Toggle navigation

- Logout
- Logout

Return to Classroom

Investigate a Dataset

- Review
- Code Review
- History

Meets Specifications

You've got your brain in gear 🥳 Congratulations! 🥇 🎉

✅ Your project meets all the requirements. So proud of you and your captivating analysis. I hope you enjoyed the journey as much I enjoyed reviewing your project. 👏

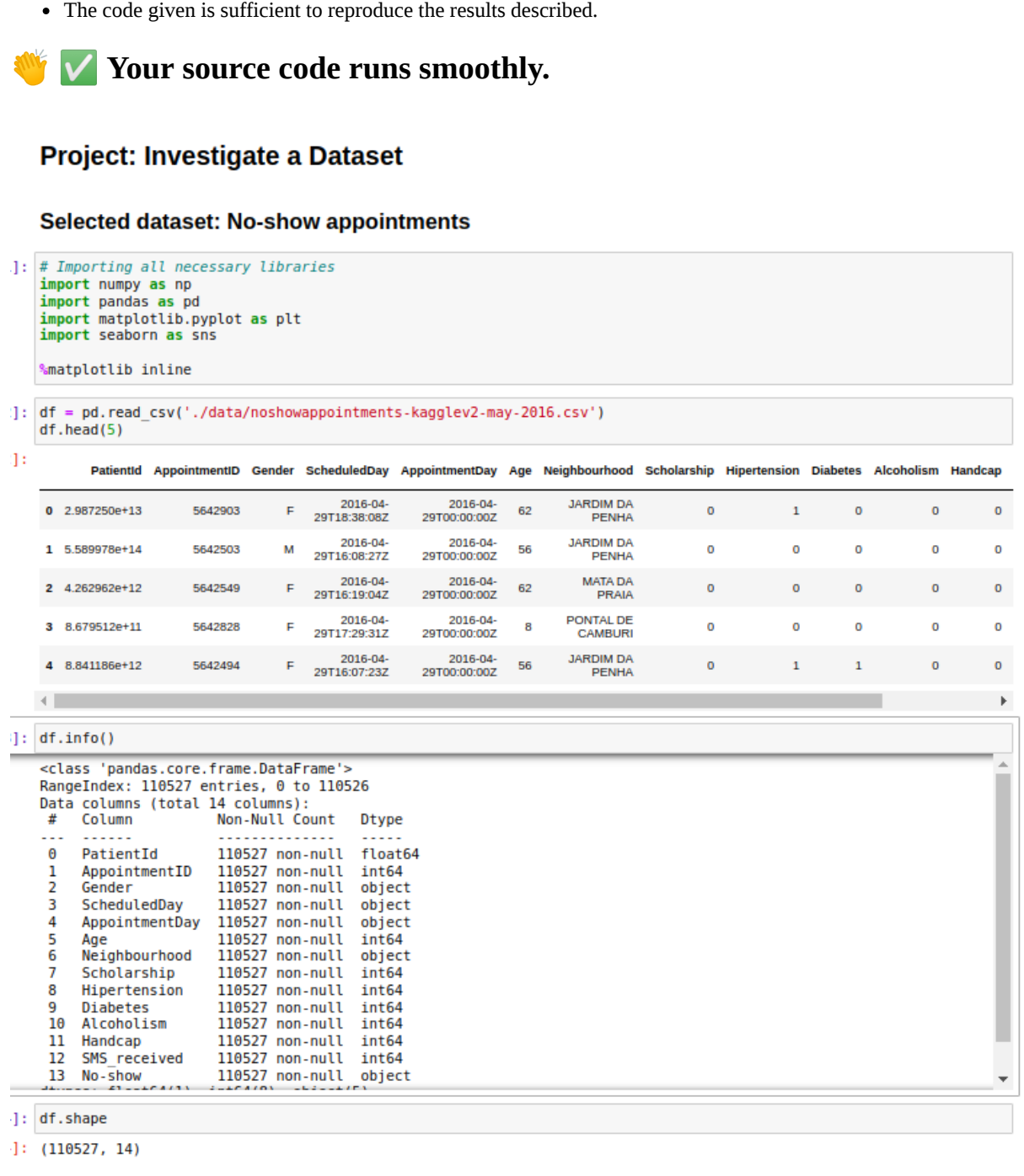
The visualizations are neat, you followed the correct steps in documenting the cleaning process and draw an accurate conclusion. Keep the great work.

I wanted to express my utmost gratitude for all the consistency you have done so far, and your analytical skills will drive you to move forward to the right direction. 😊

Code Functionality

- All code is functional and produces no errors when run.
- The code given is sufficient to reproduce the results described.

