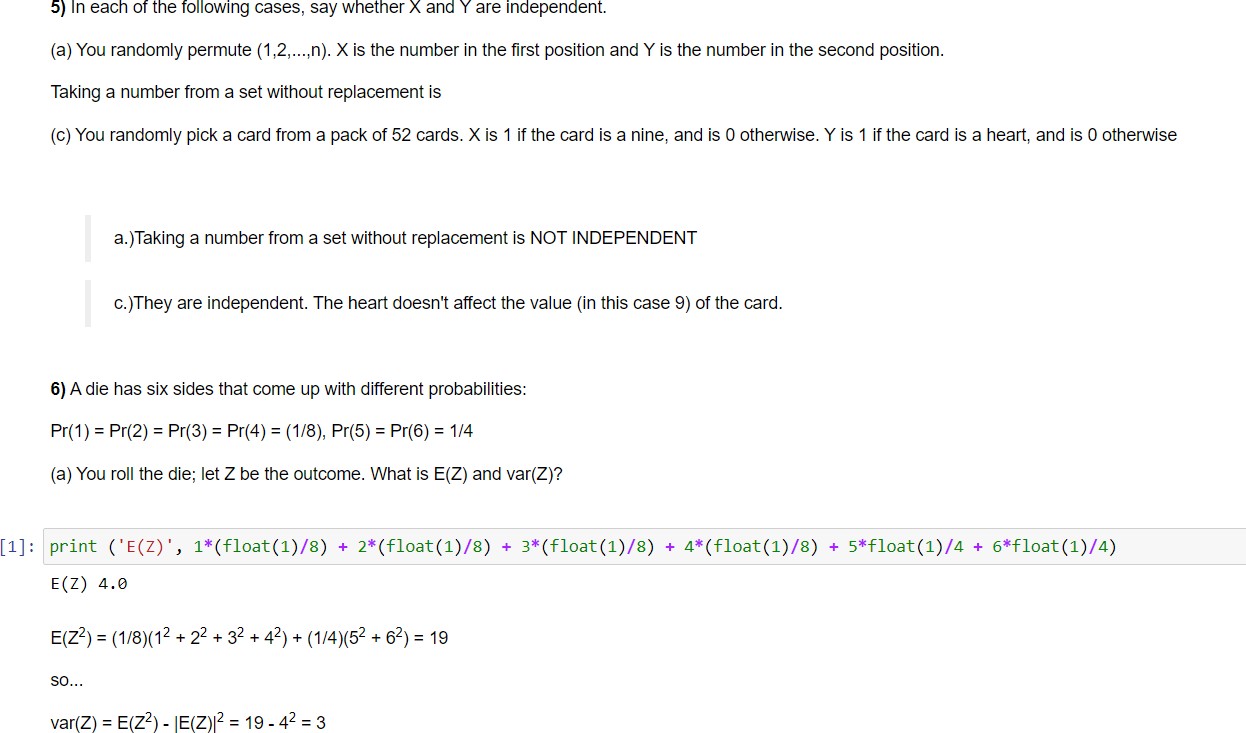
|  |
| --- |
| **Objective(s):** Write a python program to find expectation and variance in discrete distribution |
|  |
| **Problem Statement:** Implementation of expectation and variance in discrete distribution using Python. |
| **Background Study:**  In probability theory, expectation (mean) and variance are fundamental concepts that describe the behavior of a discrete random variable:   1. **Expectation (Mean):** The expectation E[X] of a discrete random variable XXX represents the long-run average value of X over many trials.   **Formula:** E[X]=∑x⋅P(X=x)   1. **Variance:** The variance Var(X) measures the spread of the random variable's values around the expectation.   **Formula:** Var(X)=E[(X−E[X])2] =∑x(x−E[X])2⋅P(X=x)   1. **Standard Deviation:** The square root of the variance, providing a measure of dispersion in the same units as the random variable. |
| **Question Bank:**   1. How do you compute the expected value and variance of a discrete random variable using its PMF?      1. What is the expectation (mean) of a discrete random variable? How does it relate to the long-term average in repeated trials?   The mean of a discrete distribution can be computed by summing the product of each possible value and its corresponding probability. Mathematically, this can be expressed as:  μ = Σ[x \* P(x)]  where μ is the mean, x is a possible value, and P(x) is the probability of x.   1. Explain why variance is used as a measure of dispersion in a distribution. How does it differ from the expectation?   Variance measures how much the values of a random variable deviate from the mean (expectation). While expectation shows the central value, variance captures the spread or dispersion around that value.  Expectation: The average outcome.  Variance: The average squared deviation from the expectation — it shows how spread out the data is.  A high variance means more variability; a low variance means values are tightly clustered around the mean.   1. How is variance computed using the expectation? Why is subtracting the mean necessary when calculating variance?   Variance is computed using expectation as:    Subtracting the mean centers the data, and squaring emphasizes the size of deviations. This captures how far values typically are from the mean, making variance a true measure of dispersion.   1. Given a random variable X with outcomes {1, 2, 3, 4} and probabilities {0.1, 0.2, 0.3, 0.4}, calculate the **expected value** of X both manually and using Python.   Given:  Outcomes: {1, 2, 3, 4}  Probabilities: {0.1, 0.2, 0.3, 0.4}     1. How are **expectation** and **variance** related in a discrete distribution?   In a discrete distribution:  Expectation (E[X]) is the average outcome.  Variance (Var(X)) measures how much the outcomes deviate from E[X].  They’re related through the formula:     1. How can you apply **expectation** and **variance** in Python to analyze real-world data, such as the distribution of customer arrivals at a store or the number of emails received in a day?   In Python, you can calculate expectation as the weighted average of outcomes and variance as the spread using libraries like NumPy or pandas. For example, analyzing customer arrivals per hour or daily email counts helps estimate average demand (expectation) and variability (variance) to plan resources efficiently.   1. What is the relationship between **variance** and **standard deviation**? How would you calculate the standard deviation in Python after computing the variance?   Standard deviation is the square root of variance. It provides a measure of spread in the same units as the data, making it easier to interpret.  In Python, after computing variance, you calculate standard deviation using the square root function (e.g., np.sqrt(variance) if using NumPy).   1. What is the relationship between **variance** and **standard deviation**? How would you calculate the standard deviation in Python after computing the variance?   Standard deviation is the square root of variance, giving a measure of spread in original data units. In Python, compute it using np.sqrt(variance) after finding variance.   1. How would you use the **scipy.stats** module in Python to calculate the **expected value** and **variance** for a given discrete distribution.   Using scipy.stats, you can define a discrete distribution (like rv\_discrete) with given outcomes and probabilities. Then, use .mean() for expected value and .var() for variance. This simplifies analyzing custom discrete distributions in real-world scenarios. |

