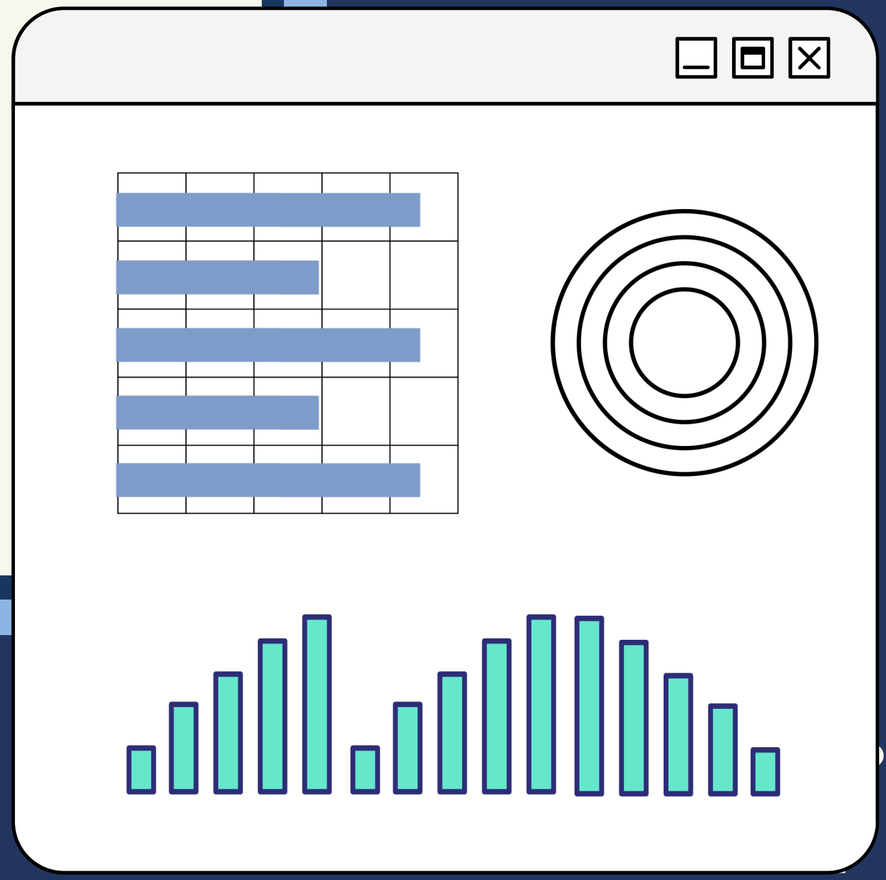
EDA-assistant

A Python package for easy EDA

Madalyn Li
University of Washington
DATA 515: Winter 2022



MOTIVATION

Target Users:

1. Data Scientists and Data Analysts
    - Seeking simpler & quicker EDA process in Python
2. Inexperienced Python Users
    - Unfamiliar with Pandas, Seaborn, Matplotlib and with analyzing data sets in Python

User Pain Points:

- Current EDA process in Python involves importing many packages and multiple lines of code
- No concise summary report format available for standard EDA information

SOLUTION:

- A one-stop shop Python package for EDA tasks
- Auto generates report containing all standard EDA summary statistics and graphs

USER DESIGN GOALS

SIMPLE UI

- Only requires necessary inputs from users
- pip installable

CONTAINS BASIC EDA TECHNIQUES

- Dataset Summary Statistics:
  - # rows, # columns, etc.
- Variable Summary Statistics:
  - mean, median, sum, etc. for numerical variables
  - number of missing values and unique values
- Univariate Graphs:
  - bar charts for distribution
- Bivariate Graphs:
  - correlation Matrix heat map
  - scatter pair plot

AESTHETIC REPORT OUTPUT

- A lower priority than the rest
- Somewhat important for end user likeability

SYSTEM DESIGN GOALS

EDA CLASS

- Add future improvements to enhance functionality
  - Outlier detection and removal function

SEPARATION OF CONCERNS

- Well thought out abstraction of tasks
  - Easily update or add calculations for summary statistics table
  - Easily update format of graphs
  - Easily update design/format of pdf output