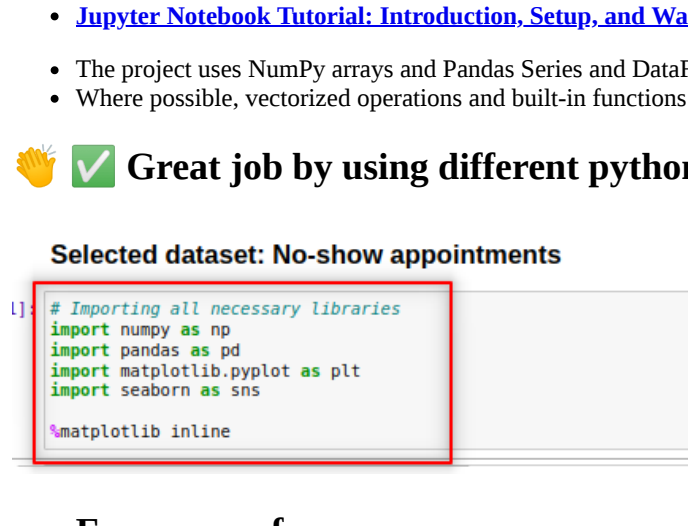
👏 ✅ Your source code runs smoothly.



For more reference:

- **10 common mistakes Python programmers make (and how to fix them)**
- **Jupyter Notebook Tutorial: Introduction, Setup, and Walkthrough**
- The project uses NumPy arrays and Pandas Series and DataFrames where appropriate rather than Python lists and dictionaries.
- Where possible, vectorized operations and built-in functions are used instead of loops.

👏 ✅ Great job by using different python libraries in your submission.



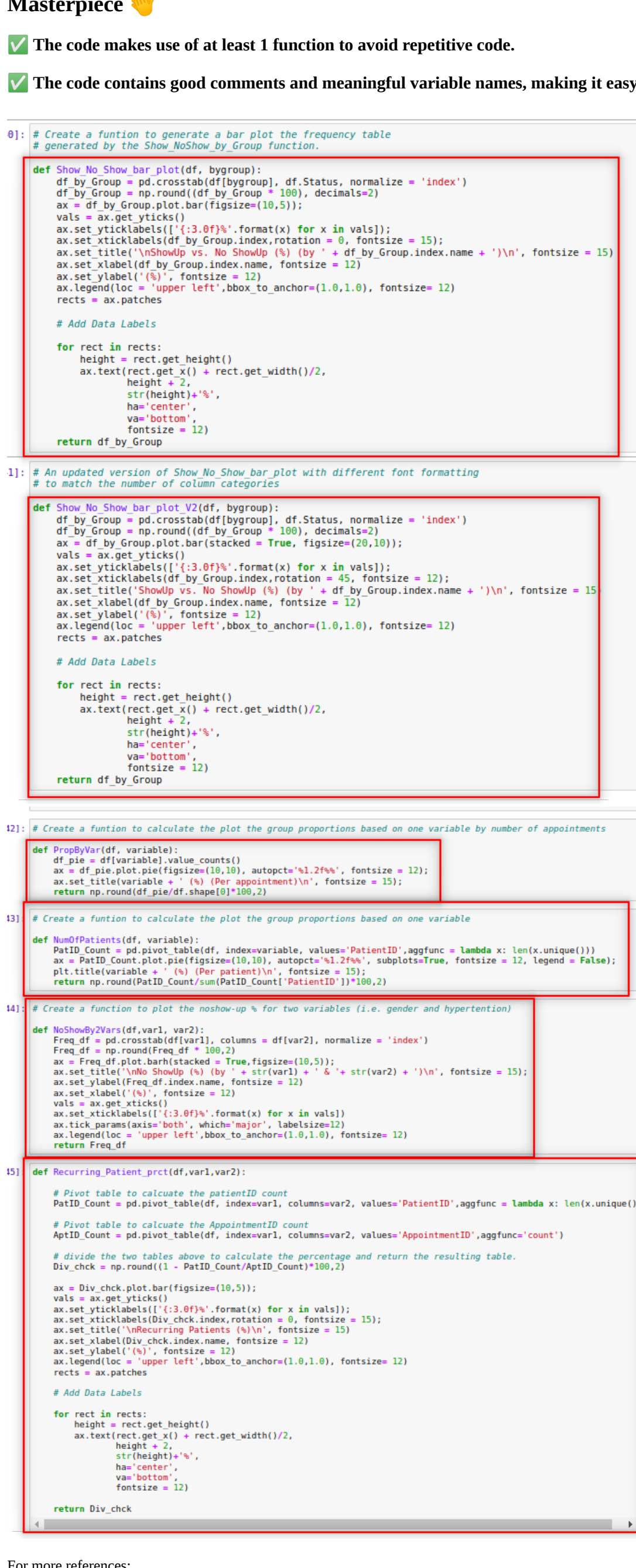
For more reference:

- **An introduction to seaborn**
- **Most Popular Python Packages in 2021**
- **The Ultimate NumPy Tutorial for Data Science Beginners**
- **Matplotlib Tutorial with Exercises - 1**
- The code makes use of at least 1 function to avoid repetitive code.
- The code contains good comments and meaningful variable names, making it easy to read.

