Project Report

Project title	**Analyzing Factors Influencing Body Performance Through Statistical Analysis**
Module Name	**WSQ Statistics for Data Science (SF)**
Qualification Name	**Diploma in Infocomm Technology (Data)** **(Synchronous and Asynchronous E-Learning)**

Student name	**Assessor name**	
Gangeswaran Rajavalarmathi	RYAN SURYANTO	
Date issued	**Completion date**	**Submitted on**
08-08-2023	09-09-2023	09-09-2023

Project Title	**Analyzing Factors Influencing Body Performance Through Statistical Analysis**

Learner declaration
I certify that the work submitted for this assignment is my own and research sources are fully acknowledged. Student signature: Date:

Content

Sr. No.	Description	Page No.
1	Project Overview	
2	Project Technical Environment	
3	Analytical Technique & Tools used	
4	Data Acquisition	
5	Descriptive Statistics Analysis	
6	Probability Analysis	
7	Hypothesis Testing	
8	Confidence Intervals	
9	Data Visualization	
10	Interpretation of Findings	
11	Conclusion	
12	Appendices • **Python Code** • **Additional Visuals / screenshots**	

1. Project Overview

Project Overview: Data Analysis and Statistical Inferences

Introduction:

- This project focuses on utilizing statistical techniques and Python programming to analyze a dataset containing various attributes related to body performance and health.
- The project aims to provide valuable insights to a research team, aiding their understanding of body performance trends and factors.

Dataset:

- The dataset comprises several attributes, including age, gender, height, weight, body fat percentage, blood pressure measurements (diastolic and systolic), grip force, sit and bend forward measurements, sit-ups counts, broad jump distance, and a class label.

Project Tasks:

1. Calculate Descriptive Statistics Measures for Data Analysis:

- Calculate essential summary statistics such as mean, median, and standard deviation for critical attributes like age, weight, and height.
- Utilize visualizations like histograms and box plots to gain insights into the distribution of key attributes.
- Interpret the descriptive statistics to comprehend the inherent characteristics of the dataset.

2. Apply Probability Concepts to Analyze Uncertain Events and Outcomes:

- Utilize probability concepts to analyze the gender distribution within the dataset.
- Estimate the probability of specific blood pressure levels using the assumptions of normal distribution.
- Apply the binomial distribution to analyze discrete outcomes, such as determining the probability of being male or female.

3. Explore Random Variables and Their Probability Distributions:

- Investigate the distribution of diastolic blood pressure and calculate the probabilities associated with specific ranges.

- Examine the probability distribution of grip force measurements and evaluate their variability.
- Employ probability distributions to identify unusual values within the dataset.

4. Perform Hypothesis Testing to Make Statistical Inferences:

- Formulate hypotheses and conduct hypothesis testing to determine whether there are significant differences in body fat percentages between genders.
- Calculate t-statistics and p-values to make informed decisions based on the results of hypothesis tests.
- Evaluate the practical significance of the observed differences in body fat percentages to assess their real-world implications.

5. Estimate Confidence Intervals and Assess Sampling Variability in Data:

- Estimate confidence intervals for the mean weight of individuals and interpret their significance in parameter estimation.
- Investigate how varying sample sizes influence the width of confidence intervals, offering insights into the impact of sample size on precision.
- Understand the relationship between confidence levels and the precision of parameter estimates, demonstrating the trade-off between precision and confidence.

Project Implementation:

- Python programming and various statistical libraries (NumPy, SciPy, Pandas) are used for data analysis, hypothesis testing, and probability calculations.
- Data visualizations are created using Matplotlib and Seaborn to facilitate a better understanding of attribute distributions.
- Statistical concepts are applied to draw meaningful insights from the dataset, and hypothesis tests are employed to make statistical inferences.

Conclusion:

- The project endeavors to deliver comprehensive insights into body performance trends and factors by employing statistical analyses and probability concepts. These insights will aid the research team in making informed decisions and better understanding the dataset's characteristics.

2. Project Technical Environment

Programming Language:

- **Python**: Python is a versatile and widely-used programming language for data analysis due to its rich ecosystem of libraries and tools.

Integrated Development Environment (IDE):

- **Jupyter Notebook**: Jupyter Notebook provides an interactive environment for data analysis, allowing you to combine code, visualizations, and documentation in a single notebook.

Libraries and Packages:

- **NumPy**: NumPy is essential for numerical operations, array handling, and mathematical functions.
- **Pandas**: Pandas is used for data manipulation and analysis, particularly for handling datasets in DataFrame structures.
- **SciPy**: SciPy provides additional scientific and statistical functions that complement NumPy.
- **Matplotlib and Seaborn**: These libraries are used for creating data visualizations, including plots, charts, histograms, and box plots.
- **Statsmodels**: Statsmodels is valuable for statistical modeling, hypothesis testing, and regression analysis.
- **Scipy.stats**: This module within SciPy offers a wide range of statistical functions and probability distributions for hypothesis testing and probability calculations.

Data Storage and Manipulation:

- **Pandas DataFrames**: Data is typically loaded into Pandas DataFrames for manipulation and analysis.
- **CSV Files**: Data may be stored in CSV files, and the `pandas.read_csv()` function is used to read data from these files.

Statistical Techniques and Concepts:

- The project may involve a variety of statistical techniques, such as descriptive statistics, hypothesis testing, confidence intervals, probability distributions, and effect size measurements.

Analytical Techniques:

1. **Descriptive Statistics:**
   - **Mean, Median, and Standard Deviation**: Calculated to understand the central tendency and variability of numerical attributes such as age, weight, and height.
   - **Histograms and Box Plots**: Used to visualize the distribution and spread of data for key attributes.
2. **Probability Concepts:**
   - **Probability Distributions**: Applied to analyze and model uncertain events and outcomes.
   - **Normal Distribution**: Assumed for estimating probabilities, especially for blood pressure levels.
   - **Binomial Distribution**: Employed for analyzing discrete outcomes, like gender distribution.
3. **Hypothesis Testing:**
   - **Two-Sample T-Test**: Utilized to determine if body fat percentages differ significantly between genders.
   - **T-Statistics and P-Values**: Calculated to assess the significance of observed differences.
   - **Effect Size Measurement (Cohen's d)**: Used to evaluate the practical significance of differences.
4. **Confidence Intervals:**
   - **Estimation of Confidence Intervals**: Employed to estimate the range of plausible values for the mean weight of individuals.
   - **Sampling Variability Assessment**: Studied to understand how sample sizes affect confidence interval width.
   - **Impact of Confidence Levels**: Analyzed to evaluate how different confidence levels affect precision.

Analytical Tools:

1. **Programming Language:**
   - **Python**: Python is the primary programming language used for data analysis, offering a wide range of libraries for statistical analysis and data visualization.
2. **Integrated Development Environment (IDE):**
   - **Jupyter Notebook**: Jupyter Notebook is employed for interactive coding, combining code, visualizations, and documentation.
3. **Libraries and Packages:**
   - **NumPy**: Used for numerical operations and array handling.
   - **Pandas**: Utilized for data manipulation and analysis with DataFrames.
   - **SciPy**: Provides additional scientific and statistical functions.
   - **Matplotlib and Seaborn**: Used for creating data visualizations.
   - **Statsmodels**: Applied for statistical modeling and regression analysis.
   - **Scipy.stats**: Offers statistical functions and probability distributions.

Data Acquisition

Data acquisition is the process of collecting and gathering data from various sources to be used for analysis, research, or decision-making. In the context of the project you described, which involves analyzing body performance and health-related data, data acquisition typically involves the following steps:

1. **Data Source Identification:** Determine where the relevant data is located. In your project, you may have identified specific sources or datasets that contain information about age, gender, weight, height, body fat percentage, blood pressure, grip force, and other attributes.
2. **Data Collection:** Collect the data from the identified sources. This may involve downloading datasets from online repositories, accessing databases, or gathering data through surveys, experiments, or measurements.
3. **Data Cleaning:** Data collected from various sources may contain errors, missing values, or inconsistencies. Data cleaning involves identifying and addressing these issues to ensure the data is accurate and reliable.
4. **Data Integration:** If your project involves multiple datasets from different sources, you may need to integrate them into a unified dataset. This process ensures that all relevant data is available for analysis.
5. **Data Transformation:** Depending on the analysis tasks you plan to perform, you may need to transform the data. This can include standardizing units of measurement, converting data types, or creating new variables.
6. **Data Storage:** Store the cleaned and transformed data in a suitable format for analysis. Common formats include CSV files, databases, or dataframes in Python.
7. **Data Documentation:** Document the data acquisition process, including the sources, collection methods, and any transformations applied. Proper documentation is essential for transparency and reproducibility.
8. **Data Privacy and Security:** Ensure that you handle the data in compliance with privacy and security regulations. This may involve anonymizing sensitive information or securing access to the data.
9. **Data Quality Assurance:** Perform checks to verify the quality of the acquired data. This may include running data validation scripts, conducting exploratory data analysis, or visualizing the data to identify outliers or anomalies.
10. **Data Version Control:** If your project is ongoing or collaborative, consider implementing version control for your data to track changes and maintain a history of data modifications.
11. **Backup and Recovery:** Implement backup procedures to prevent data loss and have a recovery plan in case of unexpected data issues.
12. **Data Ethics:** Ensure that your data acquisition process follows ethical guidelines and respects the rights and privacy of individuals whose data is included in the analysis.

5. Descriptive statistics analysis:

Descriptive statistics analysis is a fundamental step in data analysis, which involves summarizing and describing key characteristics of a dataset. It provides an overview of the data, helps identify patterns, trends, and anomalies, and prepares the data for more advanced analyses. In the context of your project, here's how you can perform a descriptive statistics analysis:

1. Calculate Summary Statistics:

- **Mean:** Calculate the average value for numerical attributes like age, weight, and height. The mean provides a measure of central tendency.
- **Median:** Find the middle value when the data is sorted. The median is another measure of central tendency and is less sensitive to outliers than the mean.
- **Standard Deviation:** Calculate the standard deviation to measure the spread or variability of the data. A higher standard deviation indicates greater variability.

2. Visualize the Data:

- **Histograms:** Create histograms for numerical attributes to visualize their distribution. Histograms show the frequency of values within specified bins.
- **Box Plots:** Generate box plots to display the distribution, median, quartiles, and outliers of numerical attributes. Box plots are useful for identifying skewness and outliers.

3. Interpret Descriptive Statistics:

- **Mean:** Interpret the mean to understand the average value of attributes. For example, the mean age can provide insights into the typical age of individuals in the dataset.
- **Median:** Use the median to identify the middle value of attributes, especially when dealing with skewed distributions. It provides an alternative measure of central tendency.
- **Standard Deviation:** Interpret the standard deviation to gauge the variability or dispersion of data points. A higher standard deviation suggests greater data spread.

4. Identify Patterns and Anomalies:

- Analyze the distribution of data through histograms and box plots to identify patterns such as bimodality, skewness, or outliers.
- Examine relationships between variables to identify potential correlations or dependencies that may be explored further.

5. Prepare Data for Advanced Analysis:

- Address any data quality issues discovered during the descriptive statistics analysis. This may involve handling missing values or outliers.
- Determine if data transformations, such as normalization or standardization, are needed for specific analyses.

6. Document Findings:

- Document the results of your descriptive statistics analysis, including summary statistics, visualizations, and any insights gained from the analysis.
- Use clear and concise descriptions in your documentation to facilitate communication with other team members or stakeholders.

6.Probability Analysis

Probability analysis is a statistical technique used to quantify and understand uncertainty in outcomes and events. In your project, probability analysis can be applied in various ways to analyze and interpret different aspects of the dataset. Here are some key areas where probability analysis can be useful:

1. **Gender Distribution Analysis:**
   - Analyze the gender distribution within the dataset using probability concepts.
   - Calculate the probabilities of being male or female based on the dataset's gender distribution.
2. **Blood Pressure Probability Estimation:**
   - Estimate the probability of specific blood pressure levels using assumptions of the normal distribution.
   - Utilize the normal distribution to model and analyze the distribution of diastolic and systolic blood pressure measurements.
3. **Discrete Outcomes Analysis:**
   - Use binomial distribution to analyze discrete outcomes, such as the probability of being male or female.
   - Apply binomial probabilities to study other binary or categorical attributes within the dataset.
4. **Random Variable Analysis:**
   - Identify random variables in the dataset. For example, diastolic blood pressure or grip force measurements can be considered random variables.
   - Explore the probability distributions of these random variables to understand their characteristics and likelihood of specific values or ranges.
5. **Unusual Value Identification:**
   - Apply probability distributions to identify unusual or extreme values within the dataset.
   - Determine the probability of observing values that fall outside certain thresholds or bounds.
6. **Confidence Interval Estimation:**
   - While not a traditional probability analysis, confidence intervals are used to estimate the range of plausible values for certain parameters, such as the mean weight of individuals.
   - Understand the concept that a confidence interval represents a range of values with a specified level of confidence.
7. **Assessment of Uncertainty:**
   - Probability analysis can help quantify and communicate the uncertainty associated with various outcomes, measurements, or predictions within the dataset.
   - Assess how different sources of uncertainty may impact the results of your analyses.
8. **Hypothesis Testing:**
   - Probability plays a central role in hypothesis testing, where you calculate p-values to assess the significance of observed differences or relationships.
   - Understand that p-values represent the probability of obtaining results as extreme as the observed ones under the null hypothesis.

7. Hypothesis Testing

Hypothesis testing is a statistical method used to make inferences about a population based on a sample of data. It allows you to test whether certain assumptions or hypotheses about the population are supported by the evidence from your data. In your project, which involves analyzing body performance and health-related data, hypothesis testing can be a valuable tool for drawing conclusions and making statistical inferences. Here are the key steps and concepts involved in hypothesis testing:

1. Formulate Hypotheses:

- **Null Hypothesis (H0):** This is the default hypothesis that there is no significant effect, relationship, or difference in the population. It represents the status quo or a statement of no effect.
- **Alternative Hypothesis (Ha):** This is the hypothesis you want to test. It suggests that there is a significant effect, relationship, or difference in the population.

2. Select a Significance Level (Alpha):

- The significance level (α) represents the probability of making a Type I error (rejecting the null hypothesis when it is true). Common choices for α include 0.05 and 0.01, but it can vary depending on the context and the level of confidence required.

3. Collect and Analyze Data:

- Collect the relevant data from your dataset for the hypothesis test you want to perform.
- Conduct the appropriate statistical test based on the characteristics of your data and research question.

4. Calculate Test Statistic:

- The test statistic is a numerical value calculated from your sample data that quantifies the evidence against the null hypothesis.
- The choice of test statistic depends on the type of data and the hypothesis being tested. Common test statistics include t-statistics, z-scores, chi-squared statistics, and more.

5. Determine the P-Value:

- The p-value is the probability of obtaining results as extreme as, or more extreme than, those observed in your sample, assuming that the null hypothesis is true.
- A small p-value (typically less than α) suggests strong evidence against the null hypothesis, leading to its rejection.
- A large p-value suggests weak evidence against the null hypothesis, indicating that you fail to reject it.

6. Make a Decision:

- Compare the calculated p-value to the chosen significance level (α).
- If the p-value is less than or equal to α, you reject the null hypothesis in favor of the alternative hypothesis.
- If the p-value is greater than α, you fail to reject the null hypothesis.

7. Interpret the Results:

- If you reject the null hypothesis, it suggests that there is evidence to support the alternative hypothesis.
- If you fail to reject the null hypothesis, it means that there is insufficient evidence to support the alternative hypothesis.

8. Confidence Intervals

Confidence intervals (CIs) are a statistical concept and technique used to estimate a range of plausible values for a population parameter, such as the mean, proportion, or variance. They provide a measure of uncertainty and are commonly used in inferential statistics. In your project, which involves analyzing body performance and health-related data, confidence intervals can be applied in various contexts. Here's an overview of confidence intervals and their use:

1. Estimation of Population Parameters:

- The primary purpose of confidence intervals is to estimate a population parameter, such as the mean weight of individuals, with a certain level of confidence.

2. Confidence Level (1 - α):

- A confidence level (1 - α) is chosen to represent the level of confidence in the estimation. Common choices include 95%, 99%, or other values.
- A 95% confidence interval, for example, implies that if you were to repeat the sampling and estimation process many times, approximately 95% of the resulting intervals would contain the true population parameter.

3. Calculation of Confidence Interval:

- The confidence interval is calculated using a formula that depends on the type of parameter being estimated (e.g., mean, proportion) and the distribution of the sample data.
- For estimating the mean of a normally distributed population, the formula often used is: $CI = \bar{X} \pm Z * (\sigma/\sqrt{n})$, where $\bar{X}$ is the sample mean, Z is the critical value from the standard normal distribution corresponding to the desired confidence level, σ is the population standard deviation (if known), and n is the sample size.
- For proportions, the formula for a confidence interval is different and is based on the binomial distribution.

4. Interpretation:

- The confidence interval is typically presented as a range of values, such as [lower bound, upper bound].
- It means that you can be (1 - α) * 100% confident that the true population parameter falls within this range.
- A narrower confidence interval indicates more precise estimation, while a wider interval indicates greater uncertainty.

5. Implications:

- If the confidence interval for a parameter contains a specific value (e.g., a null hypothesis value), it suggests that there is insufficient evidence to reject that value.
- If the interval does not contain a specific value, it may indicate evidence against that value.

6. Sample Size Impact:

- The width of a confidence interval is affected by the sample size. Larger samples tend to result in narrower intervals, providing more precise estimates.

7. Practical Significance:

- Alongside statistical significance, consider the practical significance of the confidence interval. Even if a parameter estimate is statistically significant, it may not be practically significant if the range of plausible values is too narrow or too wide.

9.Data Visualization: Data Visualization

Data visualization is a critical component of data analysis that involves representing data graphically to better understand patterns, trends, and relationships within the dataset. In my project, which involves analyzing body performance and health-related data, data visualization can help you convey insights and findings effectively. Here are some key aspects and techniques of data visualization:

1. Types of Data Visualization:

- **Scatter Plots:** Use scatter plots to visualize the relationship between two continuous variables. For example, you can create a scatter plot to explore the relationship between weight and height.
- **Histograms:** Create histograms to display the distribution of a single continuous variable. Histograms can help identify data skewness, central tendency, and variability.
- **Box Plots:** Box plots (box-and-whisker plots) are useful for visualizing the distribution, quartiles, and outliers of a numerical variable. They are particularly helpful for comparing multiple distributions.
- **Bar Charts:** Bar charts are suitable for displaying categorical data, such as gender distribution or class distribution. They can also be used for comparing categories across a single variable.
- **Line Charts:** Line charts are effective for showing trends and changes over time. For instance, you can create a line chart to visualize how sit-up counts change with age.
- **Heatmaps:** Heatmaps are useful for displaying patterns and relationships in a matrix or grid-like dataset. They are commonly used for correlation matrices or categorical data.

- **Pie Charts:** Pie charts are used to represent the composition of a whole. While they are less common in data analysis, they can be helpful for showing proportions.

2. Data Visualization Libraries:

- **Matplotlib:** Matplotlib is a popular Python library for creating a wide range of static 2D plots and charts.
- **Seaborn:** Seaborn is built on top of Matplotlib and provides a high-level interface for creating informative and attractive statistical graphics.
- **Plotly:** Plotly is a versatile library that allows for interactive and web-based visualizations. It can be used to create dynamic plots and dashboards.
- **ggplot2:** If you are using R, ggplot2 is a powerful package for creating elegant and customizable graphics.

3. Data Labeling and Annotation:

- Always label your axes and provide titles for your visualizations to make them self-explanatory.
- Use annotations and legends to clarify points of interest or provide additional context in your plots.

4. Interactive Visualizations (Optional):

- If appropriate, consider creating interactive visualizations that allow users to explore data and interact with charts for a deeper understanding.

5. Dashboard Creation (Optional):

- Combine multiple visualizations into a dashboard or report to present a comprehensive overview of your findings.

6. Data Exploration:

- Use data visualization as an exploratory tool to identify outliers, patterns, and potential research directions.