USE CASE: Creating an EDA report

User:

- Data Scientists and Data Analysts

Preconditions:

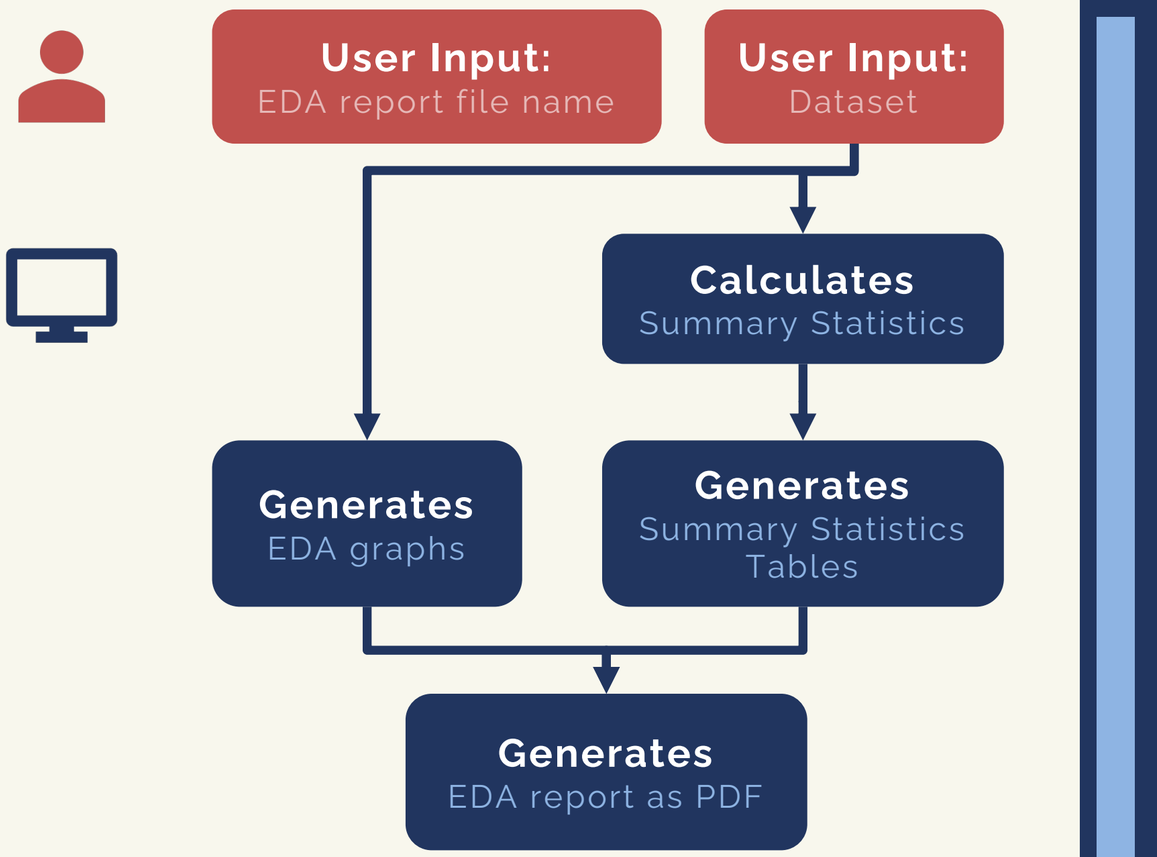
- User instantiates EDA class with a non-empty dataset

Triggers:

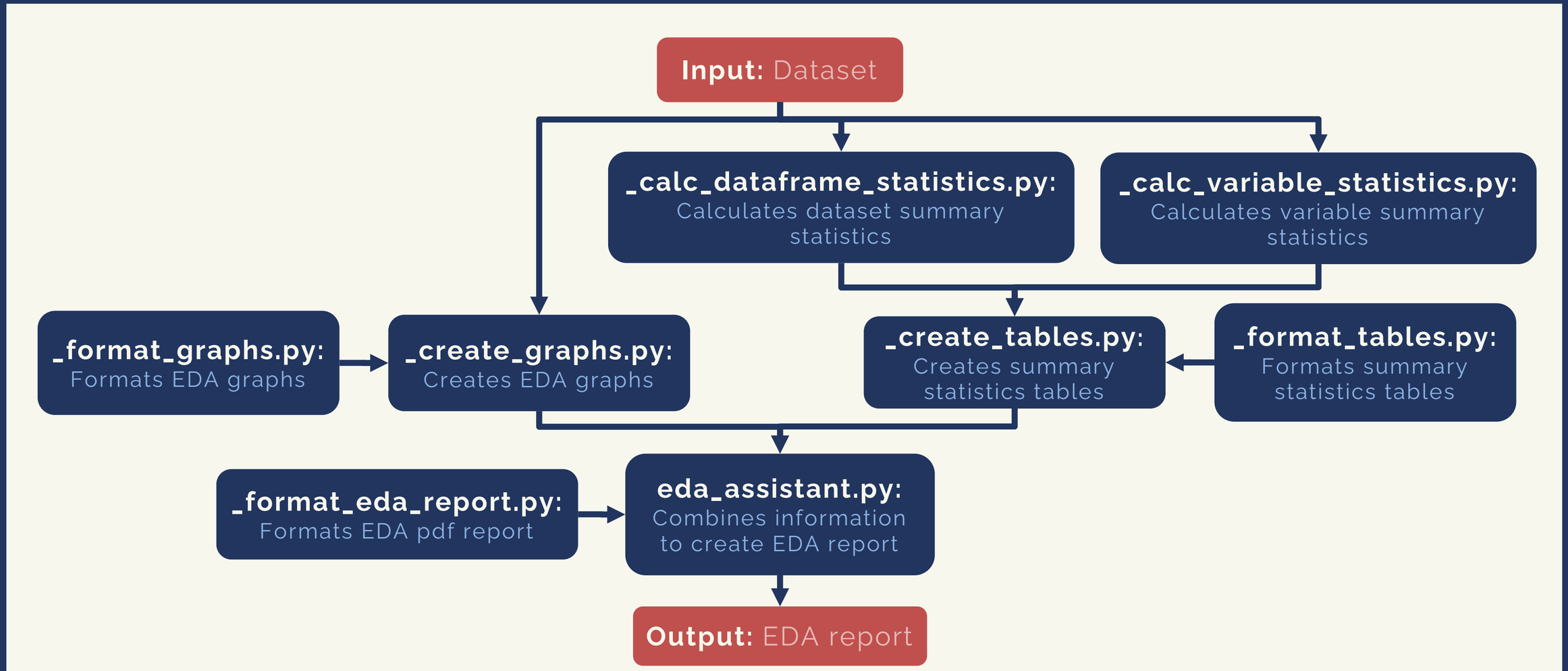
- User calls create_eda_report

Use Case Overview:

- User inputs file path to dataset and EDA report save file name
- System calculates summary statistics
- System generates summary statistics table & EDA graphs
- System generates EDA report



SYSTEM COMPONENTS



```
EDA-assistant/
 |- eda_assistant/
  |- __init__.py
  |- _calc_dataframe_statistics.py
  |- _calc_variable_statistics.py
  |- _create_graphs.py
  |- _create_tables.py
  |- _format_eda_report.py
  |- _format_graphs.py
  |- _format_tables.py
  |- eda_eassistant.py
  |- tests/
   |- __init__.py
   |- test_calc_dataframe_statistics.py
   |- test_calc_variable_statistics.py
   |- test_create_tables.py
   |- test_eda_assistant.py
   |- test_format_graphs.py
   |- test_format_tables.py
 |- data/
  |- IRIS.csv
  |- WineQT.csv
  |- cereal.csv
  |- test_create_tables_results/
   |- test_create_df_summary_cereal_results.csv
   |- test_create_var_summary_cereal_results.csv
 |- docs/
  |- EDA_assistant_final_presentation.pdf
  |- EDA_assistant_written_report.pdf
 |- examples/
  |- demo_EDA_assistant.ipynb
  |- demo_iris_eda_report.pdf
  |- demo_iris_eda_report_cat_hist.png
  |- demo_iris_eda_report_corr.png
  |- demo_iris_eda_report_df_table.png
  |- demo_iris_eda_report_num_hist.png
  |- demo_iris_eda_report_pair.png
  |- demo_iris_eda_report_var_table.png
  |- demo_wine_eda_report.pdf
 |- LICENSE
 |- README.md
 |- requirements.txt
 |- setup.py
```