Masterpiece 👏

✅ The code makes use of at least 1 function to avoid repetitive code.

✅ The code contains good comments and meaningful variable names, making it easy to read.



For more reference:

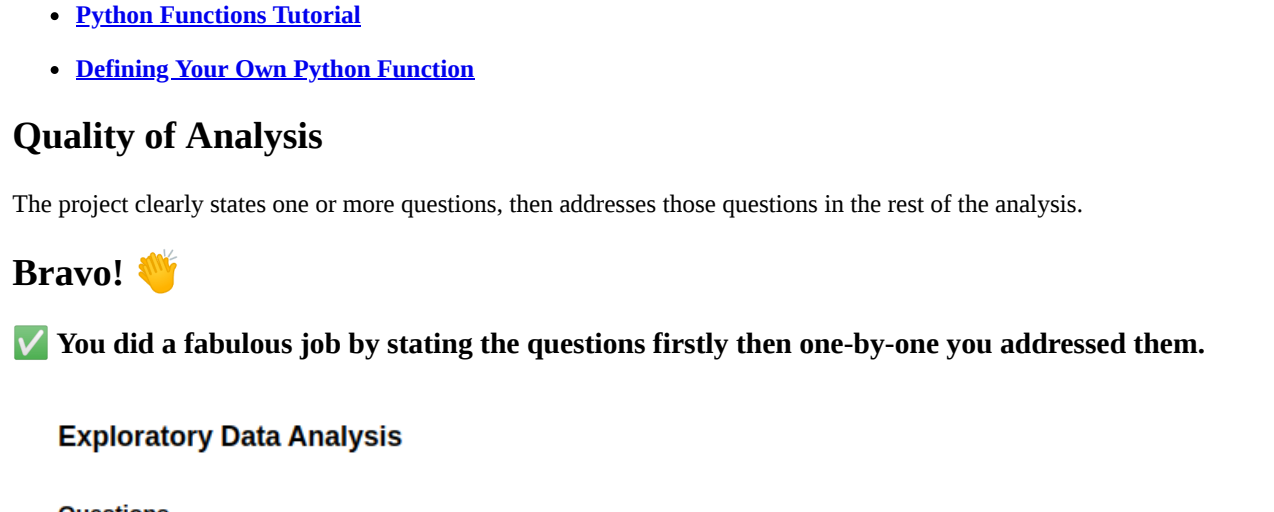
- **Clinging to memory: how Python function calls can increase memory use**
- **Python Functions: A Tutorial for Absolute Beginners #1**
- **Python Functions Tutorial**
- **Defining Your Own Python Function**

Quality of Analysis

The project clearly states one or more questions, then addresses those questions in the rest of the analysis.

Bravo! 👏

✅ You did a fabulous job by stating the questions firstly then one-by-one you addressed them.



For more reference:

- **Step 1: Identifying the focal issue with 'Problem Tree Analysis' technique**
- **Python Coding - Asking User's Questions**

Data Wrangling Phase

The project documents any changes that were made to clean the data, such as merging multiple files, handling missing values, etc.

You meet the requirements 👏 Great job!

✅ The changes in the data cleaning section is well documented!

Additional source:

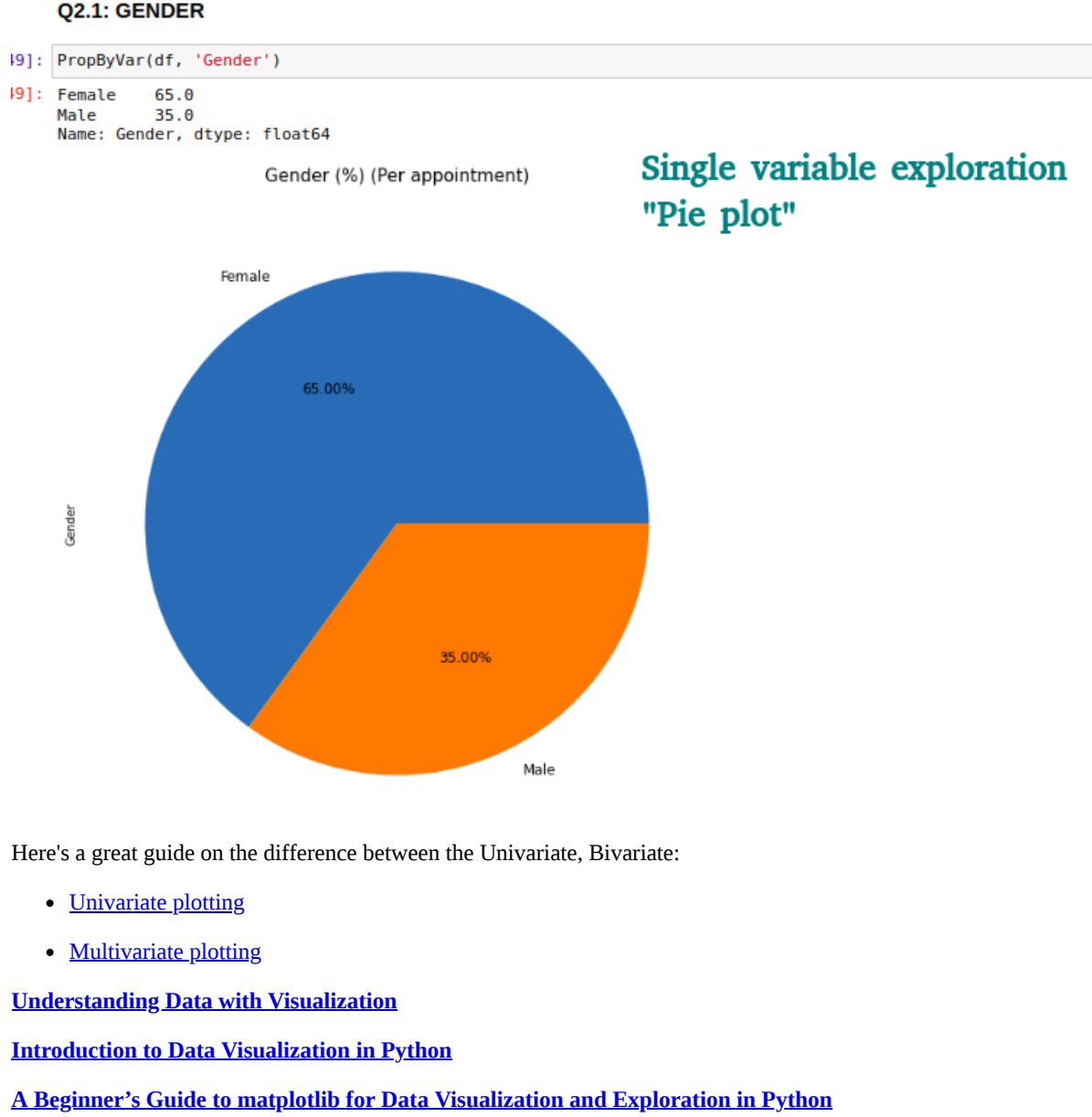
- **Data Cleaning Techniques in Python: the Ultimate Guide**
- **Handling Missing Data**
- **Pythonic Data Cleaning With Pandas and NumPy**