7. Communication:

- Visualizations are a powerful means of communicating your findings to both technical and non-technical audiences. They make complex data more accessible and understandable.

10	**Interpretation of Findings**

1. Insights and Patterns:

In this section, provide a detailed analysis of the insights and patterns you've discovered from your data analysis. Break down the discussion by attributes or variables that are relevant to your project, such as age, gender, weight, height, body fat percentage, blood pressure, etc. For each attribute, consider the following:

- **Describe Trends:** Explain any trends or patterns you observed. For example, if you found that body fat percentage tends to increase with age, discuss the nature of this relationship (linear, nonlinear) and any significant deviations.
- **Highlight Correlations:** Discuss any correlations between variables. If, for instance, there's a strong correlation between weight and blood pressure, explain its strength and direction.
- **Identify Outliers:** Mention any outliers or unusual data points that may have a significant impact on your findings. Explain whether these outliers were addressed in your analysis.
- **Group Comparisons:** If relevant, compare and contrast findings between different groups, such as gender-based differences in body fat percentage.

2. Practical Implications:

In this subsection, consider the real-world implications of your findings. Discuss how the insights and patterns you've identified could impact body performance, health, or related fields. Here are some points to cover:

- **Health Implications:** Explain how the observed patterns and trends might affect an individual's health or well-being. For example, if you found that high body fat percentages are associated with higher diastolic blood pressure, discuss the potential health risks and the importance of managing body fat.
- **Interventions:** Consider whether your findings suggest potential interventions or strategies for improving health or performance. For instance, if there's a strong negative correlation between grip force and age, you might discuss the importance of strength training for older individuals.
- **Data-Driven Decision-Making:** Explain how your findings can inform decision-makers, researchers, or healthcare professionals. Emphasize the value of data-driven decision-making based on your analysis.

3. Recommendations:

In this section, provide actionable recommendations based on your analysis and the practical implications you've discussed. Consider the following:

- **Health Recommendations:** If relevant, offer recommendations for maintaining or improving health and body performance based on your findings. These could include lifestyle changes, exercise regimens, or health screenings.
- **Research Recommendations:** Suggest areas for further research or data collection that could build upon your findings. Identify gaps in the current dataset that future studies could address.
- **Policy and Practice:** If applicable, recommend policy changes or adjustments to clinical practices based on your insights. Discuss how your analysis could influence decision-making in these areas.

11. Conclusion o Summary of Project o Key Findings

The "Conclusion" section of your project report serves as a summary of your entire analysis, highlighting the main points and key takeaways. Here's how to structure the conclusion with a summary of the project and key findings:

1. Summary of Project:

- Begin by briefly summarizing the main objectives and scope of your project. Provide context for the analysis you conducted.
- Mention any specific research questions or hypotheses you aimed to address.

2. Key Findings:

- Concisely list and summarize the most significant findings from your analysis. Highlight the key insights and patterns you've identified in the data.
- Consider organizing the key findings by category or attribute. For example, you can group findings related to age, gender, body fat percentage, and other relevant variables.
- Emphasize the statistical significance of findings where applicable. Mention any correlations, trends, or relationships that were particularly noteworthy.
- Use clear and concise language to communicate your findings. Avoid jargon or technical details in this section.

3. Implications:

- Discuss the implications of your findings in the context of your project's goals and objectives. Explain why these findings are important and how they contribute to the understanding of body performance trends and factors.
- Consider both the practical and theoretical implications. How do your findings impact the field of body performance, health, or related areas?
- If there are any unexpected or surprising findings, acknowledge them and provide possible explanations or avenues for further research.

4. Contributions and Significance:

- Reflect on the contributions of your project to the broader field of study. Explain how your analysis adds to existing knowledge and research in body performance and health-related areas.
- Discuss the practical significance of your findings. How can they inform decision-making, interventions, or policies related to body performance and health?

5. Limitations and Future Directions:

- Acknowledge any limitations or constraints of your analysis. This could include data limitations, assumptions, or potential sources of bias.
- Suggest areas for future research or data collection that could build upon your findings and address the limitations you've identified.

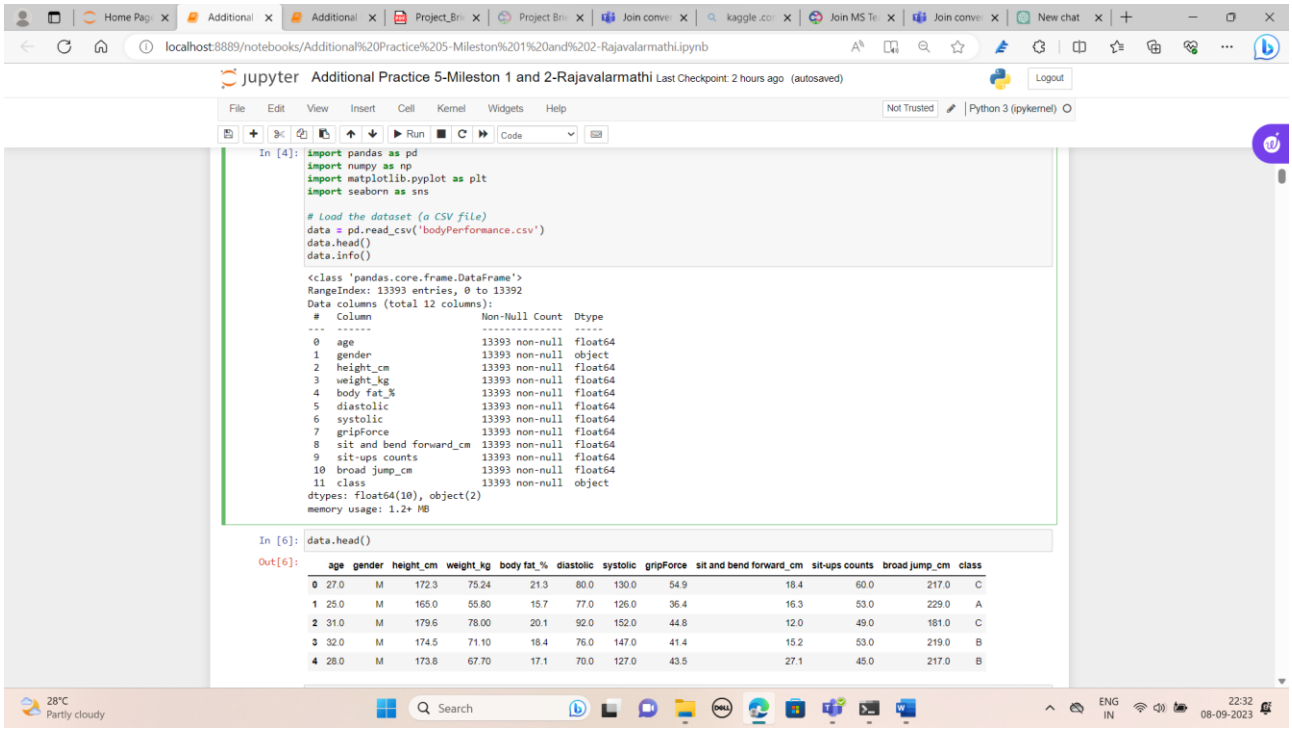
Appendices : Python Code o Additional Visual

Milestone 1and 2:

Project Task List
For the above case study, perform the following project activities:
1. Calculate Descriptive Statistics Measures for Data Analysis
• Calculate summary statistics such as mean, median, and standard deviation for attributes like age, weight, and height.
• Visualize the distribution of key attributes using histograms and box plots.
• Interpret the descriptive statistics to understand the characteristics of the dataset.

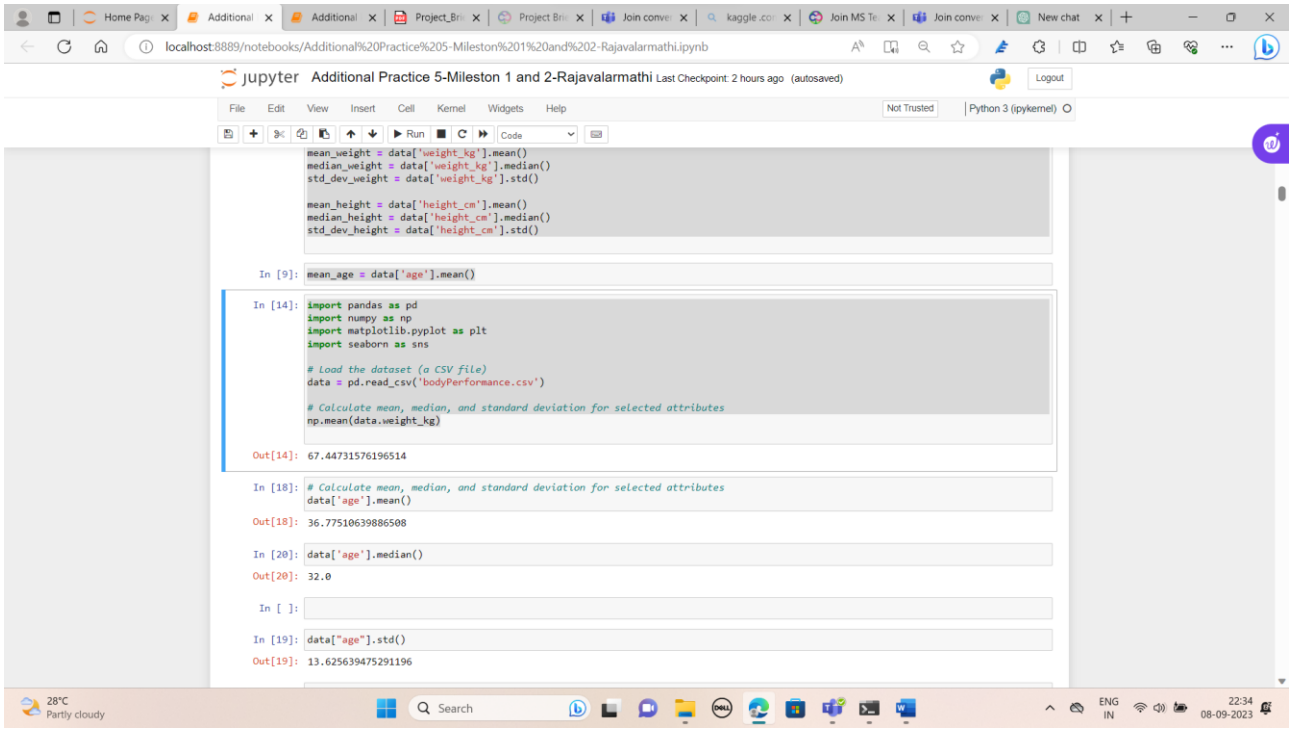
Step 1: Import Libraries and Load Data

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

# Load the dataset (a CSV file)
data = pd.read_csv('bodyPerformance.csv')
data.head()
data.info()
```

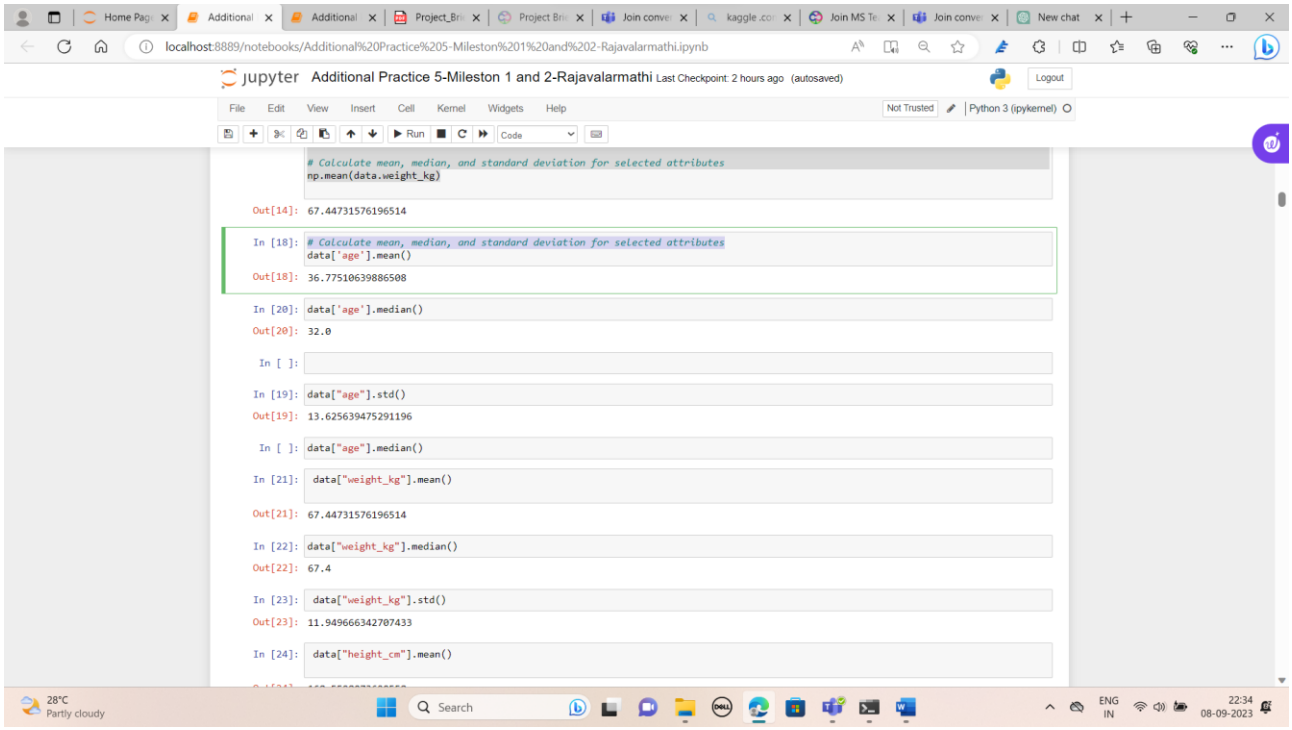


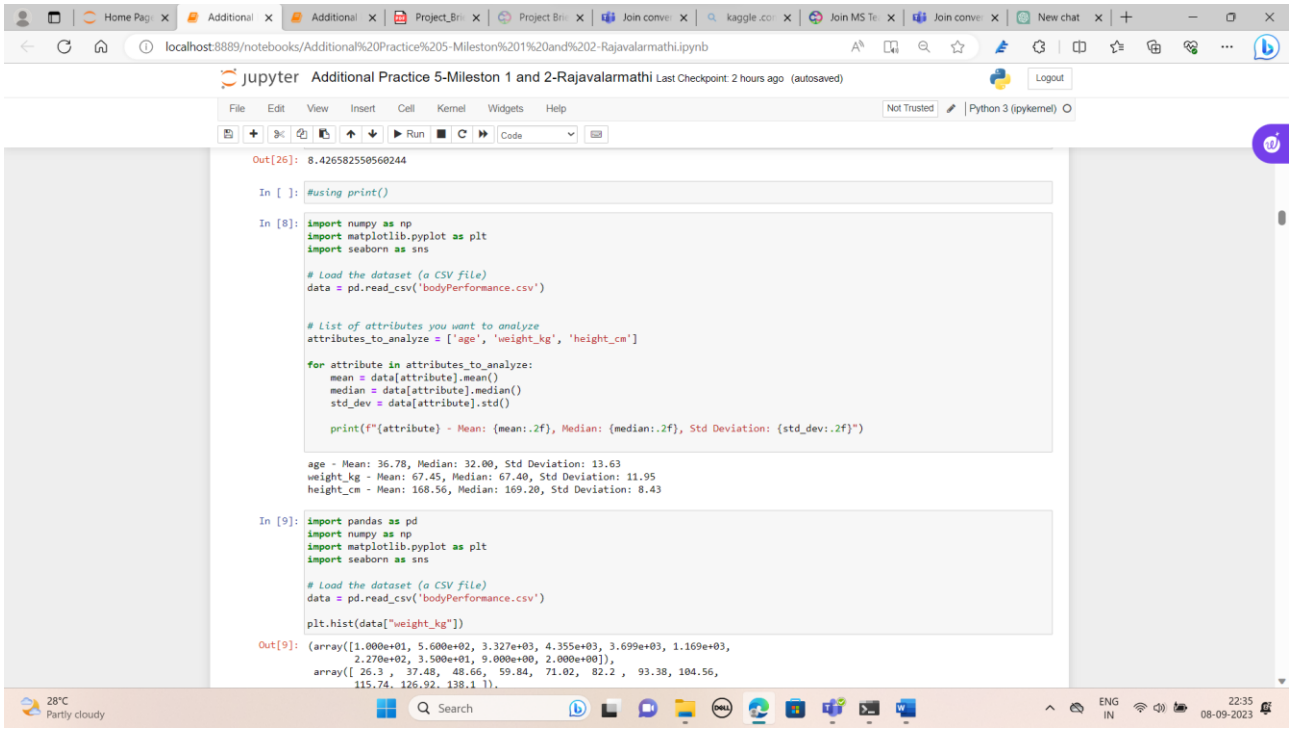
Step 2: Calculate Descriptive Statistics:

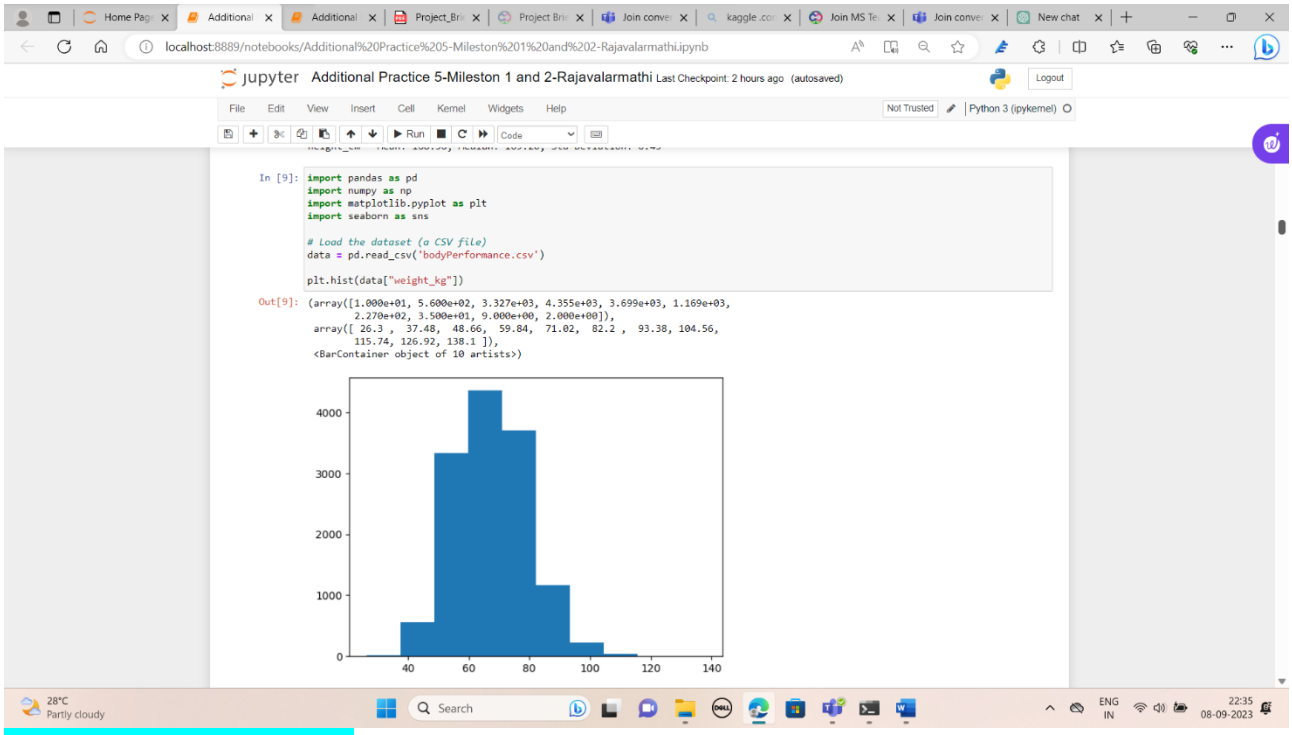
```
# Calculate mean, median, and standard deviation for selected attributes
mean_age = data['age'].mean()
median_age = data['age'].median()
std_dev_age = data['age'].std()

mean_weight = data['weight_kg'].mean()
median_weight = data['weight_kg'].median()
std_dev_weight = data['weight_kg'].std()

mean_height = data['height_cm'].mean()
median_height = data['height_cm'].median()
std_dev_height = data['height_cm'].std()

mean_age = data['age'].mean()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset (a CSV file)
data = pd.read_csv('bodyPerformance.csv')

