

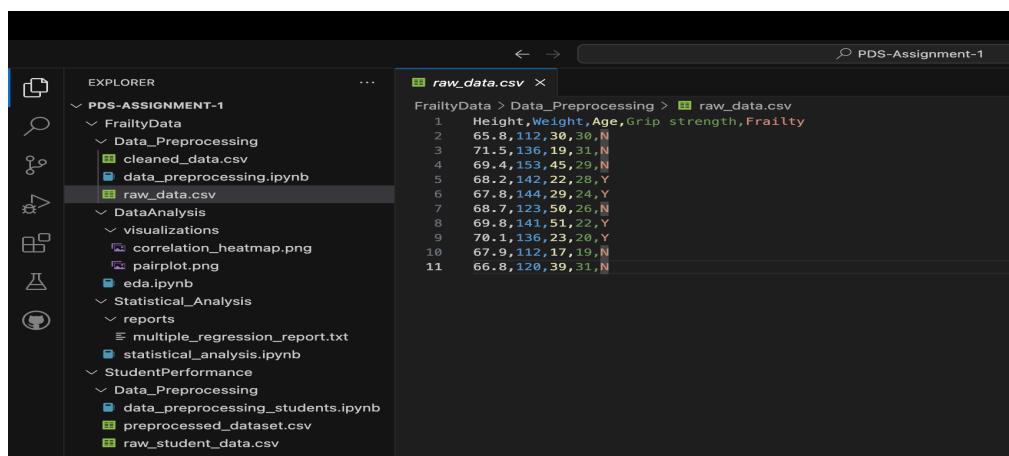
ASSIGNMENT-1
Name: ENUKONDA SAI DEEPTHI
STUDENT ID:16343756

1) Frailty is physical weakness; lack of health or strength. Reduced grip strength in females correlated with higher frailty scores and vice versa. Hand grip strength can be quantified by measuring the amount of static force that the hand can squeeze around a dynamometer. The force has most commonly been measured in kilograms and pounds. The table below represents data from 10 female participants. The Height is measured in inches, Weight in pounds, Age in years, Grip strength in kilograms. Frailty is qualitative attribute indicated the presence or absence of the symptoms. Based on the following table, design the three stages of reproducible workflow, includes the work you can do and the folder structure in each stage (reference study case in chapter 3). (5 points)

Height	Weight	Age	Grip strength	Frailty
65.8	112	30	30	N
71.5	136	19	31	N
69.4	153	45	29	N
68.2	142	22	28	Y
67.8	144	29	24	Y
68.7	123	50	26	N
69.8	141	51	22	Y
70.1	136	23	20	Y
67.9	112	17	19	N
66.8	120	39	31	N

STEP 1: DATA PREPROCESSING

- The below image shows the raw dataset before data preprocessing:

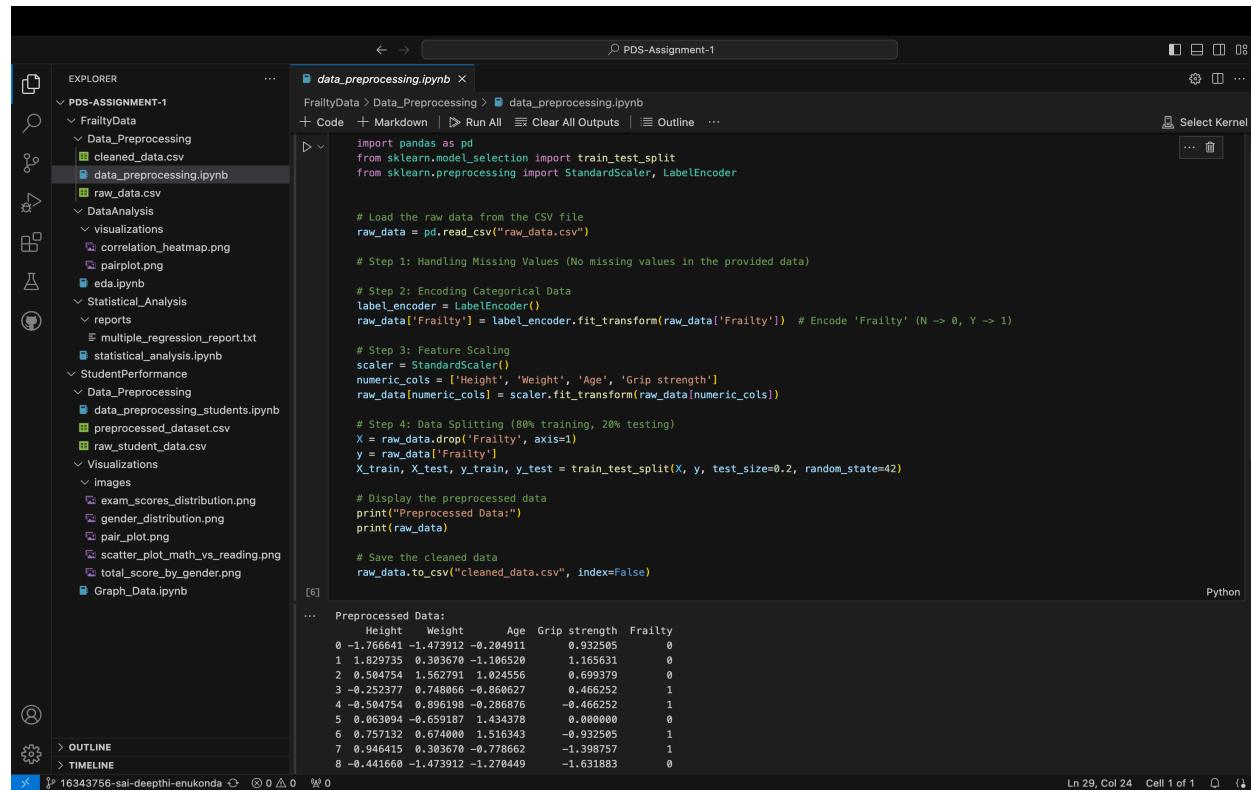


The screenshot shows a Jupyter Notebook environment with the following details:

- EXPLORER:** A sidebar showing the project structure:
 - PDS-ASSIGNMENT-1
 - FrailtyData
 - Data_Preprocessing
 - cleaned_data.csv
 - data_preprocessing.ipynb
 - raw_data.csv
 - DataAnalysis
 - visualizations
 - correlation_heatmap.png
 - pairplot.png
 - eda.ipynb
 - Statistical_Analysis
 - reports
 - multiple_regression_report.txt
 - statistical_analysis.ipynb
 - StudentPerformance
 - Data_Preprocessing
 - data_preprocessing_students.ipynb
 - preprocessed_dataset.csv
 - raw_student_data.csv

- I've performed several preprocessing steps on the given dataset:
- Data Loading:** I created a DataFrame using the provided data, which contains information about individuals' height, weight, age, grip strength, and frailty status.
 - Encoding Categorical Data:** The 'Frailty' column contains categorical data with values 'N' (No) and 'Y' (Yes). I used label encoding to convert these categorical values to numerical values:
 - 'N' is encoded as 0.
 - 'Y' is encoded as 1.
 - Standardization of Numeric Features:** I standardized the numeric columns ('Height', 'Weight', 'Age', and 'Grip strength') to ensure that they have similar scales. Standardization transforms the data so that it has a mean of 0 and a standard deviation of 1. This helps prevent certain machine learning algorithms from being sensitive to the scale of input features.
 - Data Splitting:** I split the data into a training set and a testing set using the `train_test_split` function from scikit-learn. The split was done with 80% of the data used for training and 20% for testing. This separation allows you to train machine learning models on one subset and evaluate their performance on another, helping to assess how well your models generalize to unseen data.

The code concludes by printing the preprocessed data to the console, which you can use for further analysis or modeling tasks. These preprocessing steps help prepare the data for machine learning or statistical analysis by making it more suitable and consistent for modeling techniques.



```

PDS-Assessment-1
EXPLORER
  PDS-ASSIGNMENT-1
    Data_Preprocessing
      cleaned_data.csv
      data_preprocessing.ipynb
      raw_data.csv
    DataAnalysis
      correlation_heatmap.png
      pairplot.png
      eda.ipynb
    Statistical_Analysis
      reports
        multiple_regression_report.txt
      statistical_analysis.ipynb
    StudentPerformance
      Data_Preprocessing
        data_preprocessing_students.ipynb
        preprocessed_dataset.csv
        raw_student_data.csv
      Visualizations
        exam_scores_distribution.png
        gender_distribution.png
        pair_plot.png
        scatter_plot_math_vs_reading.png
        total_score_by_gender.png
      Graph_Data.ipynb
  FrailtyData > Data_Preprocessing > data_preprocessing.ipynb
  + Code | Markdown | ▶ Run All | Clear All Outputs | Outline ...
  Select Kernel ... ⚙️ 🗑️

data_preprocessing.ipynb ×
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Load the raw data from the CSV file
raw_data = pd.read_csv("raw_data.csv")

# Step 1: Handling Missing Values (No missing values in the provided data)

# Step 2: Encoding Categorical Data
label_encoder = LabelEncoder()
raw_data['Frailty'] = label_encoder.fit_transform(raw_data['Frailty']) # Encode 'Frailty' (N -> 0, Y -> 1)

# Step 3: Feature Scaling
scaler = StandardScaler()
numeric_cols = ['Height', 'Weight', 'Age', 'Grip strength']
raw_data[numeric_cols] = scaler.fit_transform(raw_data[numeric_cols])

# Step 4: Data Splitting (80% training, 20% testing)
X = raw_data.drop('Frailty', axis=1)
y = raw_data['Frailty']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Display the preprocessed data
print("Preprocessed Data:")
print(raw_data)

# Save the cleaned data
raw_data.to_csv("cleaned_data.csv", index=False)

Python
[6]
Preprocessed Data:
   Height    Weight     Age  Grip strength  Frailty
0 -1.766641 -1.473912 -0.204911   0.932505      0
1  1.829735  0.303670 -1.106520   1.165631      0
2  0.504754  1.562791  1.024556   0.699379      0
3 -0.252377  0.748066 -0.860627   0.466252      1
4 -0.584754  0.896198 -0.286876  -0.466252      1
5  0.063094 -0.659187  1.434378   0.000000      0
6  0.757132  0.674000  1.516343  -0.932505      1
7  0.946415  0.303670 -0.778662  -1.398757      1
8 -0.441660 -1.473912 -1.270449  -1.631883      0