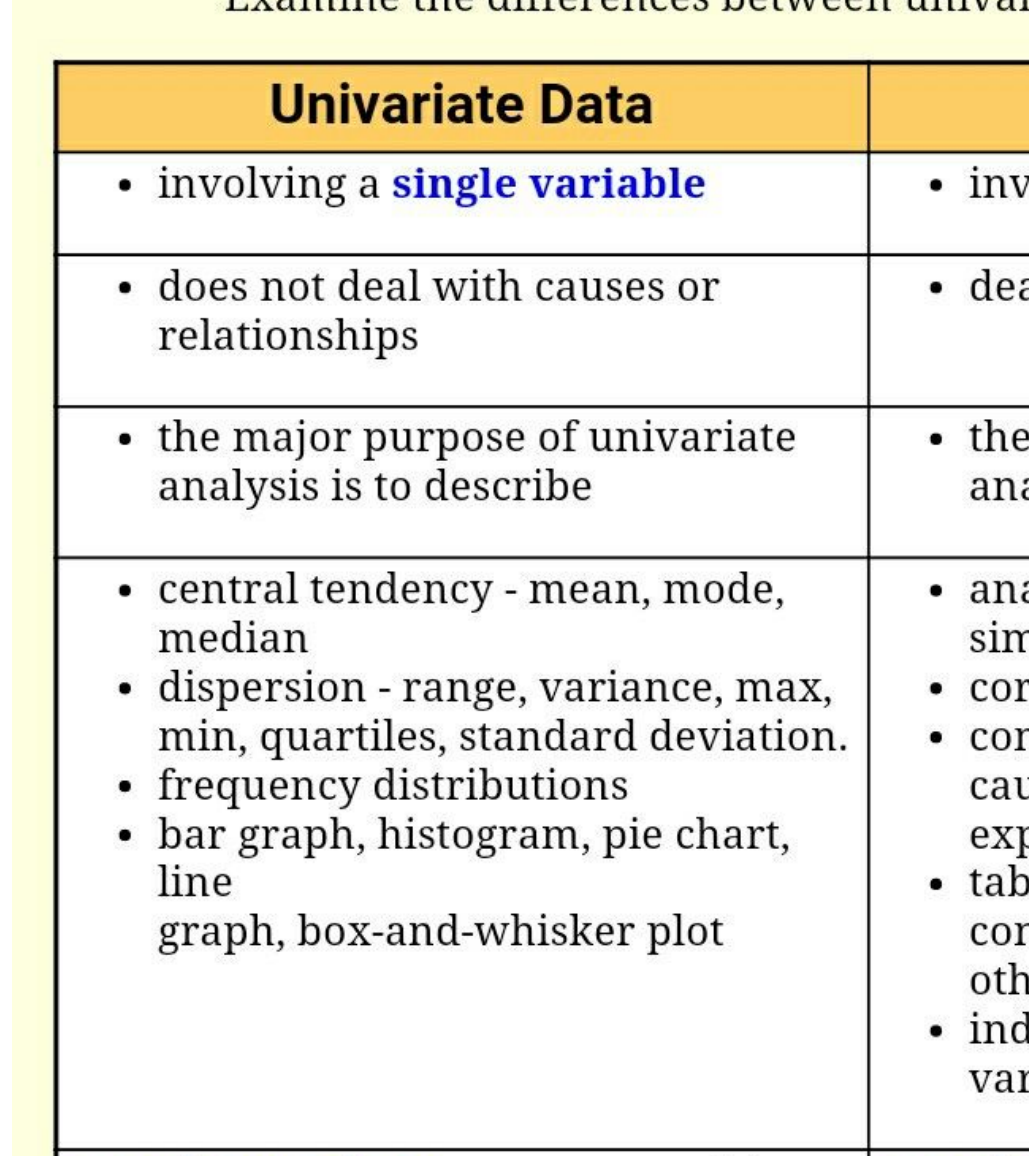
Exploration Phase

- The project investigates the stated question(s) from multiple angles.
- The project explores at least three variables in relation to the primary question. This can be an exploratory relationship between three variables of interest, or looking at how two independent variables relate to a single dependent variable of interest.
- The project performs both single-variable (1d) and multiple-variable (2d) explorations.

Faultless 👏

✅ The project investigates both single-variable (1d) and multiple-variable (2d) explorations.





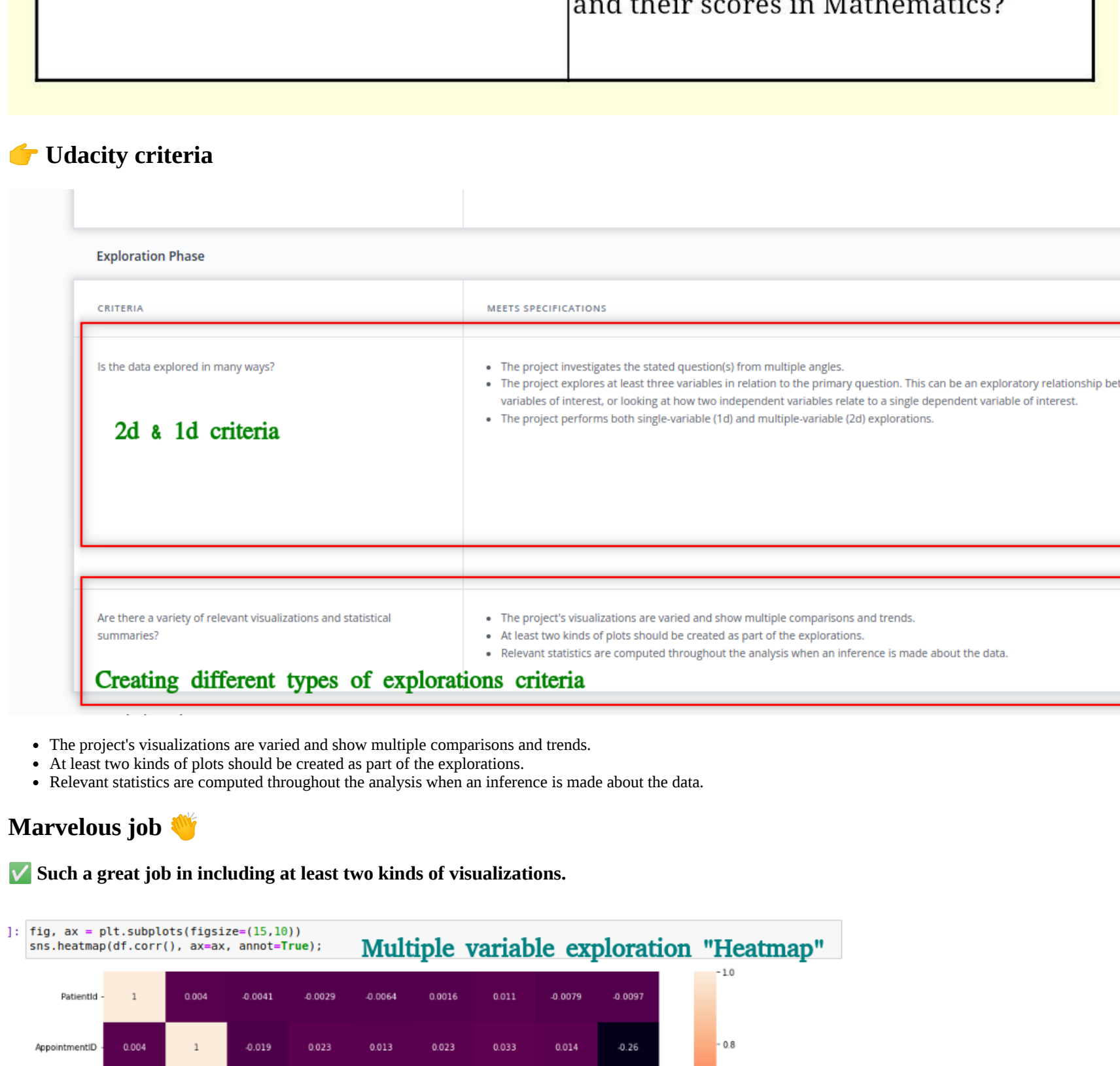
Here's a great guide on the difference between the Univariate, Bivariate:

- Univariate plotting
- Multivariate plotting

Understanding Data with Visualization

Introduction to Data Visualization in Python

A Beginner's Guide to matplotlib for Data Visualization and Exploration in Python



Examine the differences between univariate and bivariate data.

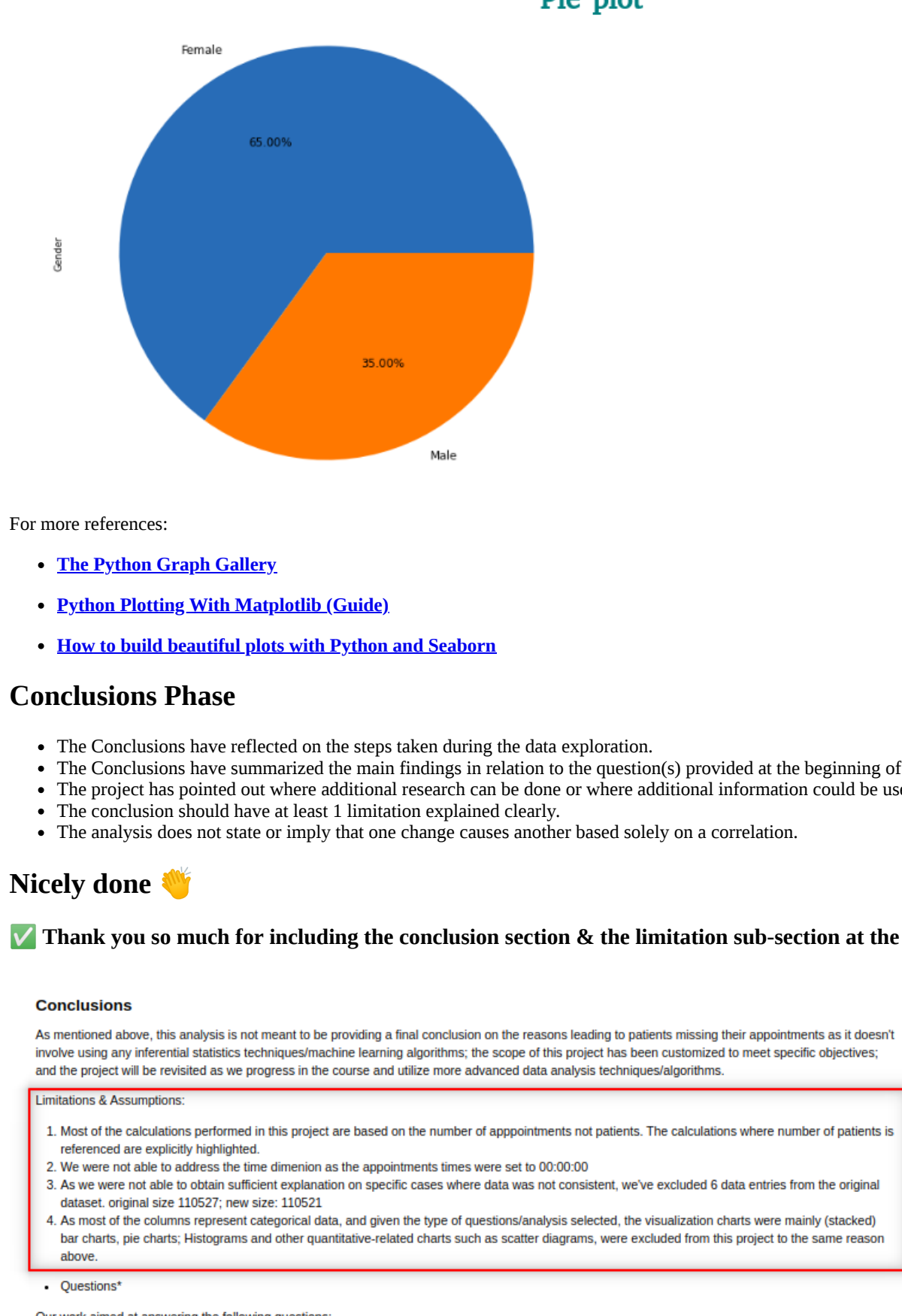
Univariate Data	Bivariate Data
• involving a **single variable**	• involving **two variables**
• does not deal with causes or relationships	• deals with causes or relationships
• the major purpose of univariate analysis is to describe	• the major purpose of bivariate analysis is to explain
• central tendency - mean, mode, median • dispersion - range, variance, max, min, quartiles, standard deviation. • frequency distributions • bar graph, histogram, pie chart, line graph, box-and-whisker plot	• analysis of two variables simultaneously • correlations • comparisons, relationships, causes, explanations • tables where one variable is contingent on the values of the other variable. • independent and dependent variables
Sample question: How many of the students in the freshman class are female?	**Sample question:** Is there a relationship between the number of females in Computer Programming and their scores in Mathematics?