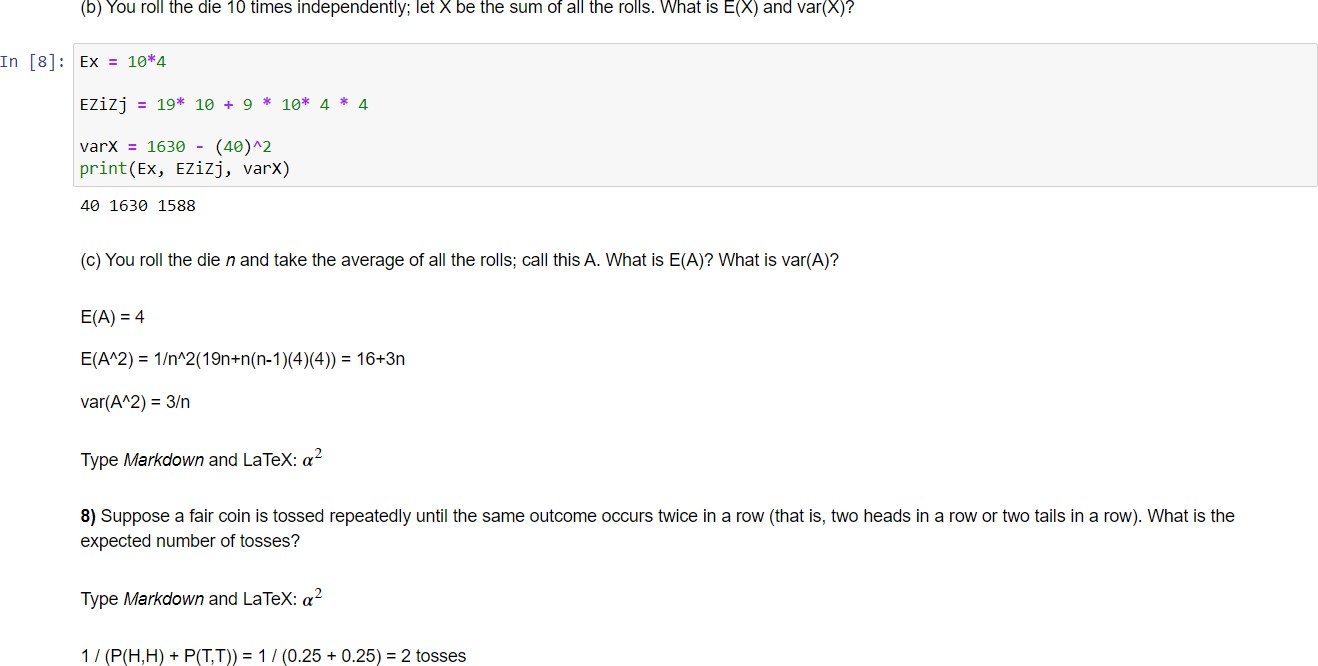
**Student Work Area**

**Algorithm/Flowchart/Code/Sample Outputs**

****

****

****

****