```

The output obtained here is the preprocessed data. Here's what each column represents:

- **Height:** Standardized height values.
- **Weight:** Standardized weight values.
- **Age:** Standardized age values.
- **Grip strength:** Standardized grip strength values.
- **Frailty:** Encoded frailty status (0 for 'N' and 1 for 'Y').

This output indicates that the preprocessing steps were successfully applied to the dataset. All numeric features have been standardized (scaled to have a mean of 0 and a standard deviation of 1), and the 'Frailty' column has been encoded into numerical values.

- The below is the cleaned data is obtained after the preprocessing

The screenshot shows a Jupyter Notebook environment. The left sidebar (EXPLORER) lists several notebooks and files under the 'PDS-ASSIGNMENT-1' project. The 'cleaned_data.csv' file is selected and highlighted in blue. The right side shows the code editor with the first few lines of the 'cleaned_data.csv' file. The file contains five columns: Height, Weight, Age, Grip strength, and Frailty. The data is in a tab-separated format with numerical values.

```
Height,Weight,Age,Grip strength,Frailty
1 -0.20491114929764814,0,0.9325048802403138,0
2 -1.766640627937739,-1.4739123367382712,-0.20491114929764814,0,0.9325048802403138,0
3 1.829734936078378,0,0.3036703889360252,-1.1065202062072999,1,1.1656310183003922,0
4 0.5047544651250759,1.5627914726219851,1,0.24557464882407,0,0.693780601802353,0
5 -0.252377232562529,0,0.7480650603545993,-0.8606268270501222,0,0.4662524841201569,1
6 -0.5047544651250669,0,0.896107953494124,-0.2868756909167074,-0.4662524041201569,1
7 0.065309430814063897,-0.6591869244708853,1,1.433788450835368,0,0,0
8 0.7571316976876049,0,0.740001137848369,1.51634250408025962,-0.9325048802403138,1
9 0.946146221095039,0,0.3036703889360252,-0.77862623673210629,1,3.987571123604706,1
10 -0.441660156984426,-1.4739123367382712,-1.2704431256451805,-1.631083414420549,0
11 -1.135697546531403,-0.8813847641801724,0,0.5327689881738852,1,1.656310103003922,0
12
```

STEP 2: DATA ANALYSIS:

This code is meant for data analysis and report generation, providing visualizations and statistics to help understand the dataset and its relationships.

Steps done as below:

1. Imported necessary libraries for data analysis and visualization.
2. Suppresses future warning messages.
3. Created a directory called 'visualizations' for storing generated images.
4. Loaded cleaned data from a CSV file.
5. Computes and prints descriptive statistics of the data.
6. Generated a pair plot to visualize relationships between numeric variables and saves it as an image.
7. Creates a correlation heatmap to show variable correlations and saves it as an image.