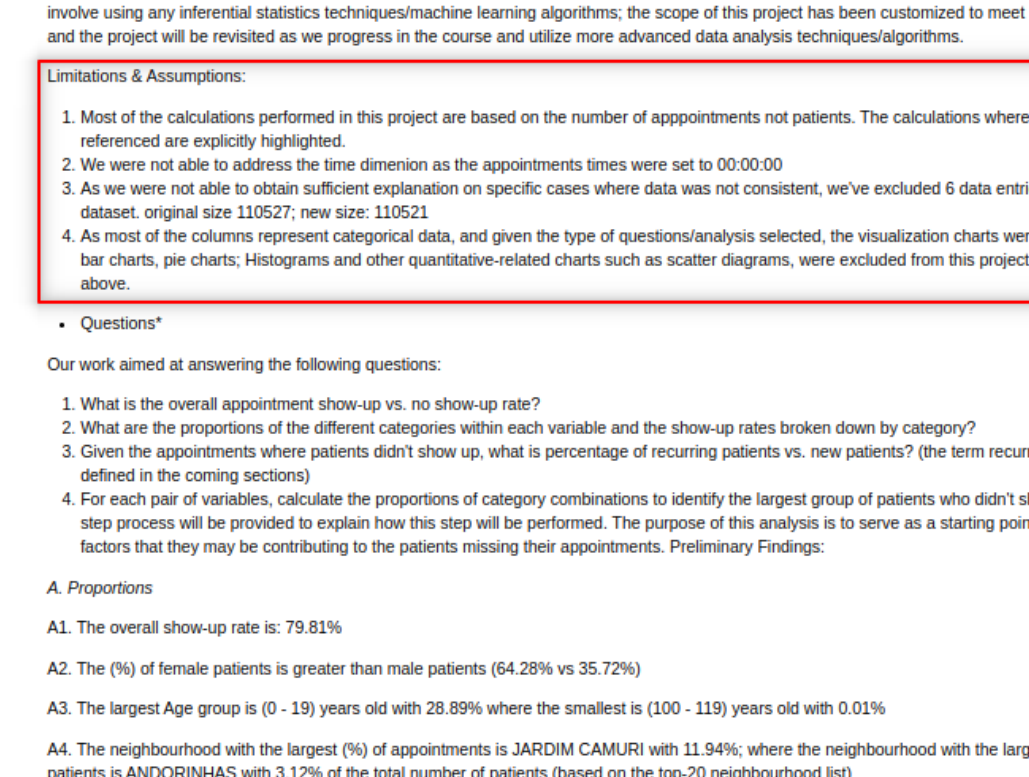
👉 Udacity criteria

- The project's visualizations are varied and show multiple comparisons and trends.
- At least two kinds of plots should be created as part of the explorations.
- Relevant statistics are computed throughout the analysis when an inference is made about the data.

Marvelous job 👏

✅ Such a great job in including at least two kinds of visualizations.





For more references:

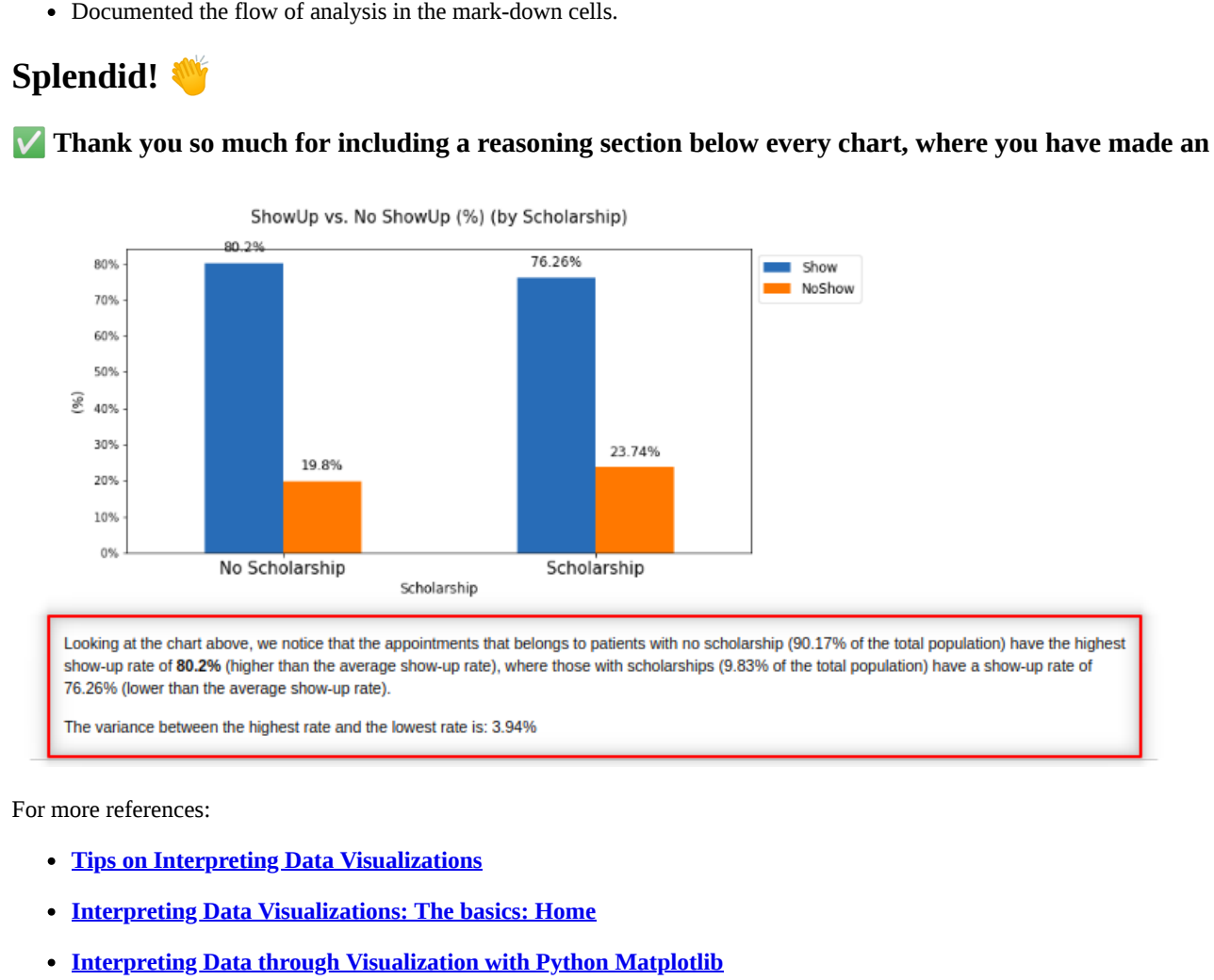
- **The Python Graph Gallery**
- **Python Plotting With Matplotlib (Guide)**
- **How to build beautiful plots with Python and Seaborn**

Conclusions Phase

- The Conclusions have reflected on the steps taken during the data exploration.
- The Conclusions have summarized the main findings in relation to the question(s) provided at the beginning of the analysis accurately.
- The project has pointed out where additional research can be done or where additional information could be useful.
- The conclusion should have at least 1 limitation explained clearly.
- The analysis does not state or imply that one change causes another based solely on a correlation.

Nicely done 👏

✅ Thank you so much for including the conclusion section & the limitation sub-section at the end of your submission.



Additional sources:

- **Python Statistics Fundamentals: How to Describe Your Data**
- **Conclusion - Learning to Program with Python 3 (basics)**
- **Limitations of the Study**
- **The Limitations of the Data in Predictive Analytics**

Communication

- The code should have ideally the following sections: Introduction; Questions; Data Wrangling; Exploratory Data Analysis; Conclusions, Limitation.
- Reasoning is provided for each analysis decision, plot, and statistical summary.
- Interpretation of plots and application of statistical tests should be correct and without error.
- Comments are used within the code cells.
- Documented the flow of analysis in the mark-down cells.

Splendid! 👏

✅ Thank you so much for including a reasoning section below every chart, where you have made an optimal analysis with different types of visualizations.

For more references:

- **Tips on Interpreting Data Visualizations**
- **Interpreting Data Visualizations: The basics: Home**
- **Interpreting Data through Visualization with Python Matplotlib**

Visualizations made in the project depict the data in an appropriate manner (i.e., has appropriate labels, scale, legends, and plot type) that allows plots to be readily interpreted.

Magnificent job 👏

✅ You did a wonderful work by making the visualization well interpreted.

TIPS:

- We add the plot title using the plt.title() function. Here's a great documentation on the title function.
- We also add the x-axis label using the plt.xlabel() function. Here's a great documentation on the xlabel function.
- We add the y-axis label using the plt.ylabel() function. Here's a great documentation on the ylabel function.

Download Download Project

Return to Path

Rate this review Start

Rate this review Start