# Calculate mean, median, and standard deviation for selected attributes
np.mean(data.weight_kg)
```



Calculate mean, median, and standard deviation for selected attributes



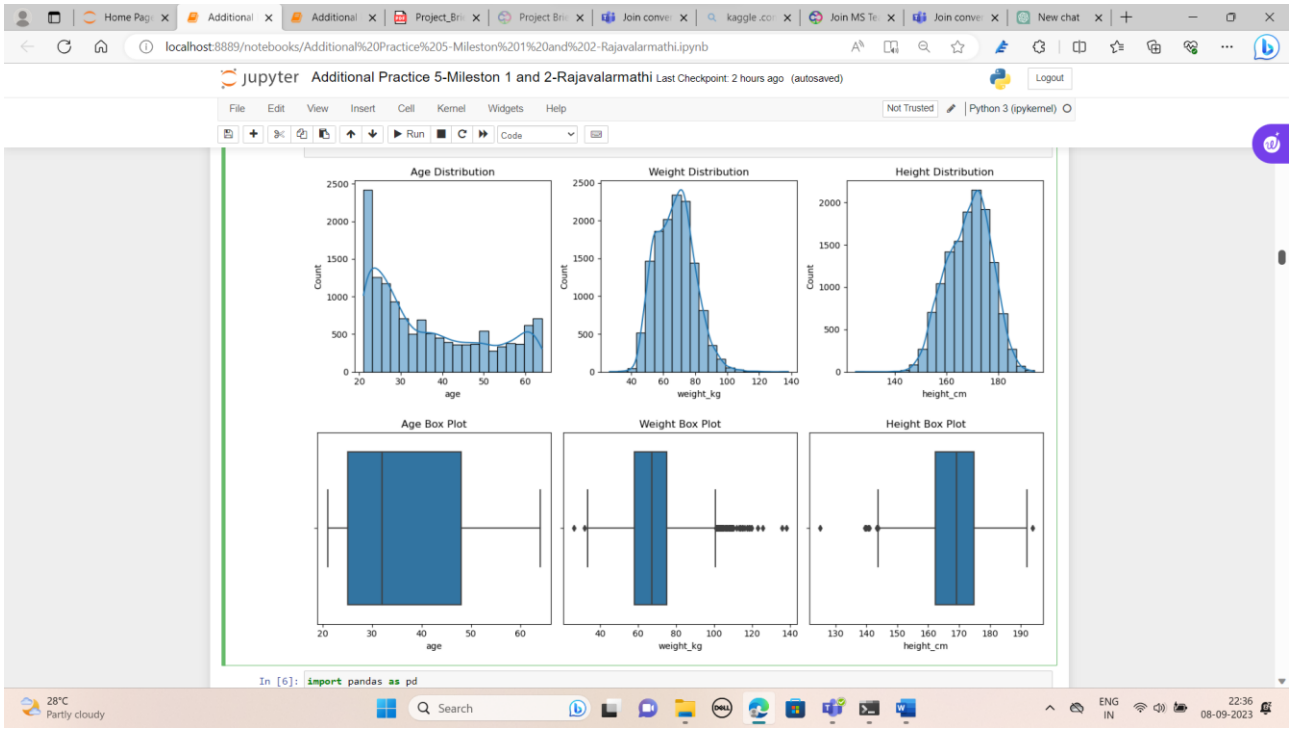




Step 3: Visualize the Data

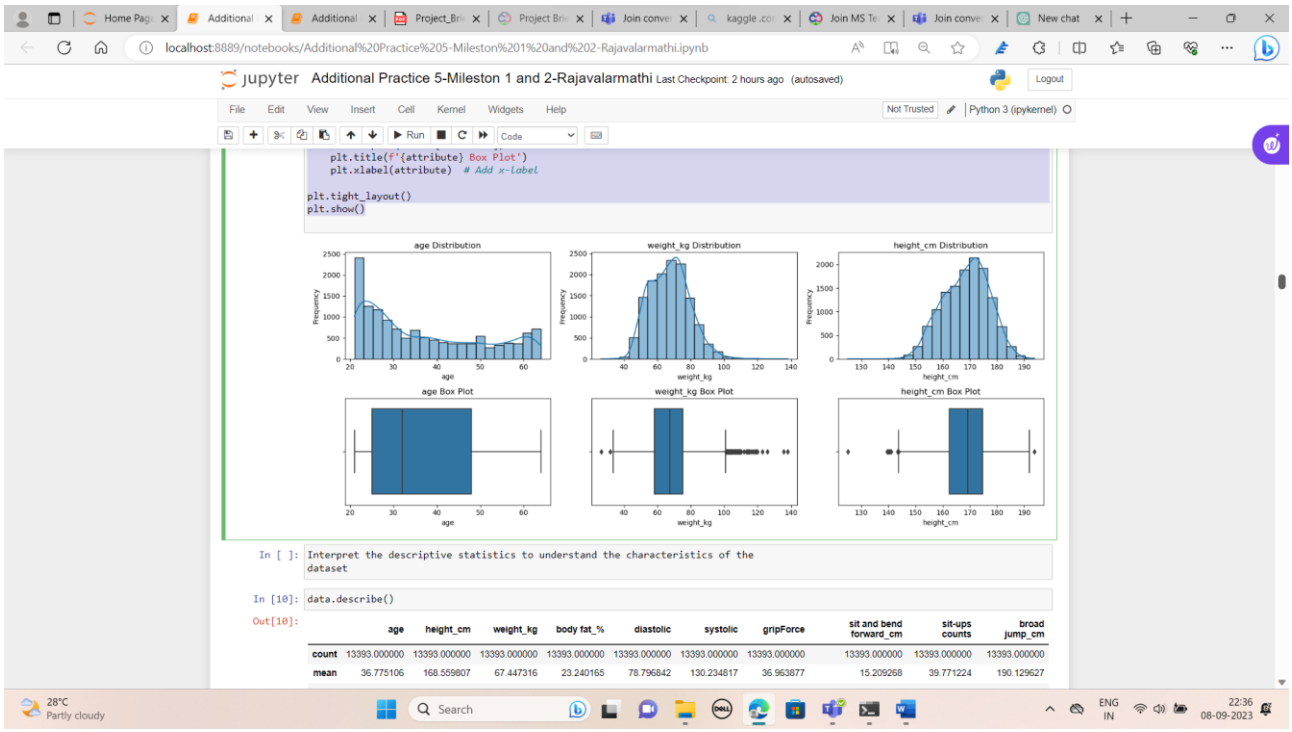
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Load the dataset (a CSV file)
data = pd.read_csv('bodyPerformance.csv')
# Create histograms for age, weight, and height
plt.figure(figsize=(12, 4))
plt.subplot(131)
sns.histplot(data['age'], bins=20, kde=True)
plt.title('Age Distribution')

plt.subplot(132)
sns.histplot(data['weight_kg'], bins=20, kde=True)
plt.title('Weight Distribution')

plt.subplot(133)
sns.histplot(data['height_cm'], bins=20, kde=True)
plt.title('Height Distribution')

plt.tight_layout()
plt.show()

# Create box plots for age, weight, and height
plt.figure(figsize=(12, 4))
plt.subplot(131)
sns.boxplot(x=data['age'])
plt.title('Age Box Plot')

plt.subplot(132)
sns.boxplot(x=data['weight_kg'])
plt.title('Weight Box Plot')

plt.subplot(133)
sns.boxplot(x=data['height_cm'])
plt.title('Height Box Plot')

plt.tight_layout()
plt.show()
```

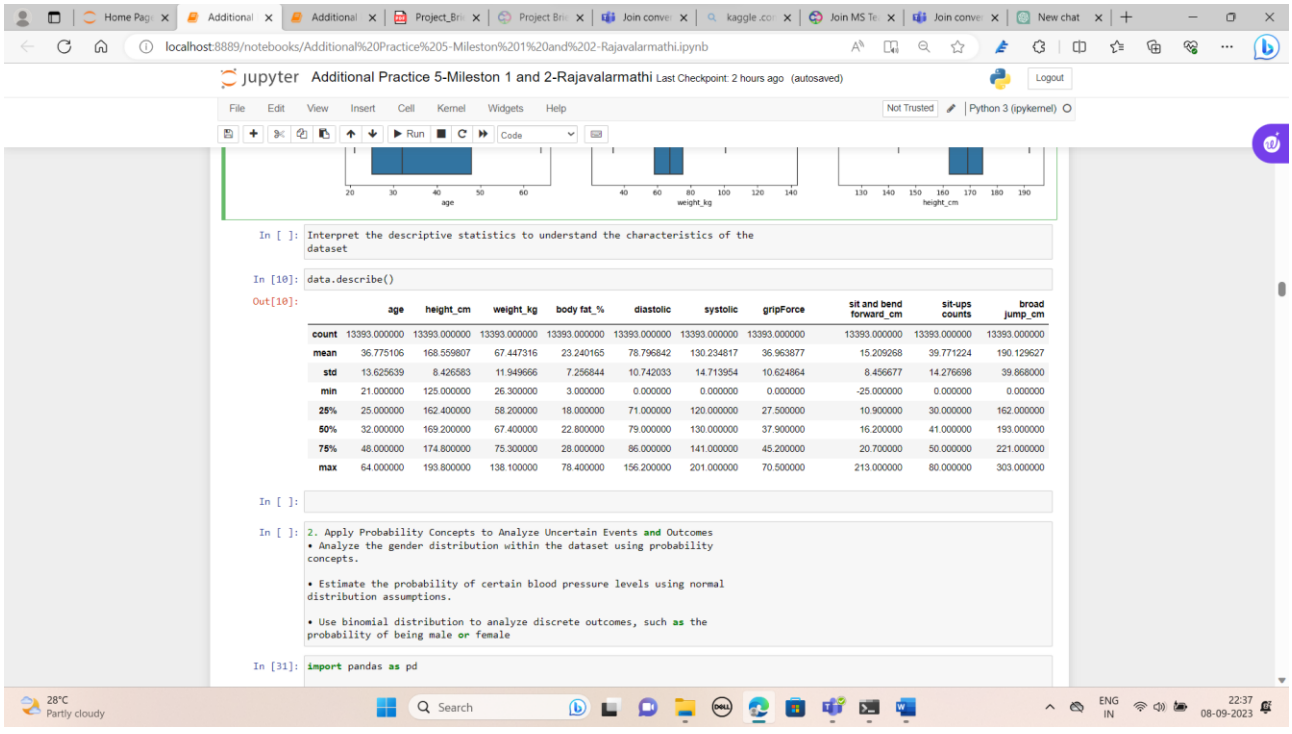



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Load  dataset
data = pd.read_csv('bodyPerformance.csv')

# List of attributes  want to analyze
attributes_to_analyze = ['age', 'weight_kg', 'height_cm']

# Visualize the distribution of attributes using histograms and box plots
plt.figure(figsize=(15, 6))

# Create histograms
for i, attribute in enumerate(attributes_to_analyze, 1):
    plt.subplot(2, 3, i)
    sns.histplot(data[attribute], bins=20, kde=True)
    plt.title(f'{attribute} Distribution')
    plt.xlabel(attribute)  # Add x-label
    plt.ylabel('Frequency')  # Add y-label

# Create box plots
for i, attribute in enumerate(attributes_to_analyze, 4):
    plt.subplot(2, 3, i)
    sns.boxplot(x=data[attribute])
    plt.title(f'{attribute} Box Plot')
    plt.xlabel(attribute)  # Add x-label
```

```
plt.tight_layout()
plt.show()
```



	age	height_cm	weight_kg	body fat_%	diastolic	systolic	gripForce	sit and bend forward_cm	sit-ups counts	broad jump_cm
count	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000
mean	36.775106	168.559807	67.447316	23.240165	78.796842	130.234817	36.963877	15.209268	39.771224	190.129627



	age	height_cm	weight_kg	body fat_%	diastolic	systolic	gripForce	sit and bend forward_cm	sit-ups counts	broad jump_cm
count	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000	13393.000000
mean	36.775106	168.559807	67.447316	23.240165	78.796842	130.234817	36.963877	15.209268	39.771224	190.129627
std	13.625639	8.426583	11.949666	7.256844	10.742033	14.713954	10.624864	8.456677	14.276698	39.868000
min	21.000000	125.000000	26.300000	3.000000	0.000000	0.000000	0.000000	-25.000000	0.000000	0.000000
25%	25.000000	162.400000	58.200000	18.000000	71.000000	120.000000	27.500000	10.900000	30.000000	162.000000
50%	32.000000	169.200000	67.400000	22.800000	79.000000	130.000000	37.900000	16.200000	41.000000	193.000000
75%	48.000000	174.800000	75.300000	28.000000	86.000000	141.000000	45.200000	20.700000	50.000000	221.000000
max	64.000000	193.800000	138.100000	78.400000	156.200000	201.000000	70.500000	213.000000	80.000000	303.000000

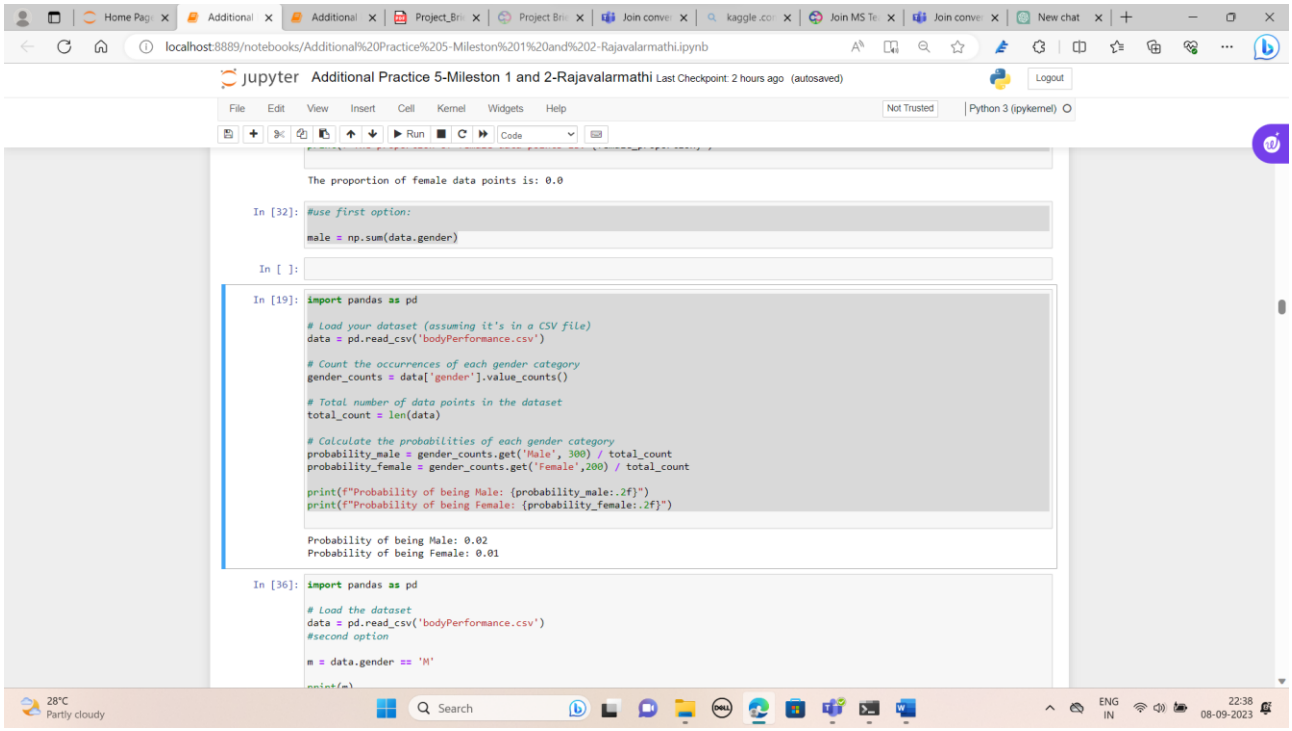
2. Apply Probability Concepts to Analyze Uncertain Events and Outcomes

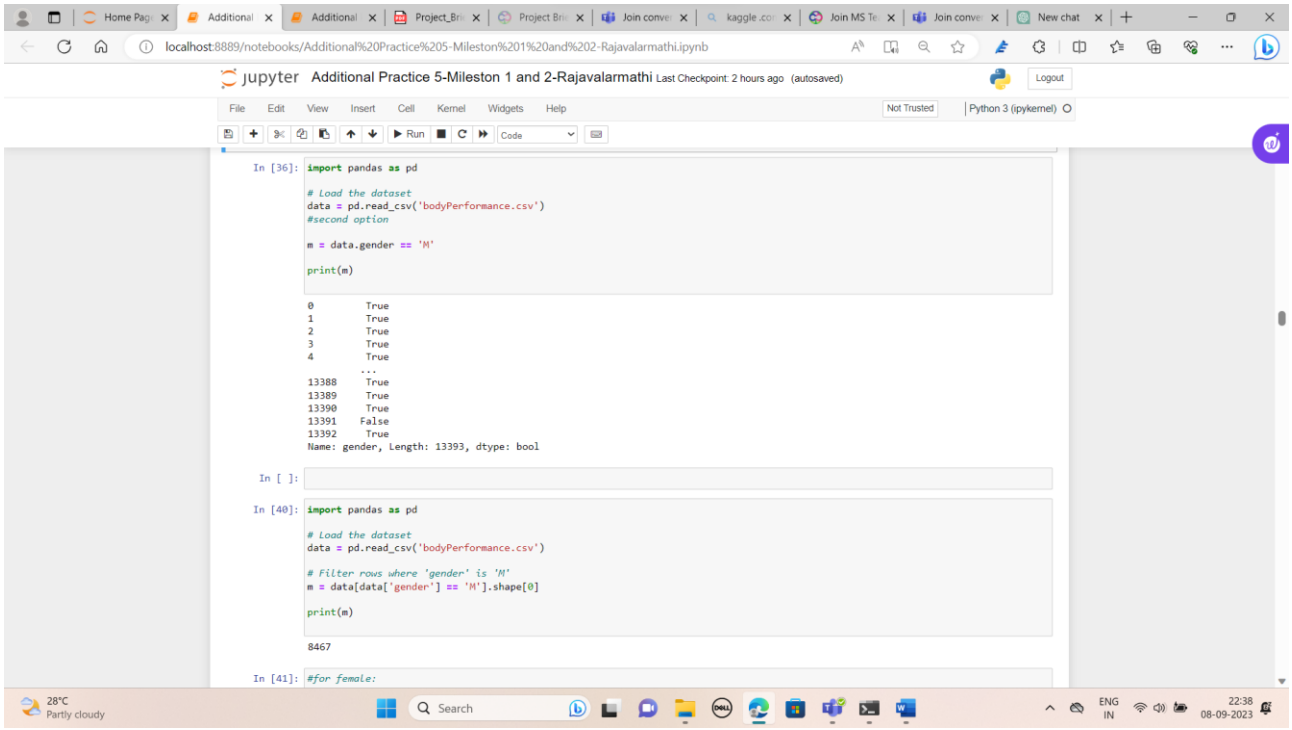
• Analyze the gender distribution within the dataset using probability concepts.

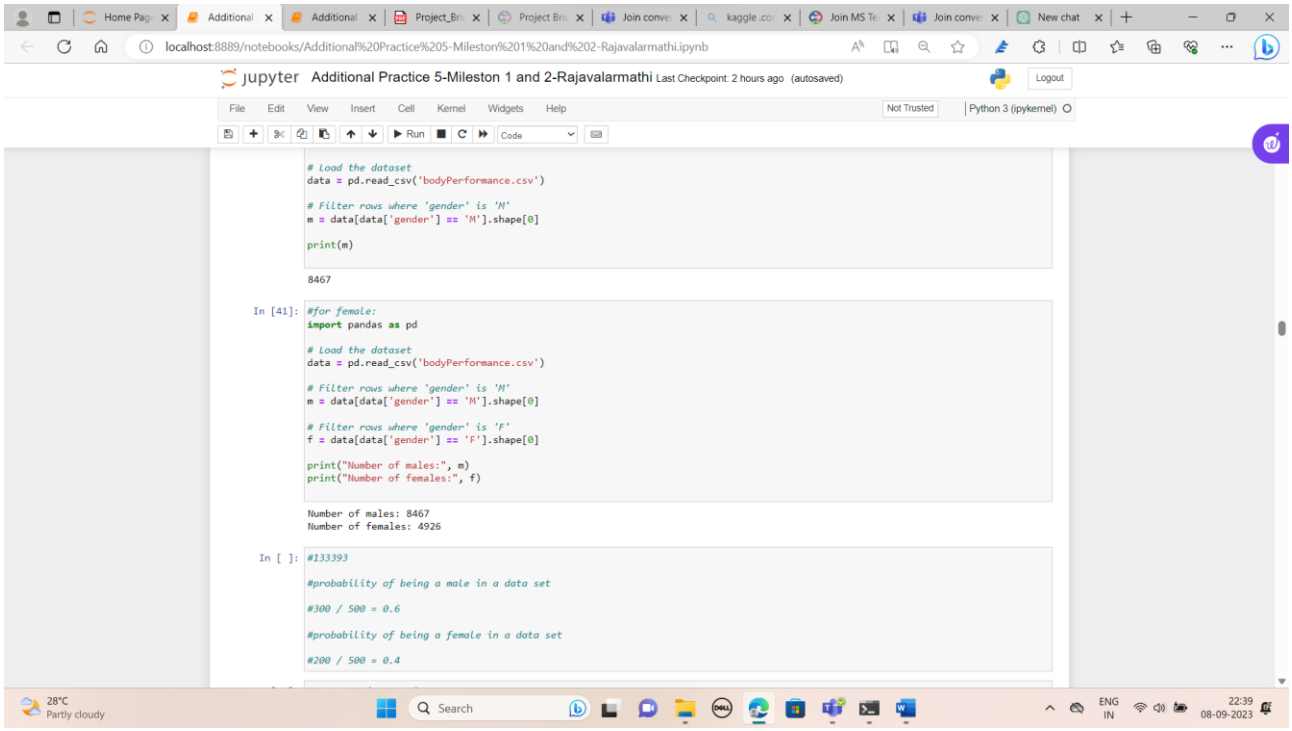
• Estimate the probability of certain blood pressure levels using normal distribution assumptions.

• Use binomial distribution to analyze discrete outcomes, such as the probability of being male or female