PDS-Assessment-1

EXPLORER

- PDS-ASSIGNMENT-1
 - FrailtyData
 - Data_Preprocessing
 - cleaned_data.csv
 - data_preprocessing.ipynb
 - raw_data.csv
 - DataAnalysis
 - visualizations
 - correlation_heatmap.png
 - pairplot.png
 - eda.ipynb
- Statistical_Analysis
 - reports
 - multiple_regression_report.txt
 - statistical_analysis.ipynb
- StudentPerformance
 - Data_Preprocessing
 - data_preprocessing_students.ipynb
 - preprocessed_dataset.csv
 - raw_student_data.csv
 - Visualizations
 - exam_scores_distribution.png
 - gender_distribution.png
 - pair_plot.png
 - scatter_plot_math_vs_reading.png
 - total_score_by_gender.png
 - Graph_Data.ipynb

Code

```
# eda.ipynb
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
import warnings

with warnings.catch_warnings():
    warnings.filterwarnings("ignore", category=FutureWarning)

    # Ensure the 'visualizations' directory exists
    if not os.path.exists("visualizations"):
        os.makedirs("visualizations")

    # Load the cleaned data
    cleaned_data = pd.read_csv("../Data_Preprocessing/cleaned_data.csv")
    # Descriptive statistics
    print(cleaned_data.describe())

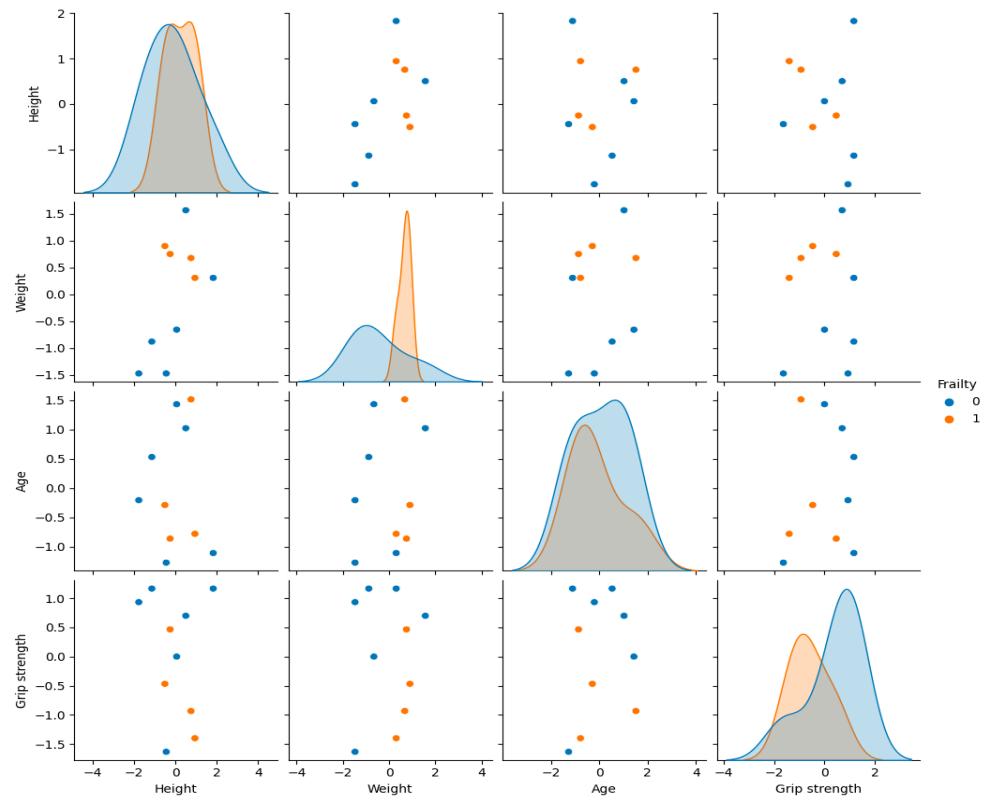
    # Pairplot of numeric variables
    sns.pairplot(cleaned_data, hue='Frailty')
    plt.savefig("visualizations/pairplot.png")
    plt.show()

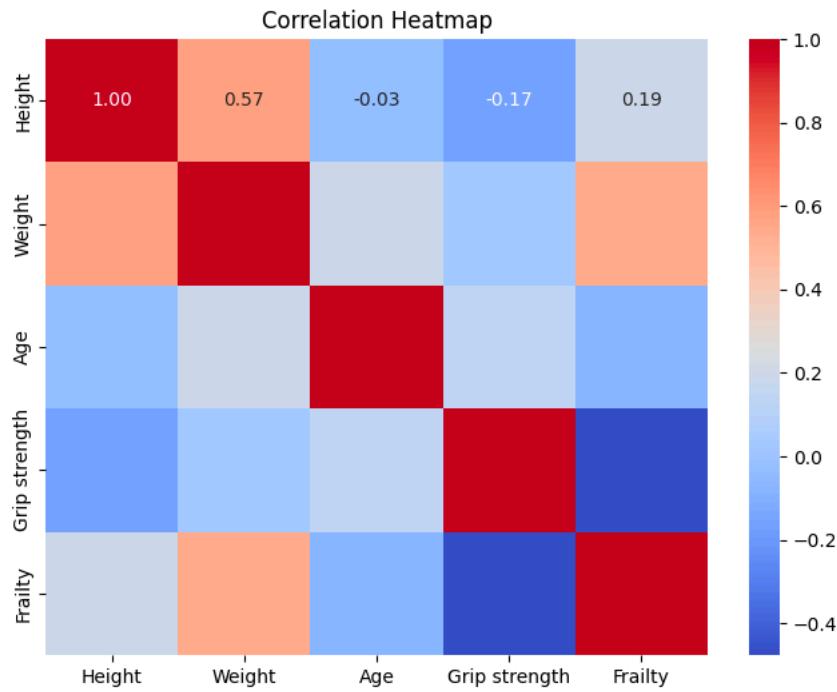
    # Correlation heatmap
    plt.figure(figsize=(8, 6))
    correlation_matrix = cleaned_data.corr()
    sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
    plt.title("Correlation Heatmap")
    plt.savefig("visualizations/correlation_heatmap.png")
    plt.show()
```