SOFTWARE & LICENSING INFORMATION

```
pip install EDA-assistant
```

GitHub repository:
https://github.com/madalynli/EDA-assistant

Python Version: 3+

License: MIT

File Size: 37KB

DEMO:

create_eda_report

DATA SOURCE:

Iris Flower Dataset

- https://www.kaggle.com/arshid/iris-flower-dataset

PIP INSTALL THE PACKAGE:

```
pip install EDA-assistant
```

IMPORT THE PACKAGE:

```
from eda_assistant import eda_assistant
```

INSTANTIATE THE EDA CLASS:

```
eda_iris = eda_assistant.EDA('IRIS.csv')
```

CREATE EDA REPORT:

```
eda_iris.create_eda_report('iris_eda_report.pdf')
```

OUTPUT: Tables

Data Set Summary Statistics:

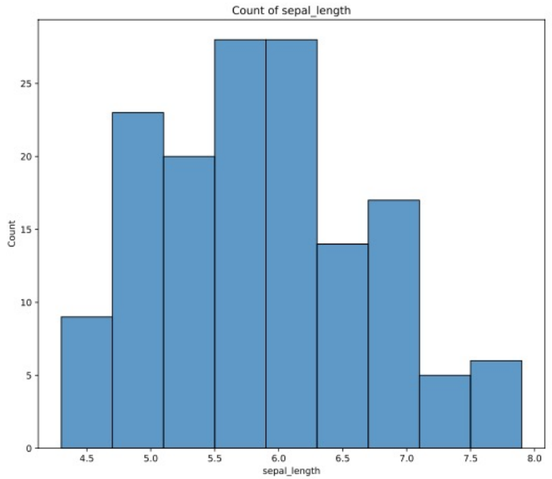
	Values
No. of Columns	5
No. of Rows	150
Total Value Count	750
Count of NaNs	0
Percent of NaNs	0.0%
Count of Duplicate Rows	3
Percent of Duplicate Rows	0.4%
Count of Numerical Variables	4
Count of Categorical Variables	1

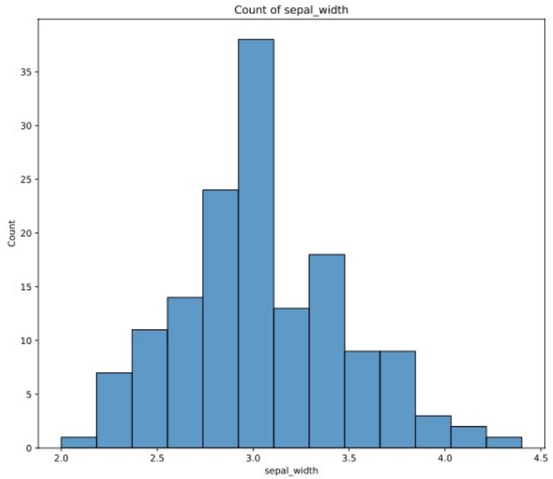
Variable Summary Statistics:

	sepal_length	sepal_width	petal_length	petal_width	species
Variable Type	float	float	float	float	object
Mean	5.84	3.05	3.76	1.2	-
Median	5.8	3.0	4.35	1.3	-
Sum	876.5	458.1	563.8	179.8	-
Variance	0.69	0.19	3.11	0.58	-
Standard Deviation	0.83	0.43	1.76	0.76	-
25 Percentile	5.1	2.8	1.6	0.3	-
75 Percentile	6.4	3.3	5.1	1.8	-
Min	4.3	2.0	1.0	0.1	-
Max	7.9	4.4	6.9	2.5	-
Skew	0.31	0.33	-0.27	-0.1	-
Count of NaNs	0	0	0	0	0
Percent of NaNs	0.0%	0.0%	0.0%	0.0%	0.0%
Count of Unique Values	35	23	43	22	3