```
import pandas as pd

# Load your dataset (assuming it's in a CSV file)
data = pd.read_csv('bodyPerformance.csv')


# Calculate the proportion of female data points
female_proportion = (data['gender'] == 'Female').sum() / len(data)

print(f"The proportion of female data points is: {female_proportion}")


#use first option:

male = np.sum(data.gender)


import pandas as pd

# Load your dataset (assuming it's in a CSV file)
data = pd.read_csv('bodyPerformance.csv')

# Count the occurrences of each gender category
gender_counts = data['gender'].value_counts()

# Total number of data points in the dataset
total_count = len(data)

# Calculate the probabilities of each gender category
probability_male = gender_counts.get('Male', 300) / total_count
probability_female = gender_counts.get('Female',200) / total_count

print(f"Probability of being Male: {probability_male:.2f}")
print(f"Probability of being Female: {probability_female:.2f}")
```

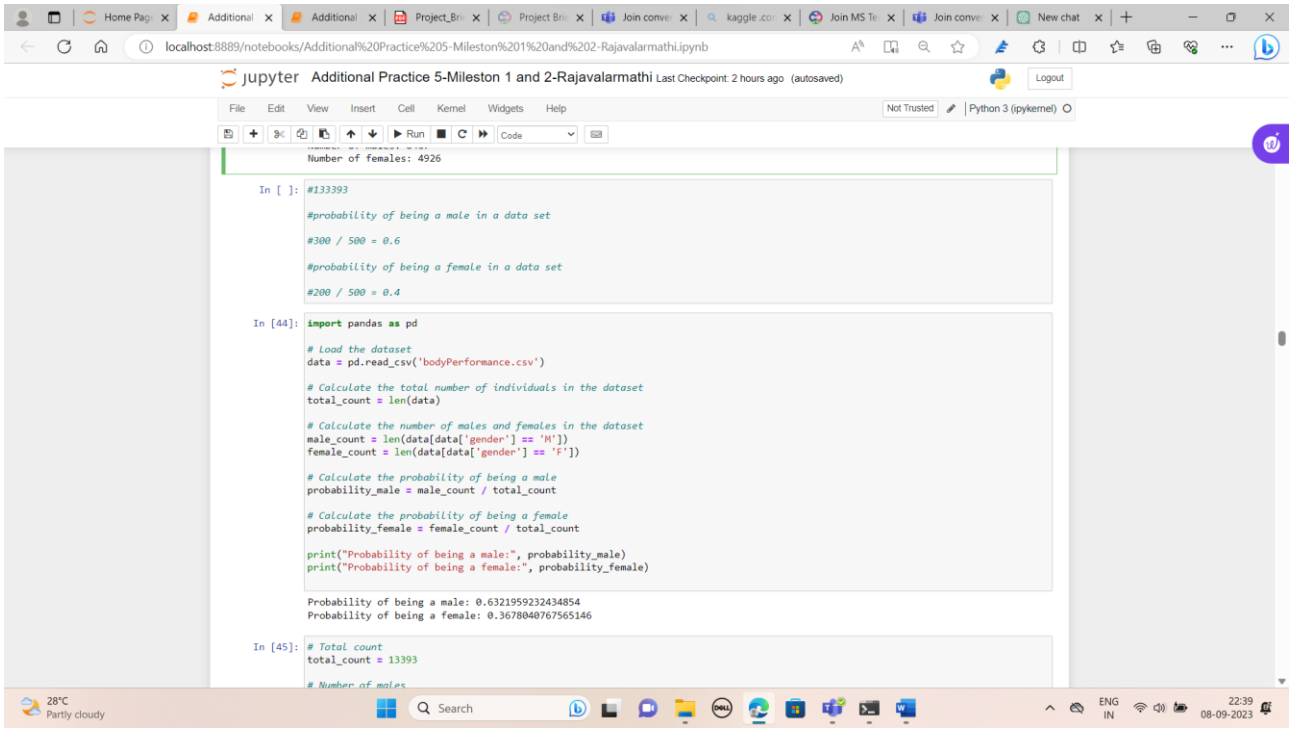

jupyter Additional Practice 5-Mileston 1 and 2-Rajavalarmathi Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
The proportion of female data points is: 0.0
In [32]:
#use first option:
male = np.sum(data.gender)
In [ ]:
In [19]:
import pandas as pd
# Load your dataset (assuming it's in a CSV file)
data = pd.read_csv('bodyPerformance.csv')
# Count the occurrences of each gender category
gender_counts = data['gender'].value_counts()
# Total number of data points in the dataset
total_count = len(data)
# Calculate the probabilities of each gender category
probability_male = gender_counts.get('Male', 300) / total_count
probability_female = gender_counts.get('Female',200) / total_count
print(f"Probability of being Male: {probability_male:.2f}")
print(f"Probability of being Female: {probability_female:.2f}")
Probability of being Male: 0.02
Probability of being Female: 0.01
In [36]:
import pandas as pd
# Load the dataset
data = pd.read_csv('bodyPerformance.csv')
#second option
m = data.gender == 'M'
28°C
Partly cloudy
Search
ENG
IN
22:38
08-09-2023


jupyter Additional Practice 5-Mileston 1 and 2-Rajavalarmathi Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [36]:
import pandas as pd
# Load the dataset
data = pd.read_csv('bodyPerformance.csv')
#second option
m = data.gender == 'M'
print(m)
0 True
1 True
2 True
3 True
4 True
...
13388 True
13389 True
13390 True
13391 False
13392 True
Name: gender, Length: 13393, dtype: bool
In [ ]:
In [40]:
import pandas as pd
# Load the dataset
data = pd.read_csv('bodyPerformance.csv')
# Filter rows where 'gender' is 'M'
m = data[data['gender'] == 'M'].shape[0]
print(m)
8467
In [41]:
#for female:
28°C
Partly cloudy
Search
ENG
IN
22:38
08-09-2023



• **Analyze the gender distribution within the dataset using probability concepts**

```
import pandas as pd

# Load the dataset
data = pd.read_csv('bodyPerformance.csv')

# Calculate the total number of individuals in the dataset
total_count = len(data)

# Calculate the number of males and females in the dataset
male_count = len(data[data['gender'] == 'M'])
female_count = len(data[data['gender'] == 'F'])

# Calculate the probability of being a male
probability_male = male_count / total_count

# Calculate the probability of being a female
probability_female = female_count / total_count

print("Probability of being a male:", probability_male)
print("Probability of being a female:", probability_female)
```



 # Total count
total_count = 13393

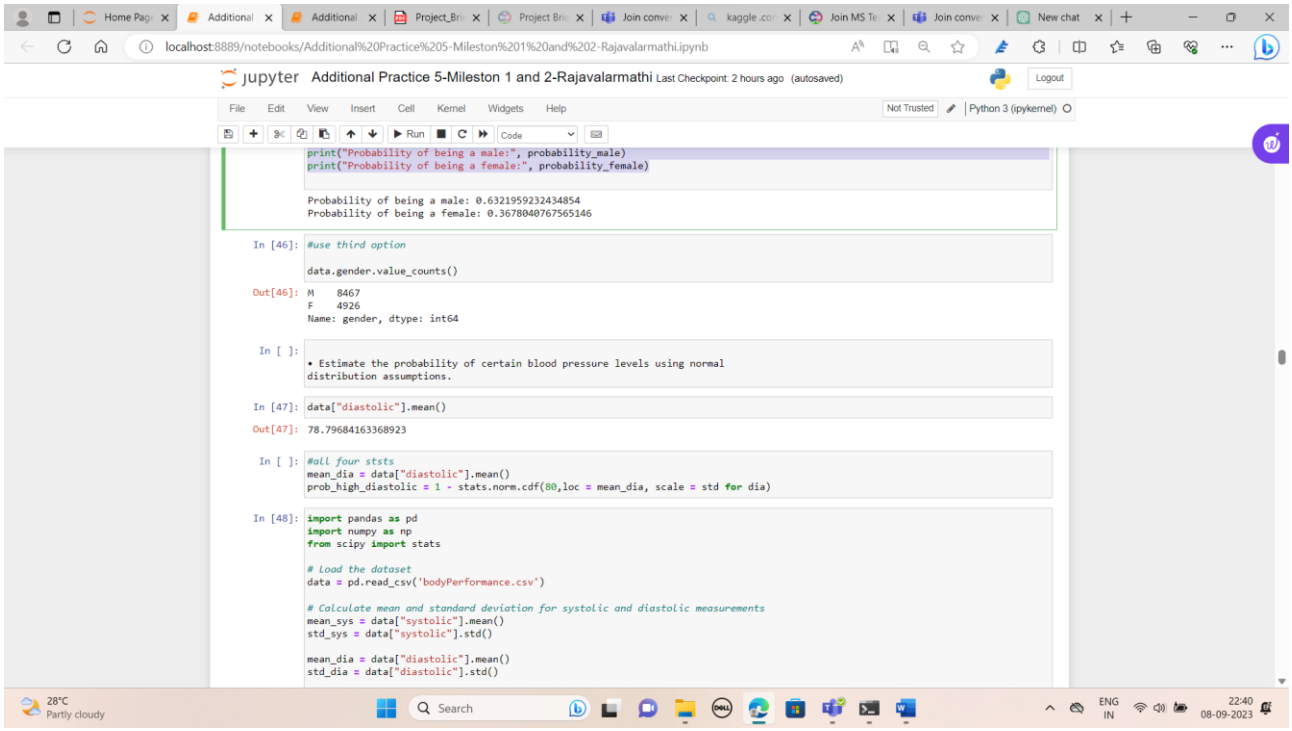
 # Number of males
male_count = 8467

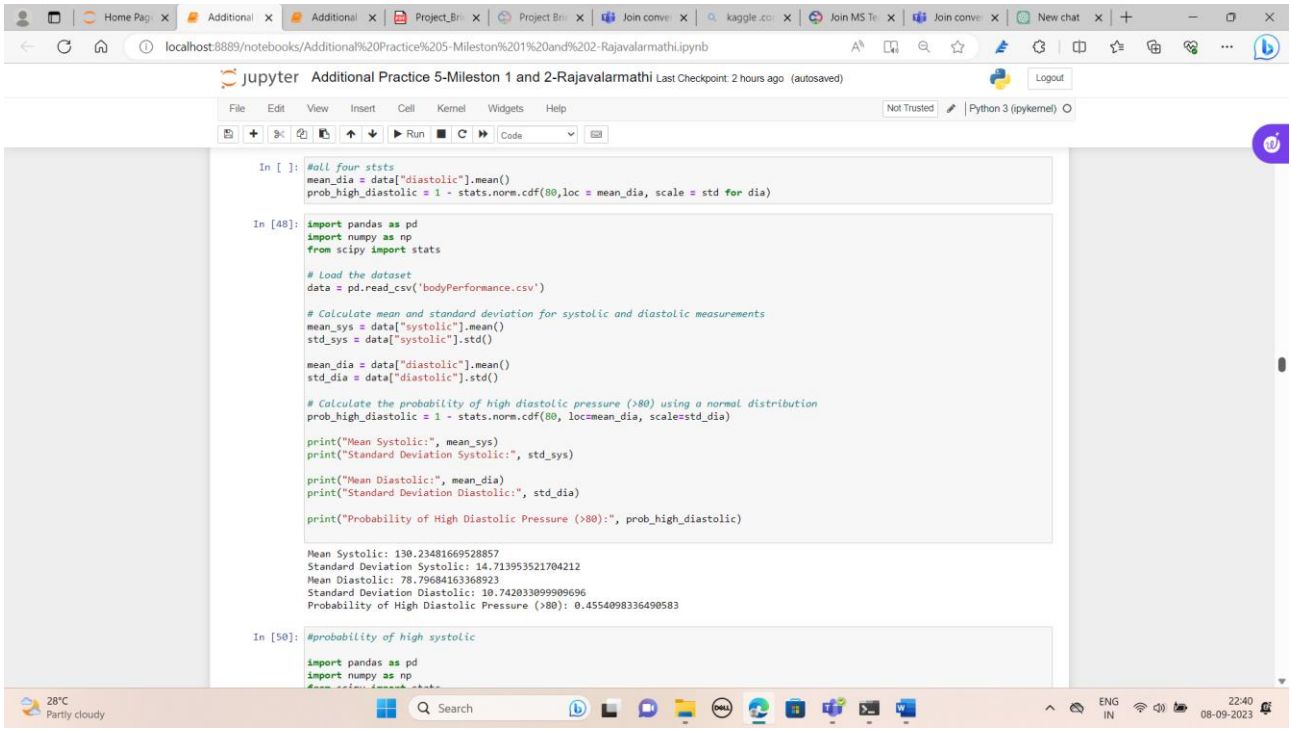
 # Number of females
female_count = 4926

 # Calculate the probability of being a male
probability_male = male_count / total_count

 # Calculate the probability of being a female
probability_female = female_count / total_count

print("Probability of being a male:", probability_male)
print("Probability of being a female:", probability_female)

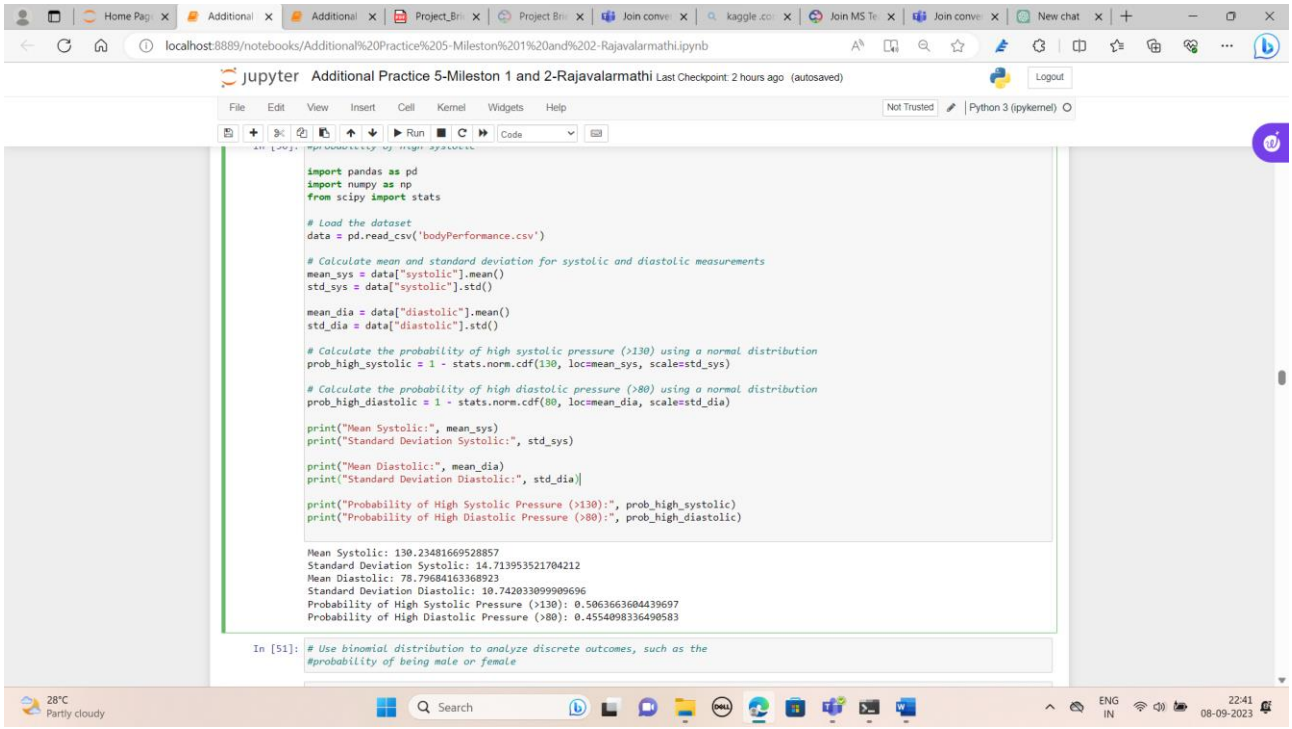


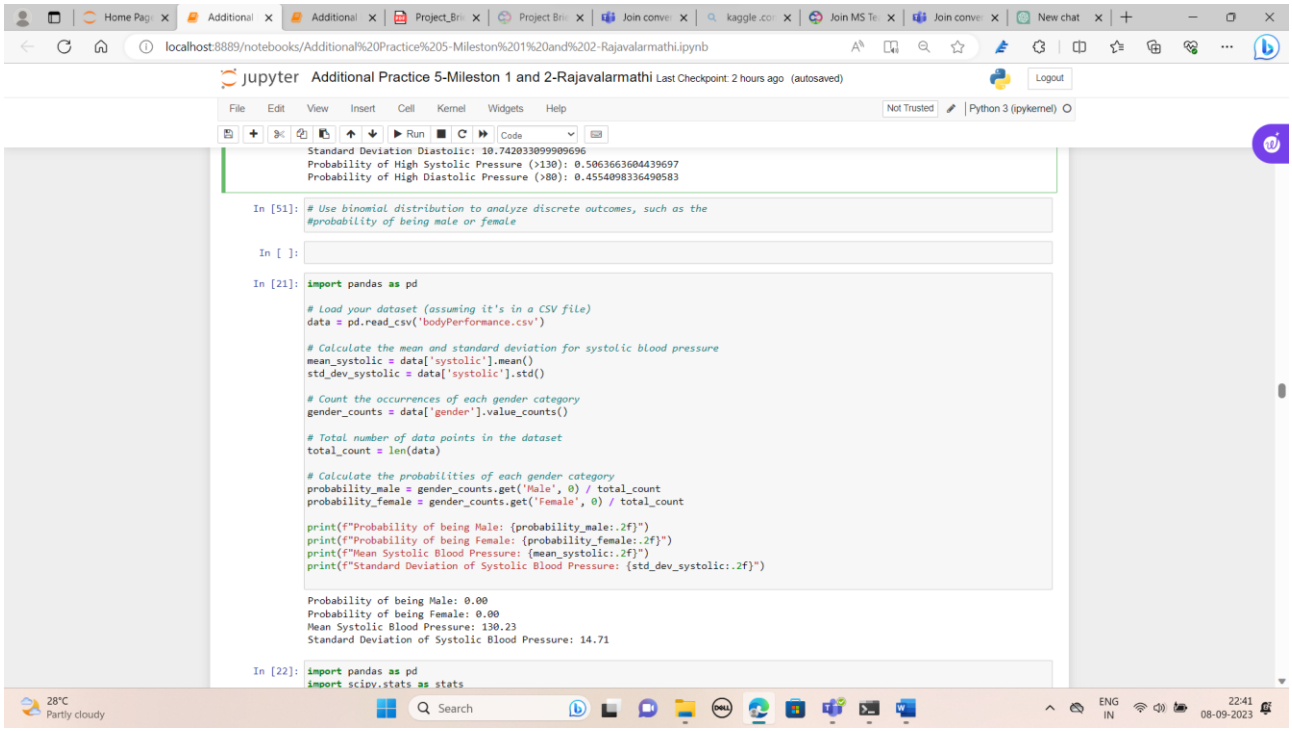


• Estimate the probability of certain blood pressure levels using normal distribution assumptions.

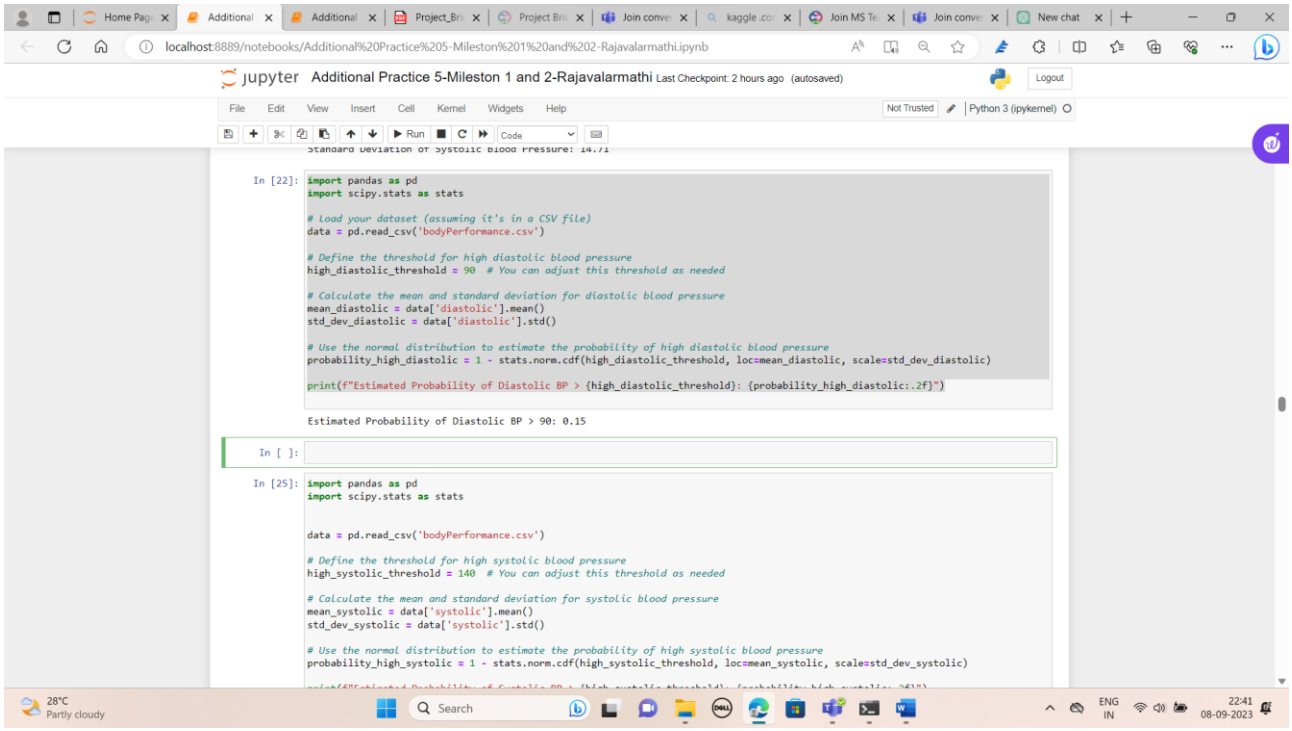
```
import pandas as pd
import numpy as np
from scipy import stats