Python

	Height	Weight	Age	Grip strength	Frailty
count	1.000000e+01	1.000000e+01	1.000000e+01	1.000000e+01	1.000000e+00
mean	3.597123e-15	-4.40892e-16	1.118223e-17	4.440892e-17	0.400000e+00
std	1.054093e+00	1.054093e+00	1.054093e+00	1.054093e+00	0.516398e+00
min	-1.766641e+00	-1.473912e+00	-1.278449e+00	-1.631883e+00	0.000000e+00
25%	-4.889809e-01	-8.258353e-01	-8.401357e-01	-8.159417e-01	0.000000e+00
50%	-9.464146e-02	3.036704e-01	-2.458934e-01	2.331262e-01	0.000000e+00

Ln 18, Col 35 Cell 1 of 1





STEP 3: STATISTICAL ANALYSIS:

This code loads data, performs a multiple regression analysis, and saves the analysis summary as a text report for further reference and reporting.

1. Imports necessary libraries, including pandas for data handling and statsmodels for statistical analysis.
2. Loaded cleaned data from a CSV file named "cleaned_data.csv" located in the "../Data_Preprocessing/" directory.
3. Performs a multiple regression analysis using the statsmodels library:
 - Defines predictor variables (Age, Height, and Weight) and adds a constant term to the predictor matrix.
 - Defines the target variable (Grip strength).
 - Fits an ordinary least squares (OLS) multiple regression model, which estimates the relationship between the predictors and the target variable.
4. Prints a summary of the regression model, including statistics like coefficients, R-squared, and p-values.
5. Saves the analysis report to a text file named "multiple_regression_report.txt" in a directory named "reports." The report contains the same summary information as displayed in the console.

```

statistical_analysis.ipynb
FrailtyData > Statistical_Analysis > statistical_analysis.ipynb
+ Code + Markdown | Run All | Clear All Outputs | Outline ...
import pandas as pd
import statsmodels.api as sm

# Load the cleaned data
cleaned_data = pd.read_csv("../Data_Preprocessing/cleaned_data.csv")

# Perform multiple regression analysis
X = cleaned_data[['Age', 'Height', 'Weight']] # Include multiple predictors
X = sm.add_constant(X) # Add a constant term
y = cleaned_data['Grip strength']

model = sm.OLS(y, X).fit()
print(model.summary())

# Save the analysis report
with open("reports/multiple_regression_report.txt", 'w') as report_file:
    report_file.write(str(model.summary()))

```

OLS Regression Results

Dep. Variable:	Grip strength	R-squared:	0.061			
Model:	OLS	Adj. R-squared:	-0.408			
Method:	Least Squares	F-statistic:	0.1300			
Date:	Tue, 26 Sep 2023	Prob (F-statistic):	0.939			
Time:	19:15:22	Log-Likelihood:	-13.874			
No. Observations:	10	AIC:	35.75			
Df Residuals:	6	BIC:	36.96			
Df Model:	3					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	4.025e-17	0.396	1.02e-16	1.000	-0.968	0.968
Age	0.0945	0.499	0.231	0.825	-0.987	1.096
Height	-0.2570	0.498	-0.525	0.619	-1.455	0.941
Weight	0.1616	0.499	0.324	0.757	-1.058	1.382
Omnibus:		0.857	Durbin-Watson:		1.131	
Prob(Omnibus):		0.652	Jarque-Bera (JB):		0.602	
Skew:		0.093	Prob (JB):		0.740	
Kurtosis:		1.812	Cond. No.		2.03	

2) Perform 5 data visualization tasks on the student performance dataset given in the link below (create 5 different visualizations). Explain what kind analysis has become easier with each of the visualizations. Create the folder structure for this question similar to question 1.