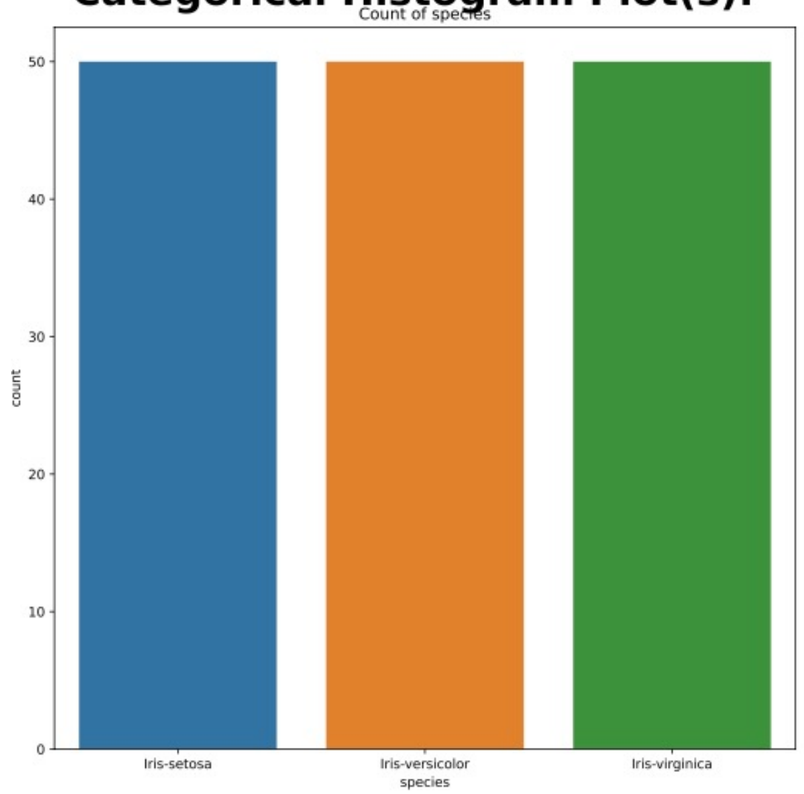
OUTPUT: Univariate Graphs

Numerical Histogram Plot(s):



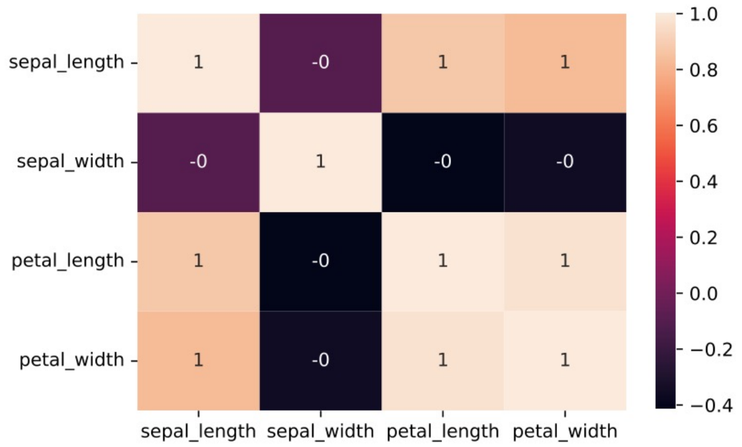


Categorical Histogram Plot(s):