# Load the dataset
data = pd.read_csv('bodyPerformance.csv')

# Calculate mean and standard deviation for systolic and diastolic measurements
```

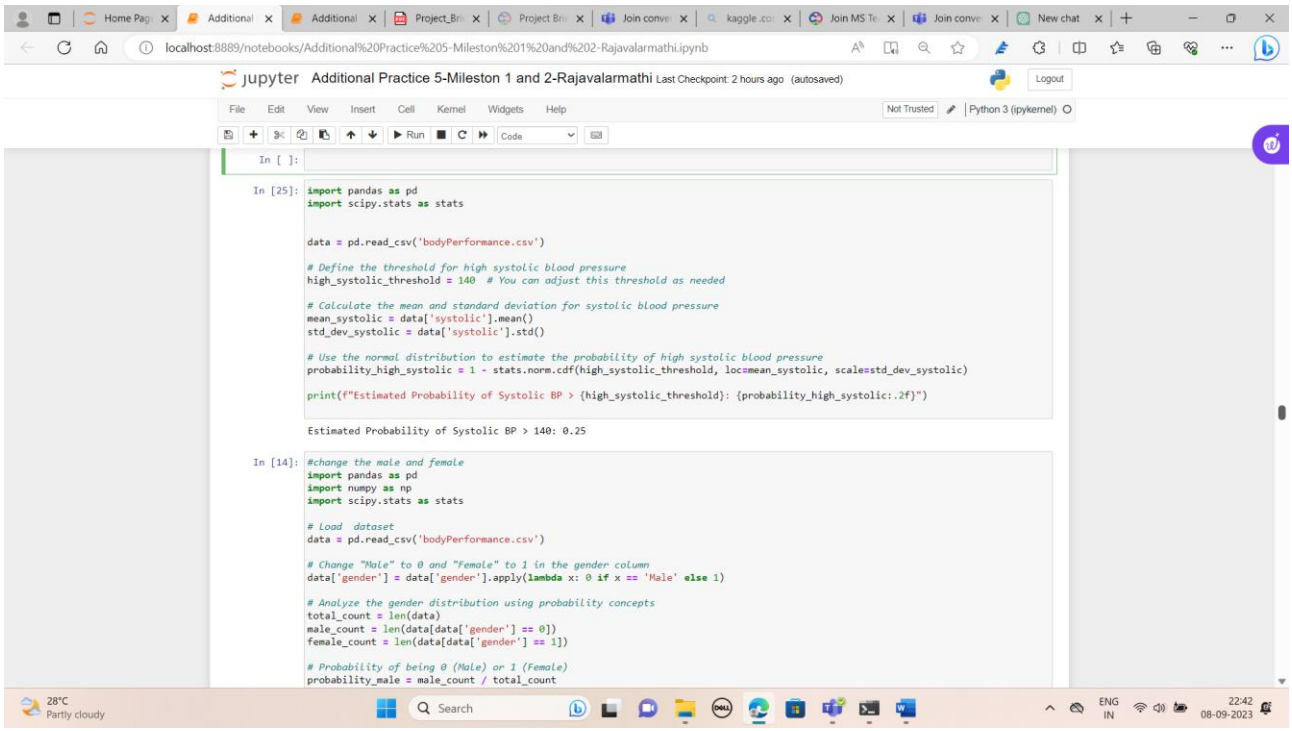
```
mean_sys = data["systolic"].mean()
std_sys = data["systolic"].std()

mean_dia = data["diastolic"].mean()
std_dia = data["diastolic"].std()

# Calculate the probability of high systolic pressure (>130) using a normal distribution
prob_high_systolic = 1 - stats.norm.cdf(130, loc=mean_sys, scale=std_sys)

# Calculate the probability of high diastolic pressure (>80) using a normal distribution
prob_high_diastolic = 1 - stats.norm.cdf(80, loc=mean_dia, scale=std_dia)

print("Mean Systolic:", mean_sys)
print("Standard Deviation Systolic:", std_sys)

print("Mean Diastolic:", mean_dia)
print("Standard Deviation Diastolic:", std_dia)

print("Probability of High Systolic Pressure (>130):", prob_high_systolic)
print("Probability of High Diastolic Pressure (>80):", prob_high_diastolic)
```





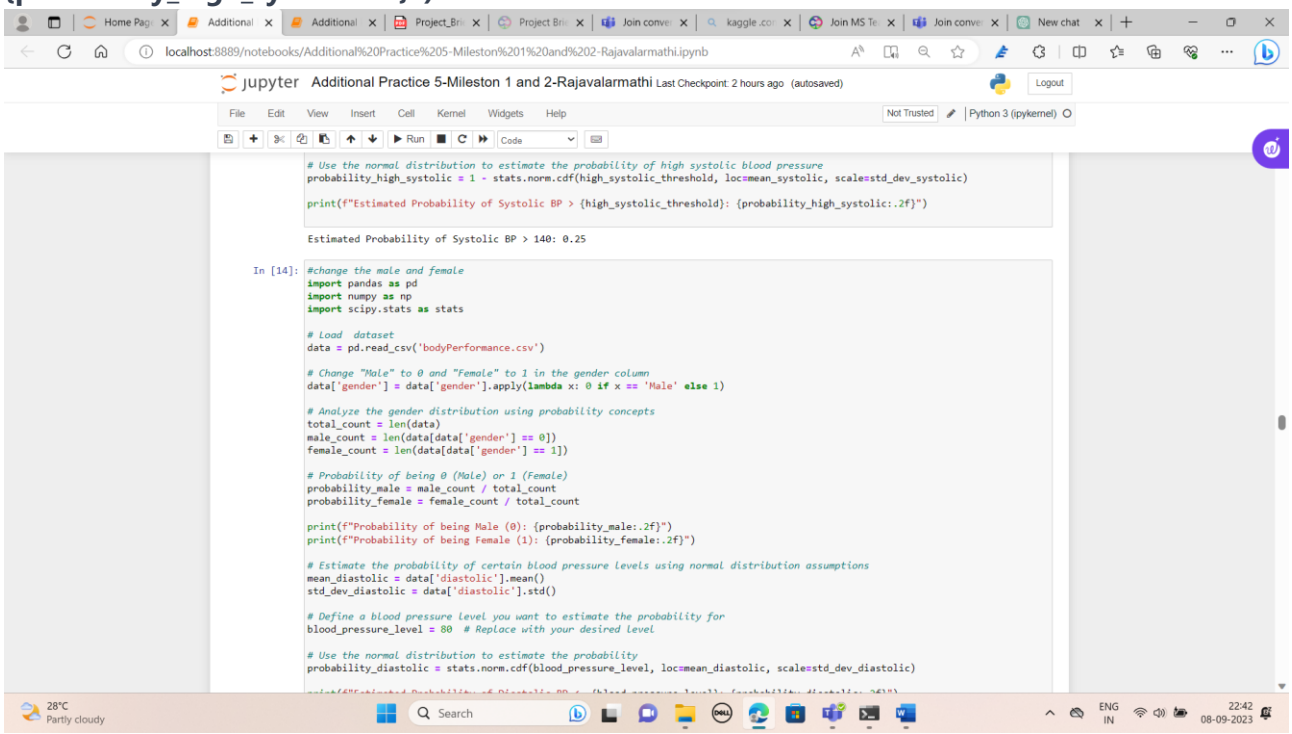

```
import pandas as pd
import scipy.stats as stats

# Load your dataset (assuming it's in a CSV file)
data = pd.read_csv('bodyPerformance.csv')

# Define the threshold for high diastolic blood pressure
high_diastolic_threshold = 90  # You can adjust this threshold as needed

# Calculate the mean and standard deviation for diastolic blood pressure
mean_diastolic = data['diastolic'].mean()
std_dev_diastolic = data['diastolic'].std()

# Use the normal distribution to estimate the probability of high diastolic blood pressure
probability_high_diastolic = 1 - stats.norm.cdf(high_diastolic_threshold, loc=mean_diastolic,
scale=std_dev_diastolic)

print(f"Estimated Probability of Diastolic BP > {high_diastolic_threshold}:
{probability_high_diastolic:.2f}")
```






```
import pandas as pd
import scipy.stats as stats


data = pd.read_csv('bodyPerformance.csv')

# Define the threshold for high systolic blood pressure
high_systolic_threshold = 140  # You can adjust this threshold as needed

# Calculate the mean and standard deviation for systolic blood pressure
```

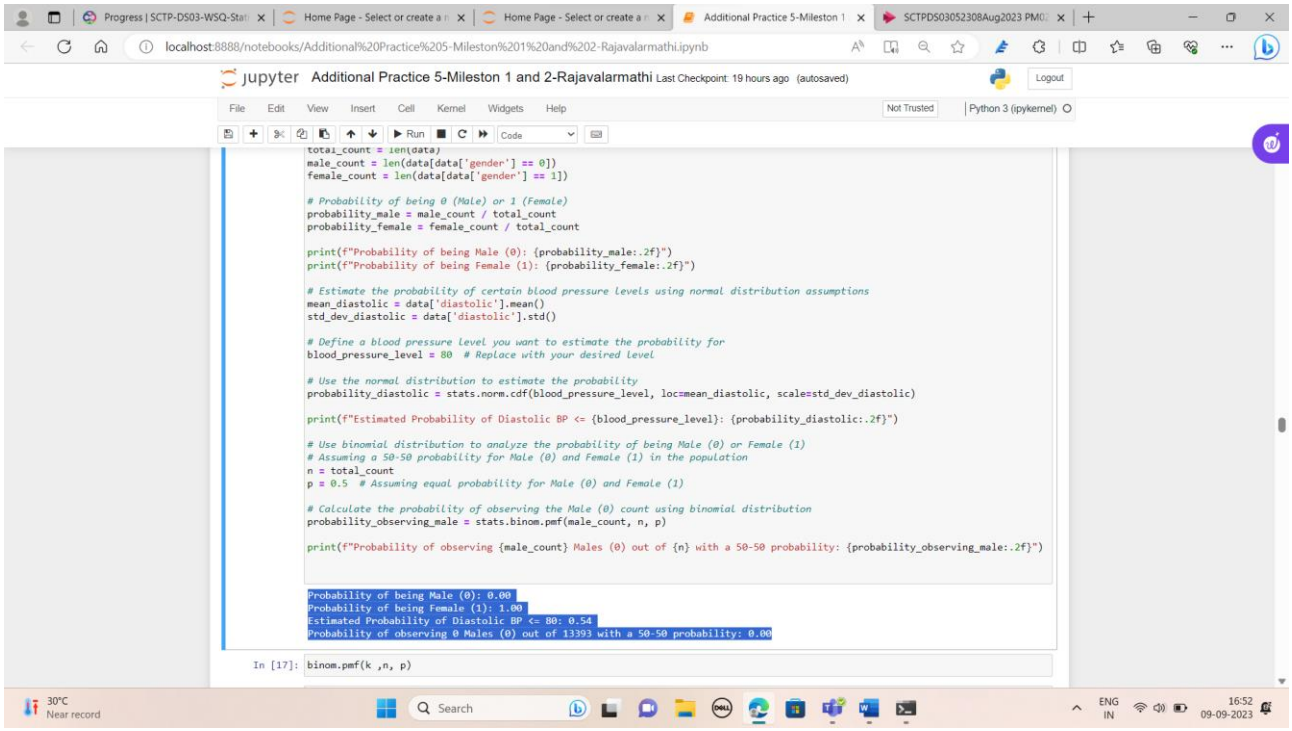
```
mean_systolic = data['systolic'].mean()
std_dev_systolic = data['systolic'].std()

# Use the normal distribution to estimate the probability of high systolic blood pressure
probability_high_systolic = 1 - stats.norm.cdf(high_systolic_threshold, loc=mean_systolic,
scale=std_dev_systolic)

print(f"Estimated Probability of Systolic BP > {high_systolic_threshold}:
{probability_high_systolic:.2f}")
```



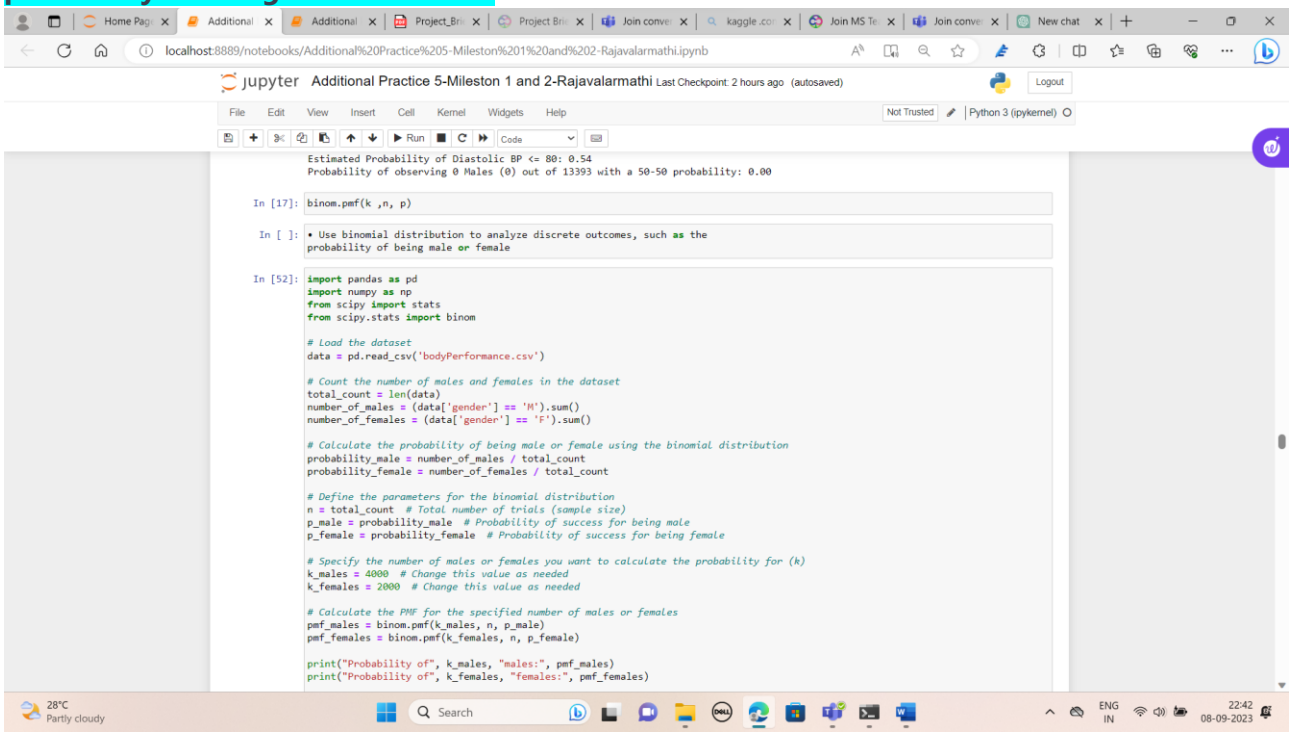