STEP 1: DATA PREPROCESSING:

The following steps are taken for the student performance dataset :

1. Loaded a dataset from a CSV file.
2. Checked for missing values in the dataset.
3. Performed one-hot encoding on categorical variables.
4. Calculated a 'total score' feature by summing individual scores.
5. Splitted the data into training and testing sets Define feature variables ('X') and the target variable ('y').Use **train_test_split** from scikit-learn to split the data into training and testing sets (80% training, 20% testing),Set a random seed ('random_state') for reproducibility.
6. Saved the preprocessed dataset to a new CSV file.

PDS-Assignment-1

StudentPerformance > Data_Preprocessing > data_preprocessing_students.ipynb

+ Code + Markdown | Run All Clear All Outputs | Outline ...

Select Kernel

```
import pandas as pd

# Load the dataset
data = pd.read_csv("../Data_Preprocessing/raw_student_data.csv")

# Check for missing values
missing_values = data.isnull().sum()
print("Missing Values:")
print(missing_values)

# Data Cleaning
# (No specific cleaning needed in this example)

# Data Exploration
summary_stats = data.describe()
print("\nSummary Statistics:")
print(summary_stats)

# Encoding Categorical Variables (One-Hot Encoding)
data = pd.get_dummies(data, columns=["gender", "race/ethnicity", "parental level of education", "lunch", "test preparation course"], drop_first=True)

# Feature Engineering (Calculate Total Score)
data['total score'] = data['math score'] + data['reading score'] + data['writing score']

# Data Visualization and further exploration (You can add your visualization code here)

# Train-Test Split (if you're building predictive models)
from sklearn.model_selection import train_test_split

X = data.drop('total score', axis=1)
y = data['total score']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Save the preprocessed data
data.to_csv("preprocessed_dataset.csv", index=False)
```

[1] ... Missing Values:
gender 0
race/ethnicity 0
parental level of education 0
lunch 0
test preparation course 0
math score 0
reading score 0
writing score 0
dtype: int64

Python

Ln 36, Col 53 Cell 1 of 1

PDS-Assignment-1

StudentPerformance > Data_Preprocessing > data_preprocessing_students.ipynb

+ Code + Markdown | Run All Clear All Outputs | Outline ...

Select Kernel

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Save the preprocessed data
data.to_csv("preprocessed_dataset.csv", index=False)
```

[1] ... Missing Values:
gender 0
race/ethnicity 0
parental level of education 0
lunch 0
test preparation course 0
math score 0
reading score 0
writing score 0
dtype: int64

	math score	reading score	writing score
count	1000.00000	1000.00000	1000.00000
mean	66.08900	69.16900	68.05400
std	15.16308	14.600192	15.195657
min	0.00000	17.00000	10.00000
25%	57.00000	59.00000	57.75000
50%	66.00000	70.00000	69.00000
75%	77.00000	79.00000	79.00000
max	100.00000	100.00000	100.00000

Python

Ln 36, Col 53 Cell 1 of 1

EXPLORER

- PDS-ASSIGNMENT-1
 - FrailtyData
 - Data_Preprocessing
 - cleaned_data.csv
 - data_preprocessing.ipynb
 - raw_data.csv
 - DataAnalysis
 - visualizations
 - correlation_heatmap.png
 - pairplot.png
 - eda.ipynb
 - Statistical_Analysis
 - reports
 - multiple_regression_report.txt
 - statistical_analysis.ipynb
 - StudentPerformance
 - Data_Preprocessing
 - data_processing_students.ipynb
 - preprocessed_dataset.csv
 - raw_student_data.csv
 - Visualizations
 - exam_scores_distribution.png
 - gender_distribution.png
 - pair_plot.png
 - scatter_plot_math_vs_reading.png
 - total_score_by_gender.png
 - Graph_Data.ipynb

OUTLINE

TIMELINE

16343756-sal-deepthi-enukonda

CSVLink Query Align Rainbow OFF

Col 1: gender Ln 1, Col 1 Spaces: 4 UTF-8 LF CSV