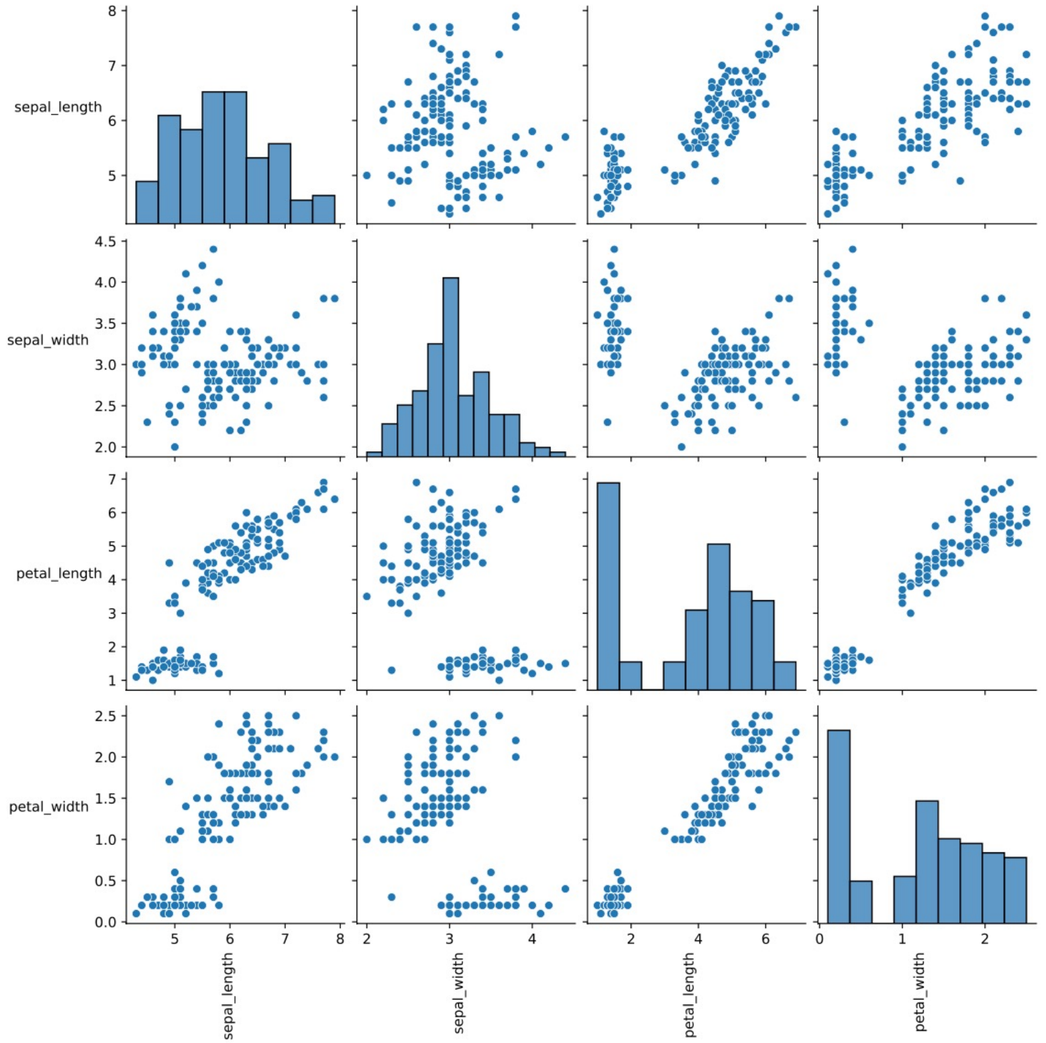


OUTPUT:
Bivariate Graphs

Correlation Matrix Heat Map:



Pair Plots:



LEARNINGS & CHALLENGES

PIP INSTALL

Creating a pip installable package with PyPI

PdfPages

Creating a multi-page PDF output in Python with matplotlib

UNIT TESTS

Writing unit tests for summary statistic calculations and tables

PYTHON PACKAGE

Creating, formatting, and documenting a Python package

FUTURE WORK

ADDING OUTLIER FUNCTIONALITY

Extend the EDA class by adding an outlier detector and remover

ADAPT TO LARGER DATA FILES

Improve run time and code complexity to handle larger data set sizes

ADD UNIT TESTS FOR GRAPHS

100% test coverage for univariate and bivariate graphs

THANK YOU!