```
#change the male and female


import pandas as pd
import numpy as np
import scipy.stats as stats

# Load  dataset
data = pd.read_csv('bodyPerformance.csv')

# Change "Male" to 0 and "Female" to 1 in the gender column
data['gender'] = data['gender'].apply(lambda x: 0 if x == 'Male' else 1)

# Analyze the gender distribution using probability concepts
total_count = len(data)
male_count = len(data[data['gender'] == 0])
female_count = len(data[data['gender'] == 1])

# Probability of being 0 (Male) or 1 (Female)
```

```
probability_male = male_count / total_count
probability_female = female_count / total_count

print(f"Probability of being Male (0): {probability_male:.2f}")
print(f"Probability of being Female (1): {probability_female:.2f}")

# Estimate the probability of certain blood pressure levels using normal distribution
assumptions
mean_diastolic = data['diastolic'].mean()
std_dev_diastolic = data['diastolic'].std()

# Define a blood pressure level you want to estimate the probability for
blood_pressure_level = 80  # Replace with your desired level

# Use the normal distribution to estimate the probability
probability_diastolic = stats.norm.cdf(blood_pressure_level, loc=mean_diastolic,
scale=std_dev_diastolic)

print(f"Estimated Probability of Diastolic BP <= {blood_pressure_level}:
{probability_diastolic:.2f}")

# Use binomial distribution to analyze the probability of being Male (0) or Female (1)
# Assuming a 50-50 probability for Male (0) and Female (1) in the population
n = total_count
p = 0.5  # Assuming equal probability for Male (0) and Female (1)

# Calculate the probability of observing the Male (0) count using binomial distribution
probability_observing_male = stats.binom.pmf(male_count, n, p)

print(f"Probability of observing {male_count} Males (0) out of {n} with a 50-50 probability:
{probability_observing_male:.2f}")
```

output:
```
Probability of being Male (0): 0.00
Probability of being Female (1): 1.00
Estimated Probability of Diastolic BP <= 80: 0.54
Probability of observing 0 Males (0) out of 13393 with a 50-50 probability:
0.00
```



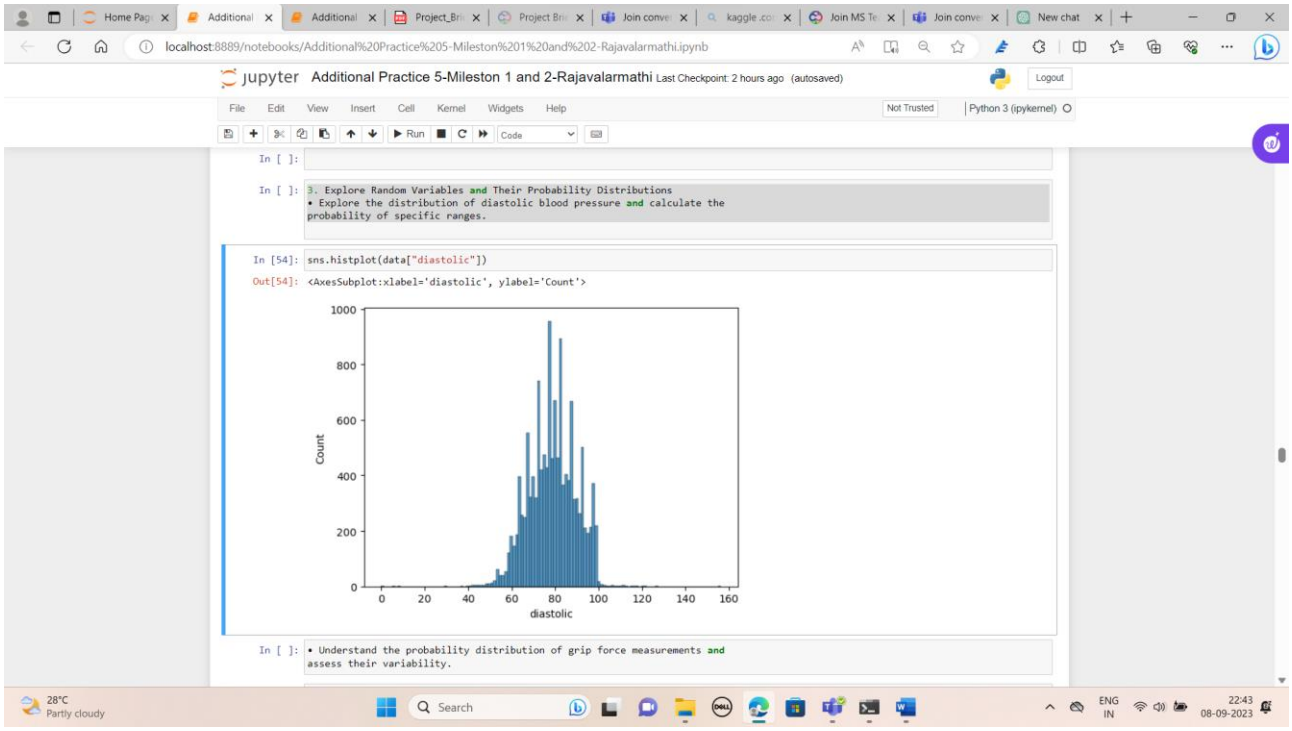
Use binomial distribution to analyze discrete outcomes, such as the probability of being male or female

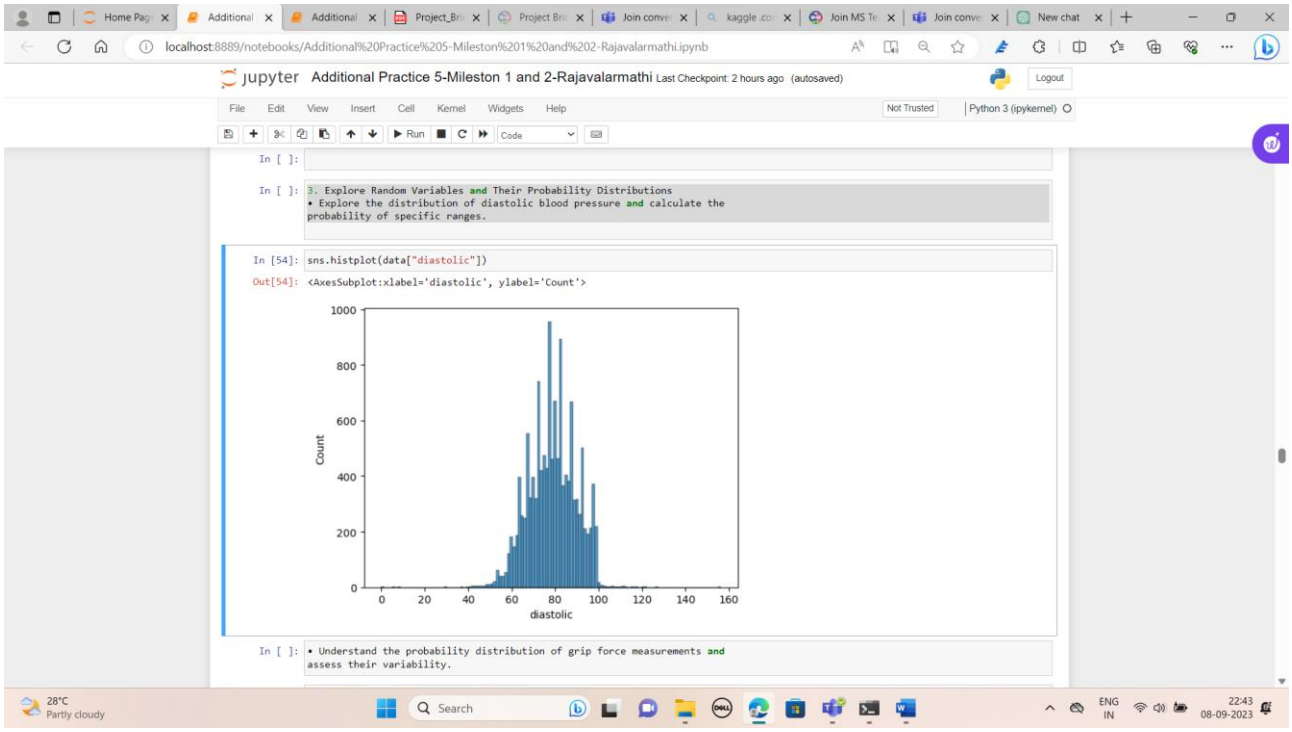


Milestone 3: 3. Explore Random Variables and Their Probability Distributions

• Explore the distribution of diastolic blood pressure and calculate the probability of specific ranges.

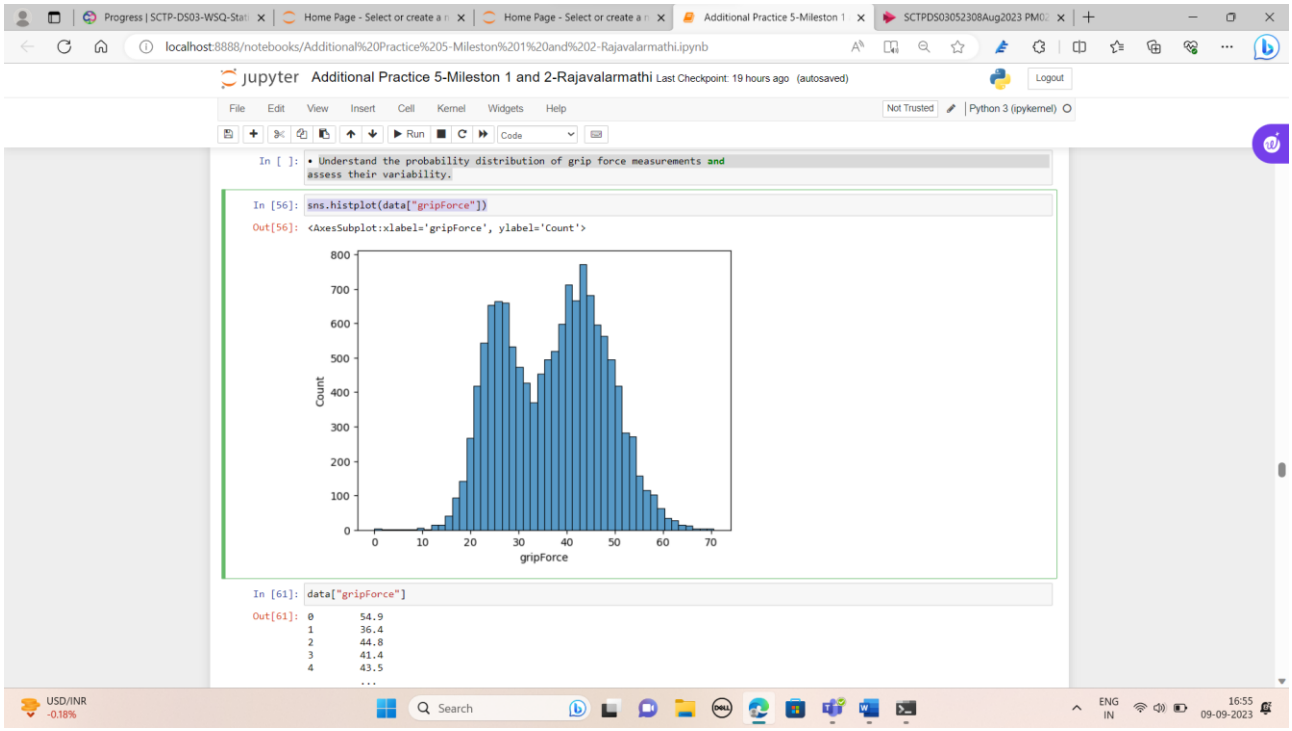
sns.histplot(data["diastolic"])

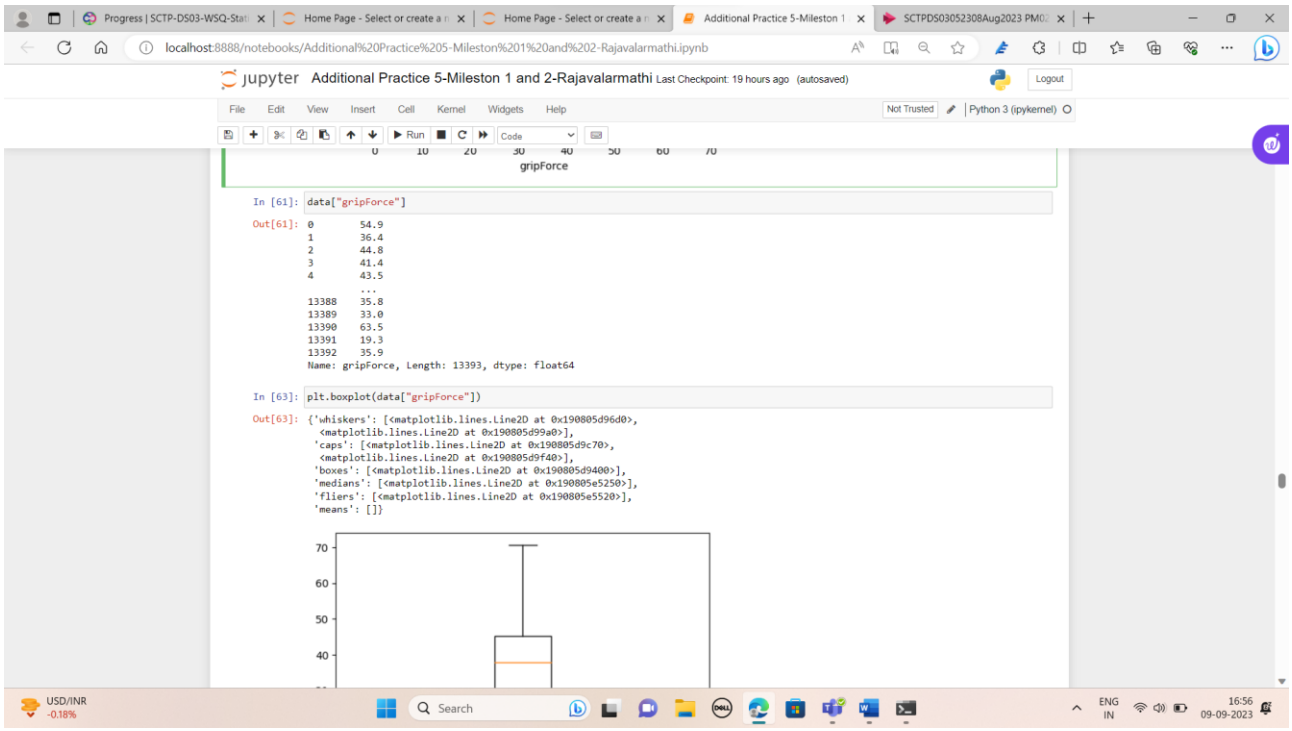




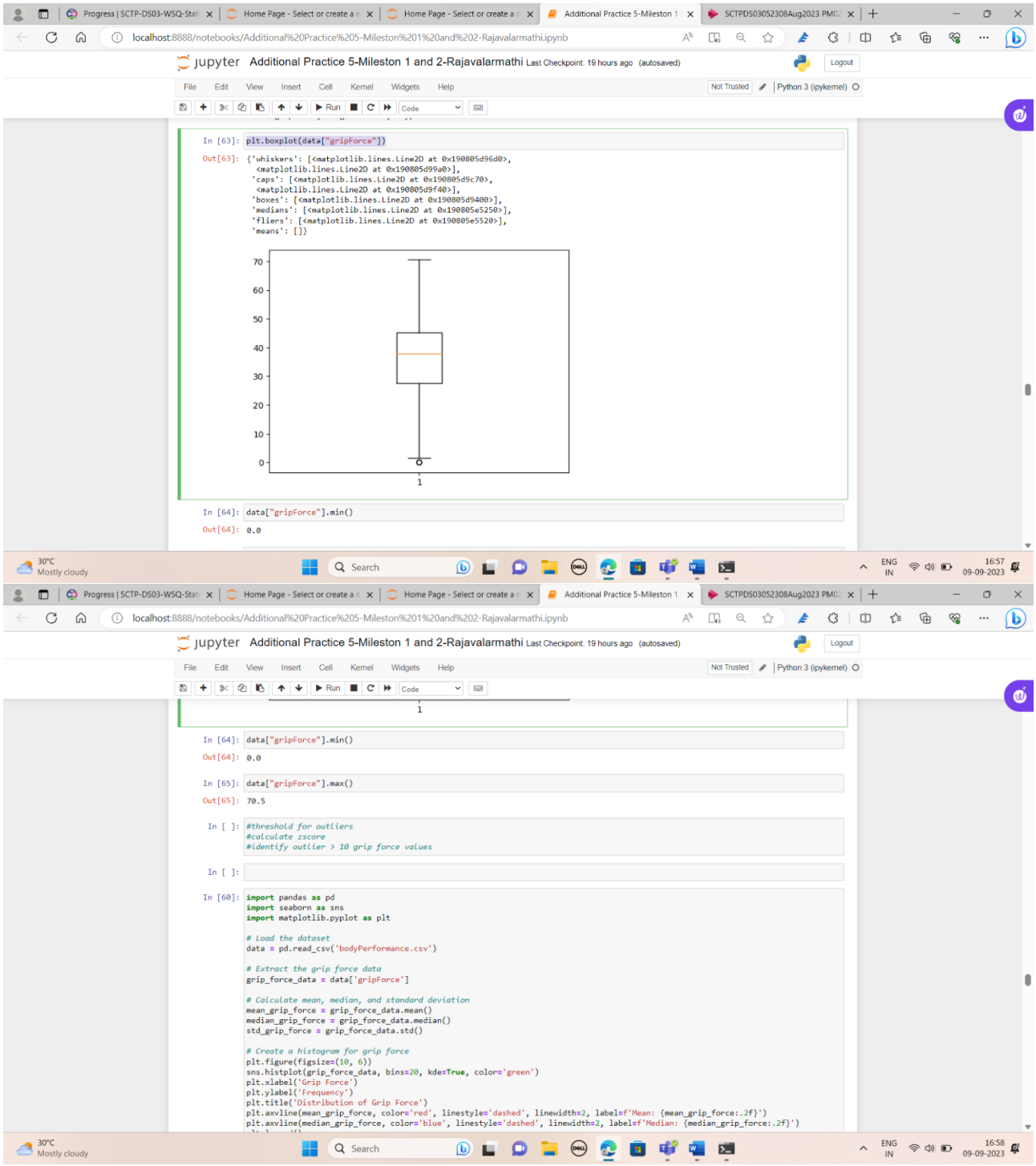
Understand the probability distribution of grip force measurements and assess their variability.

sns.histplot(data["gripForce"])





plt.boxplot(data["gripForce"])



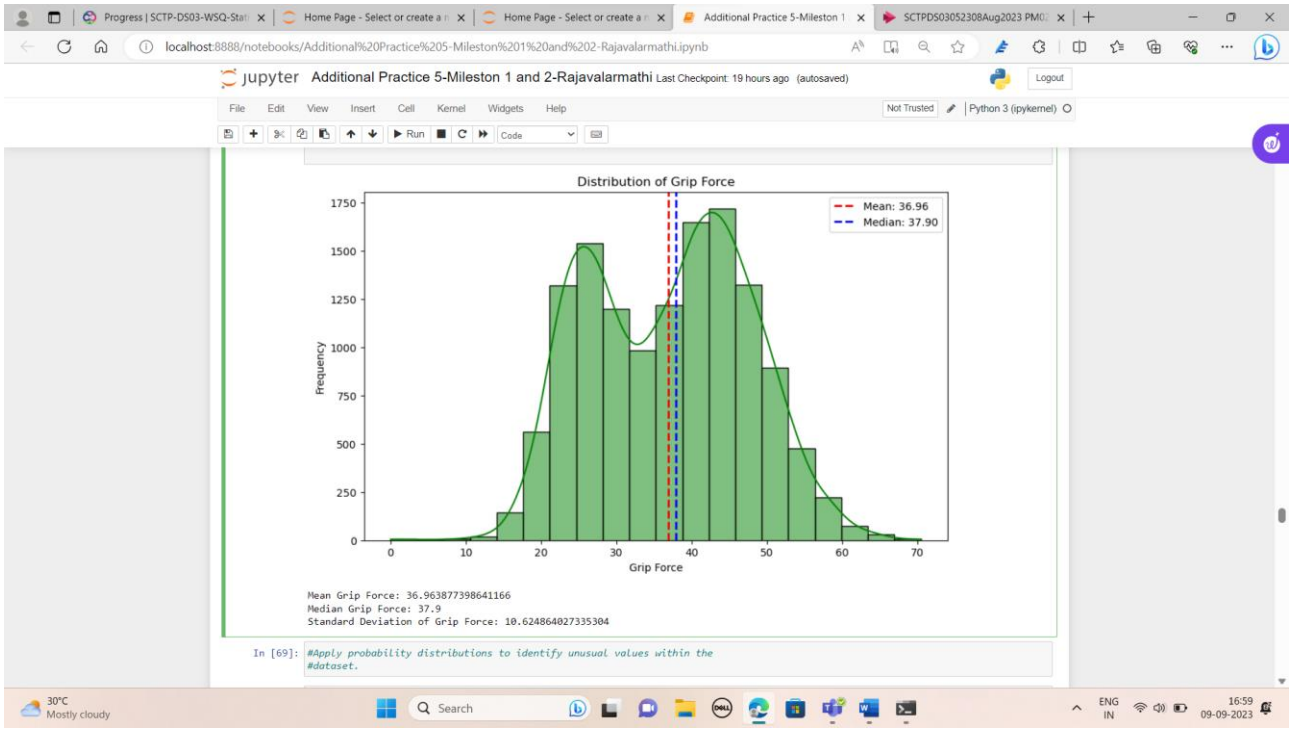
#threshold for outliers

#calculate zscore

#identify outlier > 10 grip force values

import pandas as pd

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('bodyPerformance.csv')

# Extract the grip force data
grip_force_data = data['gripForce']

# Calculate mean, median, and standard deviation
mean_grip_force = grip_force_data.mean()
median_grip_force = grip_force_data.median()
std_grip_force = grip_force_data.std()

# Create a histogram for grip force
plt.figure(figsize=(10, 6))
sns.histplot(grip_force_data, bins=20, kde=True, color='green')
plt.xlabel('Grip Force')
plt.ylabel('Frequency')
plt.title('Distribution of Grip Force')
plt.axvline(mean_grip_force, color='red', linestyle='dashed', linewidth=2, label=f'Mean: {mean_grip_force:.2f}')
plt.axvline(median_grip_force, color='blue', linestyle='dashed', linewidth=2, label=f'Median: {median_grip_force:.2f}')
plt.legend()
plt.show()

print("Mean Grip Force:", mean_grip_force)
print("Median Grip Force:", median_grip_force)
print("Standard Deviation of Grip Force:", std_grip_force)
```



#Apply probability distributions to identify unusual values within the

#dataset.