```

1 "gender","race/ethnicity","parental level of education","lunch","test preparation course","math score","reading score","writing score"
2 "female","group B","bachelor's degree","standard","none","72","72","74"
3 "female","group C","some college","standard","completed","69","98","88"
4 "female","group B","master's degree","standard","none","98","95","93"
5 "male","group A","associate's degree","free/reduced","none","47","57","44"
6 "male","group C","some college","standard","none","76","78","75"
7 "female","group B","associate's degree","standard","none","11","83","78"
8 "female","group B","some college","standard","completed","88","95","92"
9 "male","group B","some college","free/reduced","none","40","43","39"
10 "male","group D","high school","free/reduced","completed","64","64","67"
11 "female","group B","high school","free/reduced","none","38","60","58"
12 "male","group C","associate's degree","standard","none","58","54","52"
13 "male","group D","associate's degree","standard","none","40","52","43"
14 "female","group B","high school","standard","none","65","91","73"
15 "male","group A","some college","standard","completed","78","72","70"
16 "female","group A","master's degree","standard","none","58","53","58"
17 "female","group C","some high school","standard","none","69","75","78"
18 "male","group C","high school","standard","none","88","89","86"
19 "female","group B","some high school","free/reduced","none","18","32","28"
20 "male","group C","master's degree","free/reduced","completed","46","42","46"
21 "female","group C","associate's degree","free/reduced","none","55","55","55"
22 "male","group D","high school","standard","none", Col 5: test preparation course
23 "female","group B","some college","free/reduced","completed","65","75","70"
24 "male","group D","some college","standard","none","44","54","53"
25 "female","group C","some high school","standard","none","60","73","73"
26 "male","group D","bachelor's degree","free/reduced","completed","74","71","80"
27 "male","group A","master's degree","free/reduced","none","73","74","72"
28 "male","group B","some college","standard","none","69","54","55"
29 "female","group C","bachelor's degree","standard","none","67","69","75"
30 "male","group C","high school","standard","none","70","70","65"
31 "female","group D","master's degree","standard","none","62","70","75"
32 "female","group D","some college","standard","none","69","74","74"
33 "female","group B","some college","standard","none","63","65","61"
34 "female","group E","master's degree","free/reduced","none","56","72","65"
35 "male","group D","some college","standard","none","40","42","38"
36 "male","group E","some college","standard","none","97","87","82"
37 "male","group D","associate's degree","standard","completed","81","81","79"
38 "female","group D","associate's degree","standard","none","74","91","83"
39 "female","group D","some high school","free/reduced","none","58","64","59"
40 "female","group D","associate's degree","free/reduced","completed","75","90","88"
41 "male","group B","associate's degree","free/reduced","none","57","56","57"
42 "male","group C","associate's degree","free/reduced","none","55","61","54"
43 "female","group C","associate's degree","standard","none","58","73","68"
44 "female","group B","associate's degree","standard","none","53","58","65"
45 "male","group B","some college","free/reduced","completed","59","65","66"

```

EXPLORER

- PDS-ASSIGNMENT-1
 - FrailtyData
 - Data_Preprocessing
 - cleaned_data.csv
 - data_preprocessing.ipynb
 - raw_data.csv
 - DataAnalysis
 - visualizations
 - correlation_heatmap.png
 - pairplot.png
 - eda.ipynb
 - Statistical_Analysis
 - reports
 - multiple_regression_report.txt
 - statistical_analysis.ipynb
 - StudentPerformance
 - Data_Preprocessing
 - data_processing_students.ipynb
 - preprocessed_dataset.csv
 - raw_student_data.csv
 - Visualizations
 - exam_scores_distribution.png
 - gender_distribution.png
 - pair_plot.png
 - scatter_plot_math_vs_reading.png
 - total_score_by_gender.png
 - Graph_Data.ipynb

OUTLINE

TIMELINE

16343756-sal-deepthi-enukonda

CSVLink Query Align Rainbow OFF

Col 1: math score Ln 1, Col 1 Spaces: 4 UTF-8 LF CSV