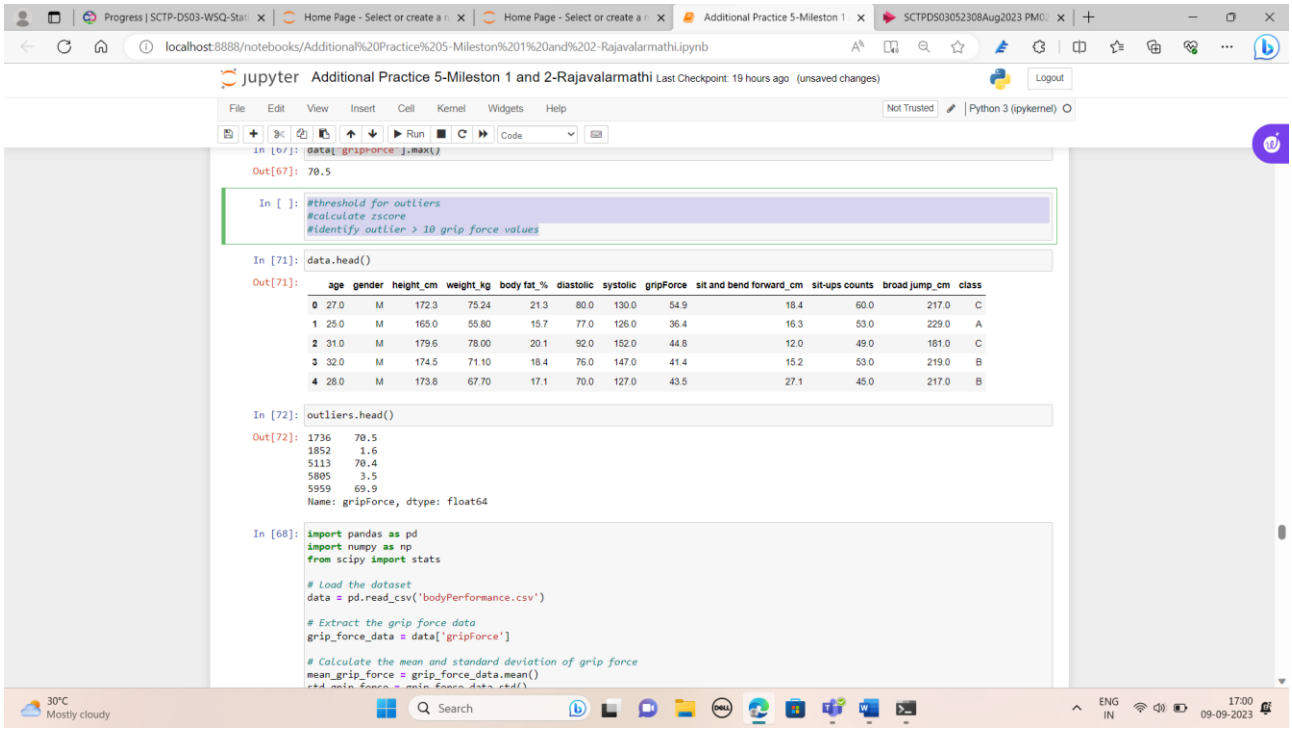
data["gripForce"].min()

data["gripForce"].max()

#threshold for outliers

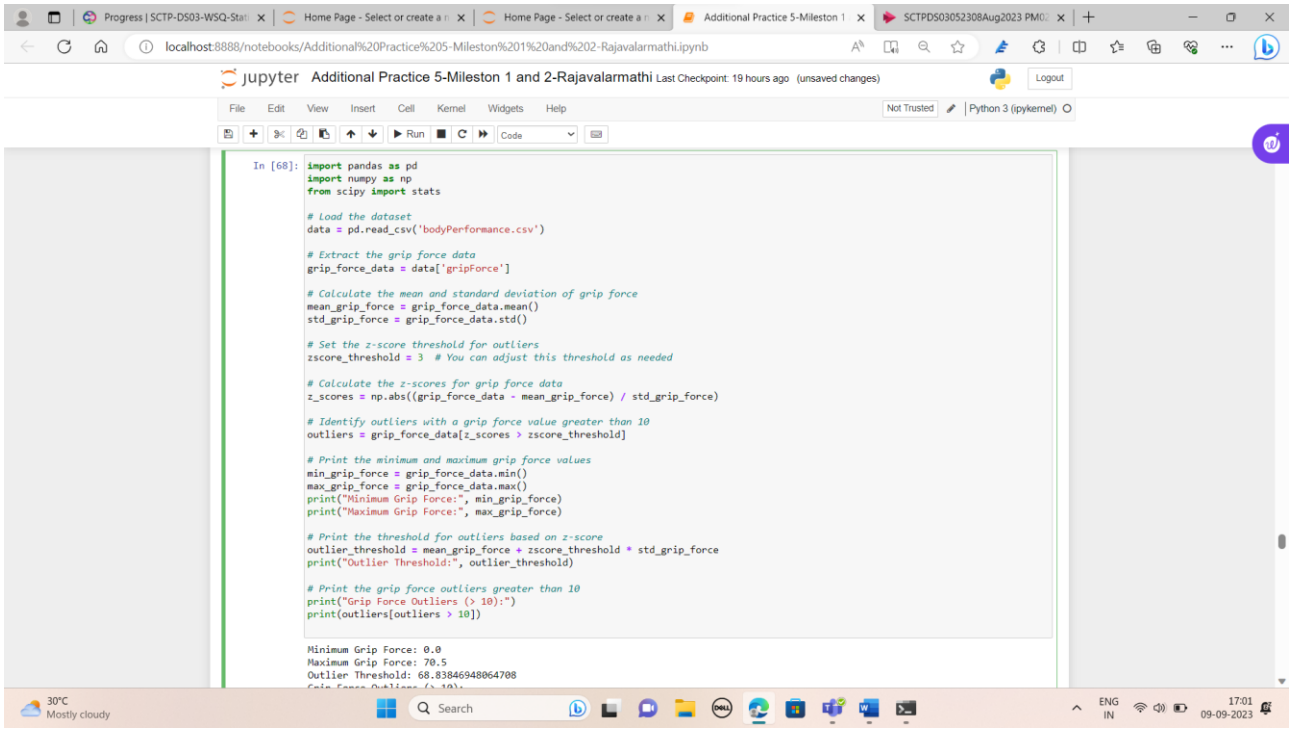
#calculate zscore

#identify outlier > 10 grip force values




```
import pandas as pd
import numpy as np
from scipy import stats

# Load the dataset
data = pd.read_csv('bodyPerformance.csv')

# Extract the grip force data
grip_force_data = data['gripForce']

# Calculate the mean and standard deviation of grip force
mean_grip_force = grip_force_data.mean()
std_grip_force = grip_force_data.std()

# Set the z-score threshold for outliers
zscore_threshold = 3  # You can adjust this threshold as needed

# Calculate the z-scores for grip force data
z_scores = np.abs((grip_force_data - mean_grip_force) / std_grip_force)
```

```
# Identify outliers with a grip force value greater than 10
outliers = grip_force_data[z_scores > zscore_threshold]

# Print the minimum and maximum grip force values
min_grip_force = grip_force_data.min()
max_grip_force = grip_force_data.max()
print("Minimum Grip Force:", min_grip_force)
print("Maximum Grip Force:", max_grip_force)

# Print the threshold for outliers based on z-score
outlier_threshold = mean_grip_force + zscore_threshold * std_grip_force
print("Outlier Threshold:", outlier_threshold)

# Print the grip force outliers greater than 10
print("Grip Force Outliers (> 10):")
print(outliers[outliers > 10])
```



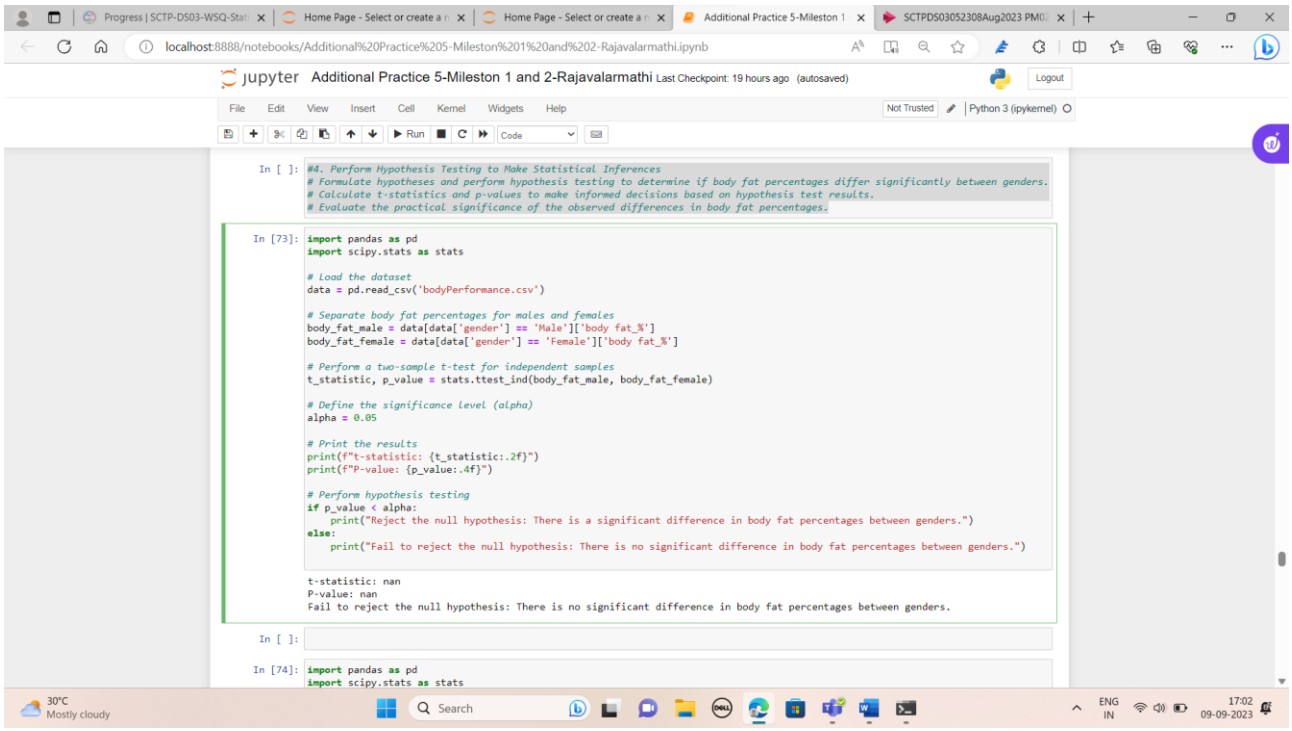
Milestone:4: #4. Perform Hypothesis Testing to Make Statistical Inferences

Formulate hypotheses and perform hypothesis testing to determine if body fat percentages differ significantly between genders.

Calculate t-statistics and p-values to make informed decisions based on hypothesis test results.

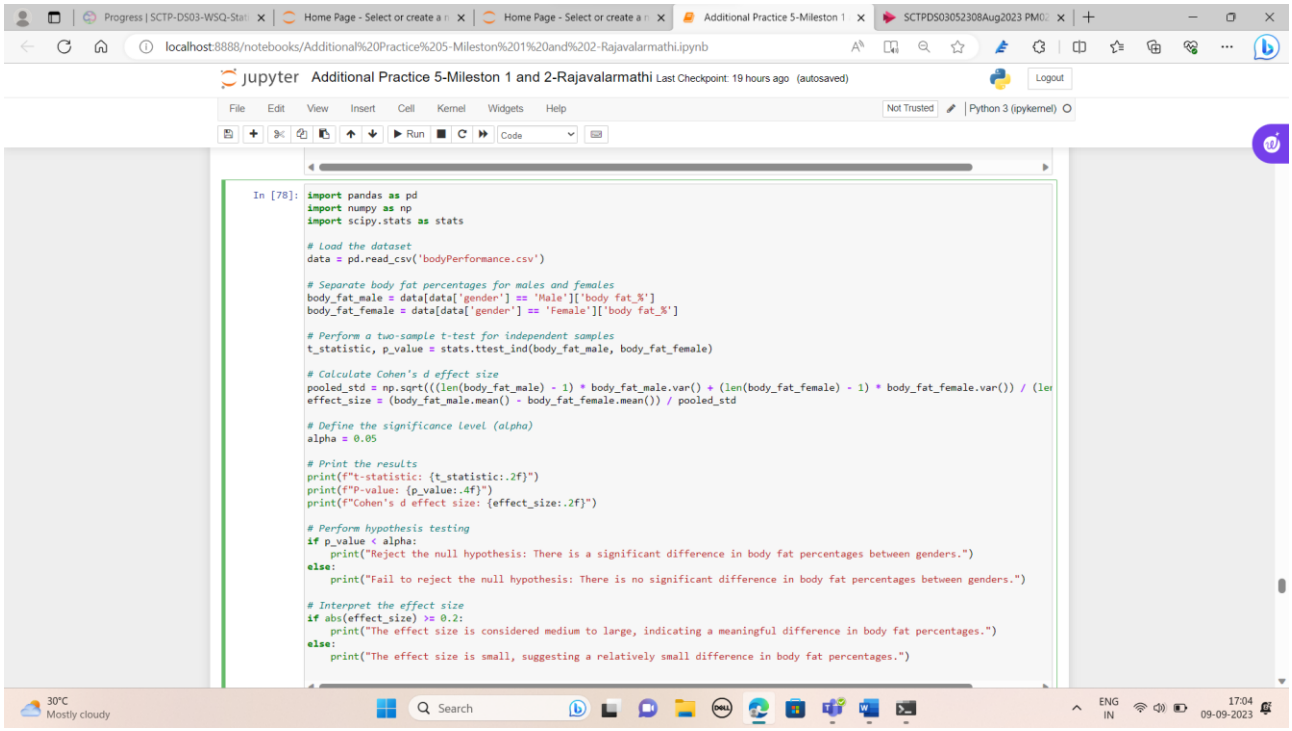
Evaluate the practical significance of the observed differences in body fat percentages.

```
import pandas as pd
import scipy.stats as stats

# Load the dataset
data = pd.read_csv('bodyPerformance.csv')

# Separate body fat percentages for males and females
body_fat_male = data[data['gender'] == 'Male']['body fat_%']
body_fat_female = data[data['gender'] == 'Female']['body fat_%']

# Perform a two-sample t-test for independent samples
t_statistic, p_value = stats.ttest_ind(body_fat_male, body_fat_female)

# Define the significance level (alpha)
alpha = 0.05

# Print the results
print(f"t-statistic: {t_statistic:.2f}")
print(f"P-value: {p_value:.4f}")

# Perform hypothesis testing
if p_value < alpha:
    print("Reject the null hypothesis: There is a significant difference in body fat percentages between genders.")
else:
    print("Fail to reject the null hypothesis: There is no significant difference in body fat percentages between genders.")
```




```
import numpy as np

# Calculate Cohen's d effect size
pooled_std = np.sqrt(((len(body_fat_male) - 1) * body_fat_male.var() + (len(body_fat_female) - 1) * body_fat_female.var()) / (len(data) - 2))
effect_size = (body_fat_male.mean() - body_fat_female.mean()) / pooled_std

import pandas as pd
import numpy as np
import scipy.stats as stats

# Load the dataset
data = pd.read_csv('bodyPerformance.csv')

# Separate body fat percentages for males and females
body_fat_male = data[data['gender'] == 'Male']['body fat_%']
body_fat_female = data[data['gender'] == 'Female']['body fat_%']
```

```
# Perform a two-sample t-test for independent samples
t_statistic, p_value = stats.ttest_ind(body_fat_male, body_fat_female)

# Calculate Cohen's d effect size
pooled_std = np.sqrt(((len(body_fat_male) - 1) * body_fat_male.var() + (len(body_fat_female) - 1) * body_fat_female.var()) / (len(data) - 2))
effect_size = (body_fat_male.mean() - body_fat_female.mean()) / pooled_std

# Define the significance level (alpha)
alpha = 0.05

# Print the results
print(f"t-statistic: {t_statistic:.2f}")
print(f"P-value: {p_value:.4f}")
print(f"Cohen's d effect size: {effect_size:.2f}")

# Perform hypothesis testing
if p_value < alpha:
    print("Reject the null hypothesis: There is a significant difference in body fat percentages between genders.")
else:
    print("Fail to reject the null hypothesis: There is no significant difference in body fat percentages between genders.")

# Interpret the effect size
if abs(effect_size) >= 0.2:
    print("The effect size is considered medium to large, indicating a meaningful difference in body fat percentages.")
else:
    print("The effect size is small, suggesting a relatively small difference in body fat percentages.")
```



Output: `t-statistic: nan`
`P-value: nan`
`Cohen's d effect size: nan`
`Fail to reject the null hypothesis: There is no significant difference in bo dy fat percentages between genders.`
`The effect size is small, suggesting a relatively small difference in body f at percentages.`

Milestone 5: 5. Estimate Confidence Intervals and Assess Sampling Variability in Data

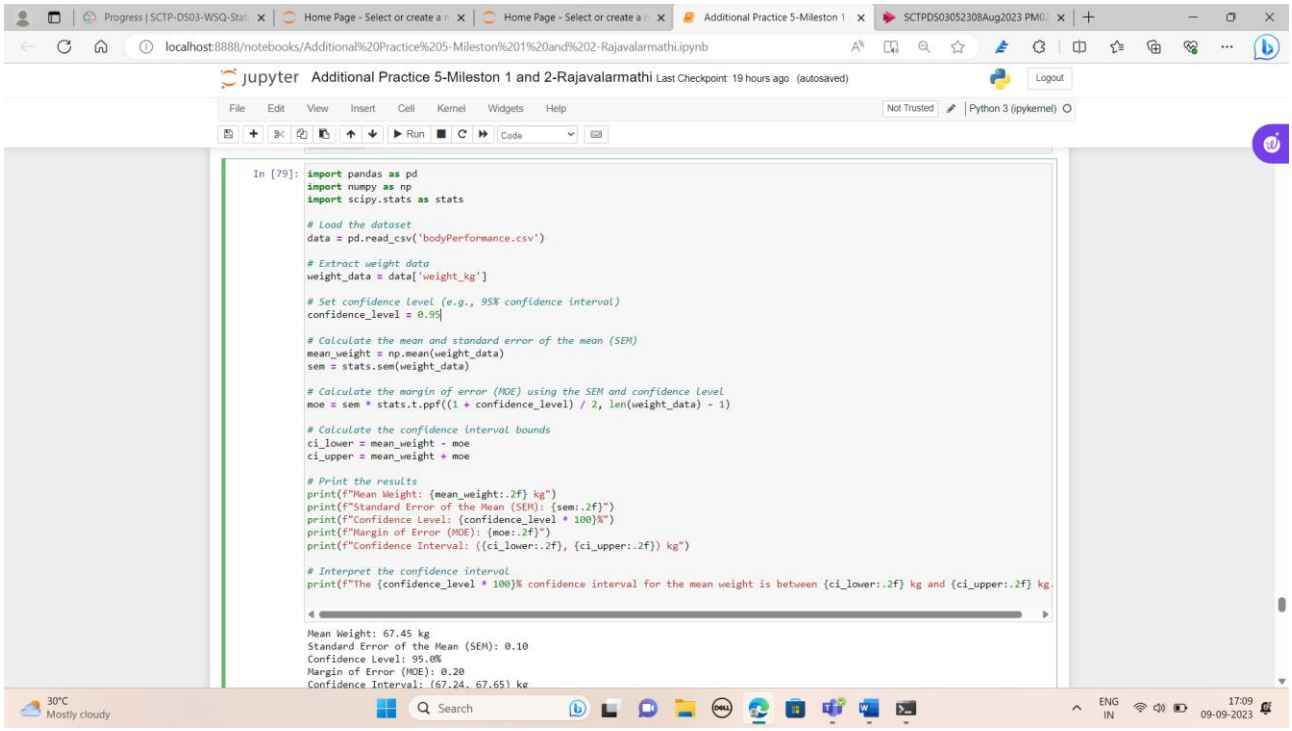
• Estimate confidence intervals for the mean weight of individuals and interpret their significance.

• Assess how varying sample sizes affect the width of confidence intervals.

• Understand the impact of confidence levels on the precision of parameter estimates.

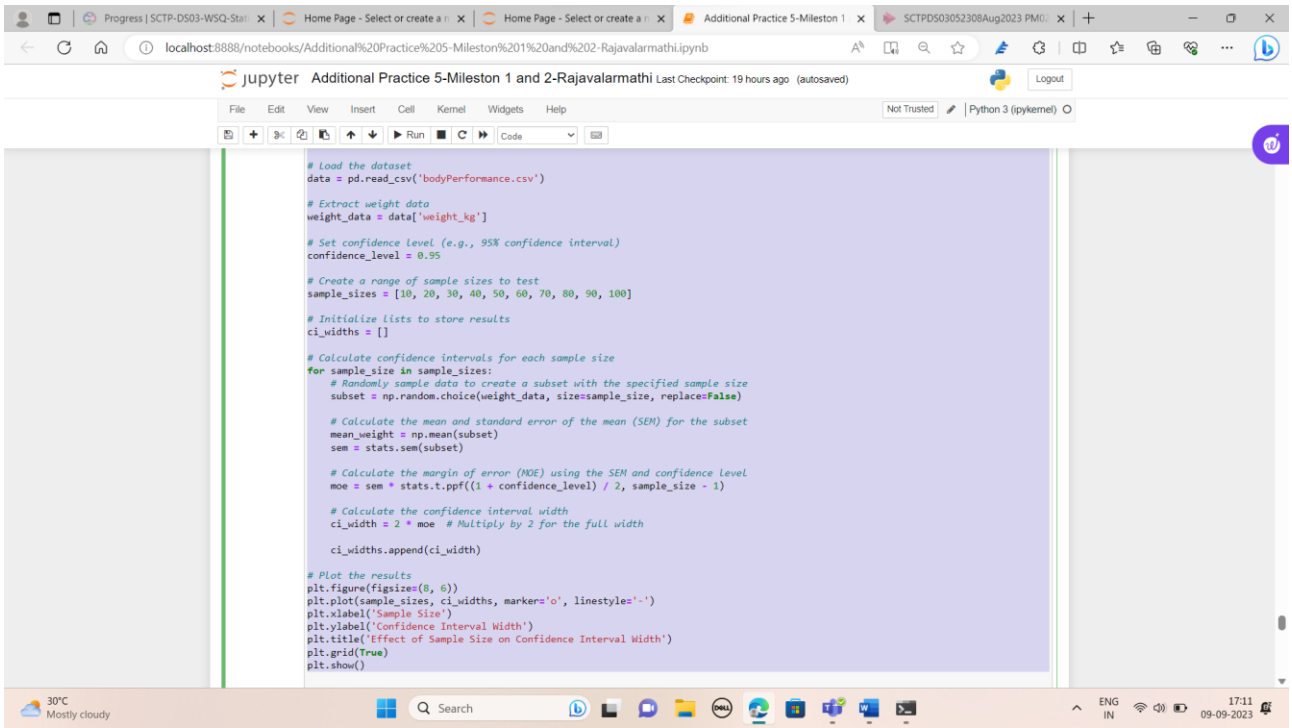
```
import pandas as pd
import numpy as np
import scipy.stats as stats

# Load the dataset
data = pd.read_csv('bodyPerformance.csv')
```

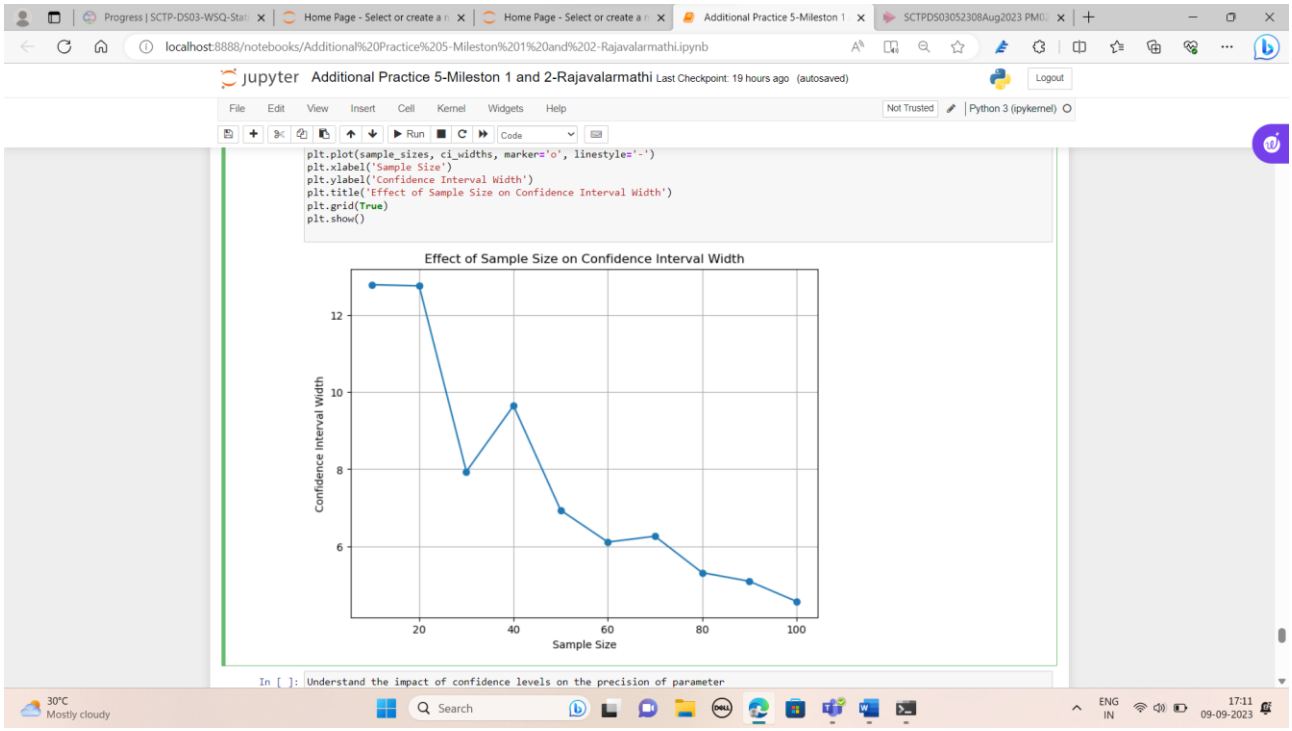
```
# Extract weight data
weight_data = data['weight_kg']

# Set confidence level (e.g., 95% confidence interval)
confidence_level = 0.95

# Calculate the mean and standard error of the mean (SEM)
mean_weight = np.mean(weight_data)
sem = stats.sem(weight_data)

# Calculate the margin of error (MOE) using the SEM and confidence level
moe = sem * stats.t.ppf((1 + confidence_level) / 2, len(weight_data) - 1)

# Calculate the confidence interval bounds
ci_lower = mean_weight - moe
ci_upper = mean_weight + moe

# Print the results
print(f"Mean Weight: {mean_weight:.2f} kg")
print(f"Standard Error of the Mean (SEM): {sem:.2f}")
print(f"Confidence Level: {confidence_level * 100}%")
print(f"Margin of Error (MOE): {moe:.2f}")
print(f"Confidence Interval: ({ci_lower:.2f}, {ci_upper:.2f}) kg")

# Interpret the confidence interval
print(f"The {confidence_level * 100}% confidence interval for the mean weight is between
{ci_lower:.2f} kg and {ci_upper:.2f} kg.")
```



Output: `Mean Weight: 67.45 kg`
`Standard Error of the Mean (SEM): 0.10`
`Confidence Level: 95.0%`
`Margin of Error (MOE): 0.20`
`Confidence Interval: (67.24, 67.65) kg`
`The 95.0% confidence interval for the mean weight is between 67.24 kg and 67.65 kg.`

Assess how varying sample sizes affect the width of confidence intervals.

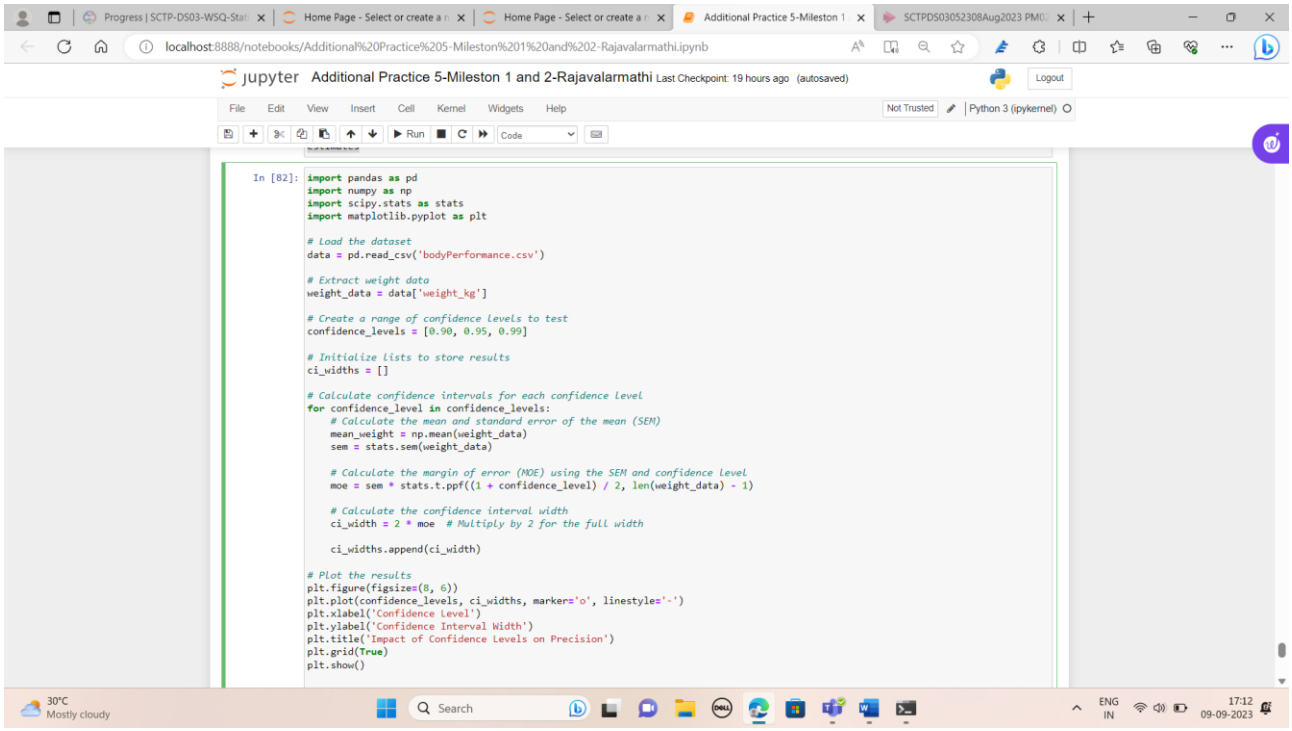


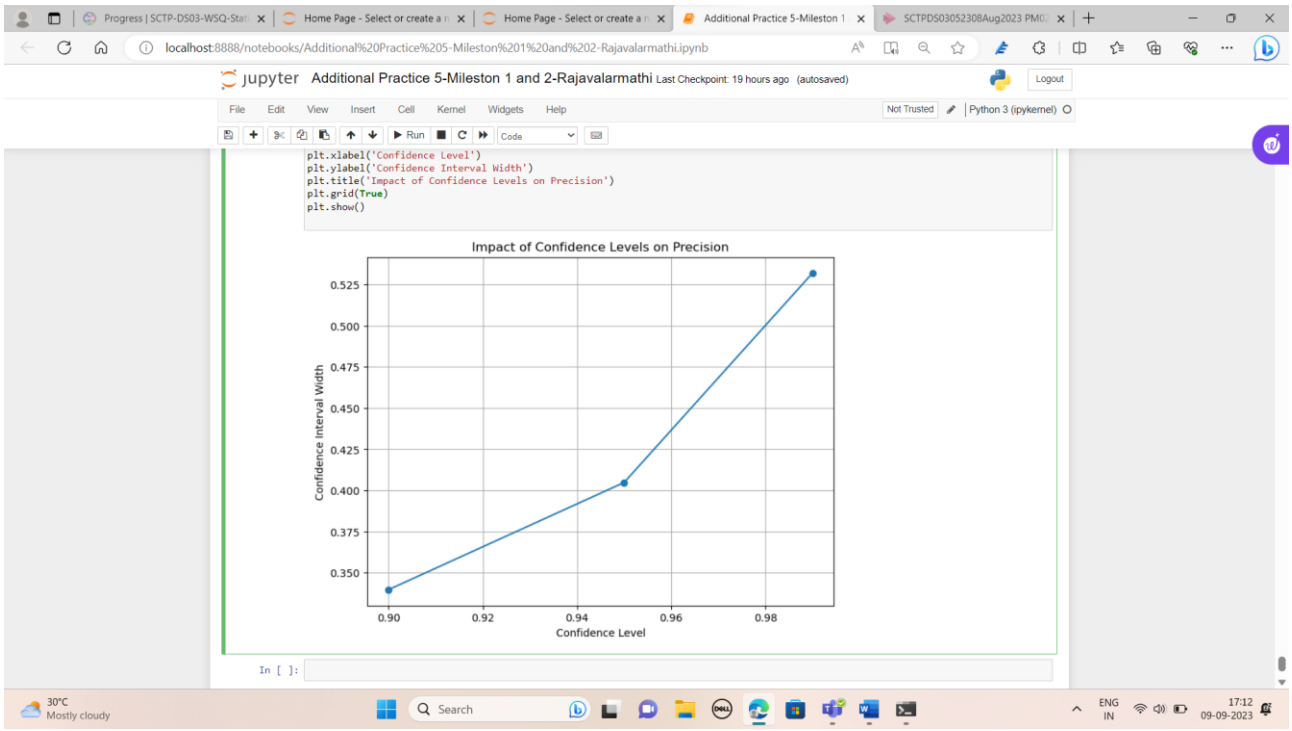


Understand the impact of confidence levels on the precision of parameter Estimates

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

# Load the dataset
data = pd.read_csv('bodyPerformance.csv')

# Extract weight data
weight_data = data['weight_kg']

# Create a range of confidence levels to test
confidence_levels = [0.90, 0.95, 0.99]

# Initialize lists to store results
ci_widths = []
```

```
# Calculate confidence intervals for each confidence level
for confidence_level in confidence_levels:
    # Calculate the mean and standard error of the mean (SEM)
    mean_weight = np.mean(weight_data)
    sem = stats.sem(weight_data)

    # Calculate the margin of error (MOE) using the SEM and confidence level
    moe = sem * stats.t.ppf((1 + confidence_level) / 2, len(weight_data) - 1)

    # Calculate the confidence interval width
    ci_width = 2 * moe  # Multiply by 2 for the full width

    ci_widths.append(ci_width)

# Plot the results
plt.figure(figsize=(8, 6))
plt.plot(confidence_levels, ci_widths, marker='o', linestyle='-')
plt.xlabel('Confidence Level')
plt.ylabel('Confidence Interval Width')
plt.title('Impact of Confidence Levels on Precision')
plt.grid(True)
plt.show()
```


jupyter Additional Practice 5-Mileston 1 and 2-Rajavalarmathi Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [82]:
import pandas as pd
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
# Load the dataset
data = pd.read_csv('bodyPerformance.csv')
# Extract weight data
weight_data = data['weight_kg']
# Create a range of confidence levels to test
confidence_levels = [0.90, 0.95, 0.99]
# Initialize lists to store results
ci_widths = []
# Calculate confidence intervals for each confidence level
for confidence_level in confidence_levels:
# Calculate the mean and standard error of the mean (SEM)
mean_weight = np.mean(weight_data)
sem = stats.sem(weight_data)
# Calculate the margin of error (MOE) using the SEM and confidence level
moe = sem * stats.t.ppf((1 + confidence_level) / 2, len(weight_data) - 1)
# Calculate the confidence interval width
ci_width = 2 * moe # Multiply by 2 for the full width
ci_widths.append(ci_width)
# Plot the results
plt.figure(figsize=(8, 6))
plt.plot(confidence_levels, ci_widths, marker='o', linestyle='-')
plt.xlabel('Confidence Level')
plt.ylabel('Confidence Interval Width')
plt.title('Impact of Confidence Levels on Precision')
plt.grid(True)
plt.show()
30°C
Mostly cloudy
Search
ENG
IN
17:12
09-09-2023


jupyter Additional Practice 5-Mileston 1 and 2-Rajavalarmathi Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
plt.xlabel('Confidence Level')
plt.ylabel('Confidence Interval Width')
plt.title('Impact of Confidence Levels on Precision')
plt.grid(True)
plt.show()
Impact of Confidence Levels on Precision
Confidence Interval Width
0.525
0.500
0.475
0.450
0.425
0.400
0.375
0.350
0.90
0.92
0.94
0.96
0.98
Confidence Level
In [ ]:
30°C
Mostly cloudy
Search
ENG
IN
17:12
09-09-2023