```

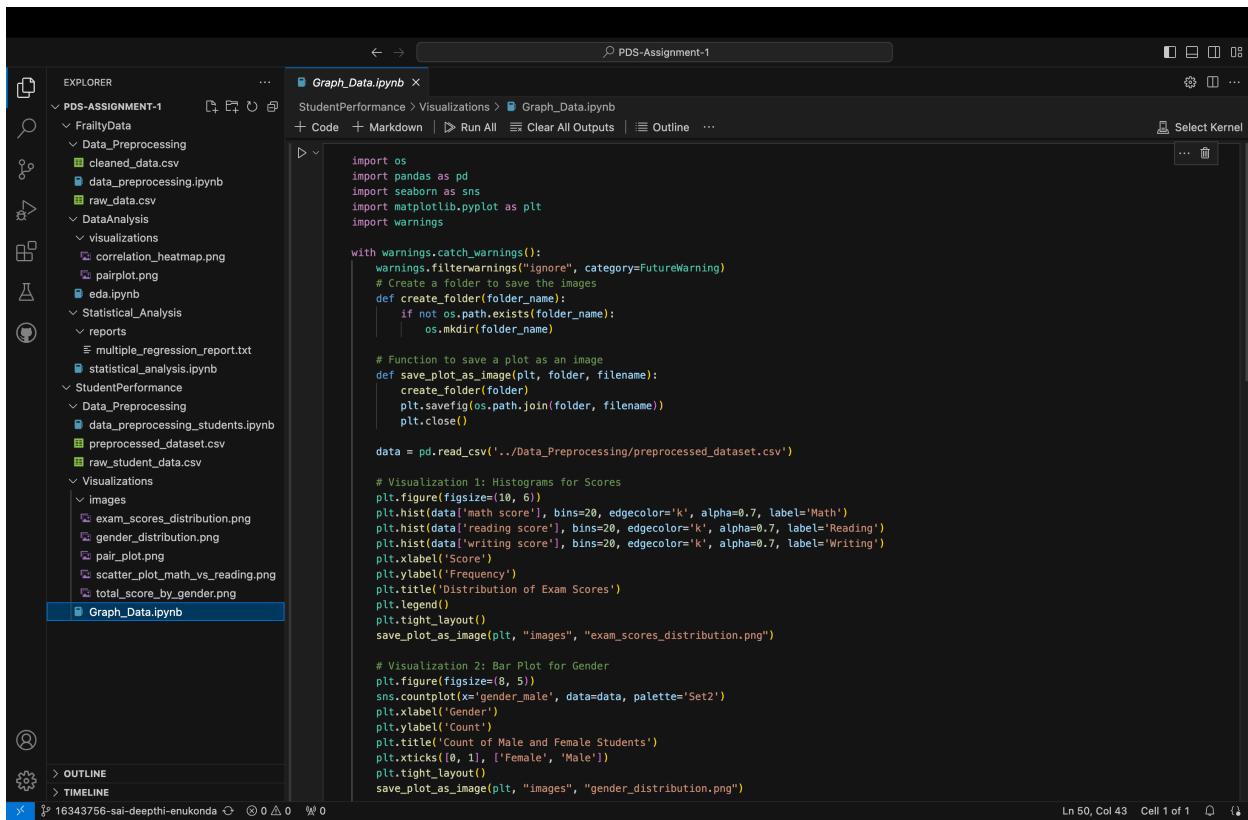
1 math score,reading score,writing score,gender,male,race/ethnicity_group B,race/ethnicity_group C,race/ethnicity_group D,race/ethnicity_group E
2 72,72,74,True,False,True,False,False,True,False,False,True,True,218
3 69,98,88,True,False,True,False,False,False,False,True,False,True,247
4 90,95,93,True,False,True,False,False,False,False,True,False,True,278
5 47,57,44,True,False,False,False,False,False,False,False,True,148
6 76,78,75,True,False,True,False,False,False,False,False,True,229
7 71,83,78,True,False,True,False,False,False,False,False,True,232
8 88,95,92,True,False,True,False,False,False,False,True,False,True,275
9 40,43,39,True,True,False,False,False,False,False,True,False,True,122
10 64,64,67,True,False,True,False,False,False,False,True,False,True,195
11 38,68,50,True,False,True,False,False,False,True,False,False,True,148
12 58,54,52,True,False,True,False,False,False,False,True,False,True,164
13 40,52,43,True,False,True,False,False,False,False,False,True,135
14 65,81,73,True,False,True,False,False,False,False,True,False,True,219
15 78,72,70,True,False,True,False,False,False,False,False,True,220
16 50,53,58,True,False,False,False,False,False,False,True,False,True,161
17 69,75,78,True,False,True,False,False,False,False,True,True,222
18 88,89,86,True,False,True,False,False,True,False,False,True,True,263
19 18,32,28,True,False,True,False,False,False,False,True,False,True,78
20 46,47,46,True,False,True,False,False,False,True,False,False,True,134
21 54,58,61,True,False,True,False,False,False,False,False,True,True,173
22 66,69,65,True,False,True,False,False,False,False,True,False,True,198
23 65,75,70,True,False,True,False,False,False,False,True,False,True,210
24 44,54,53,True,False,True,False,False,False,False,True,True,151
25 69,73,73,True,False,True,False,False,False,False,True,True,215
26 74,71,80,True,False,True,False,True,False,False,False,False,True,225
27 73,74,72,True,False,True,False,False,False,True,False,False,True,219
28 69,54,55,True,False,True,False,False,False,False,True,False,True,178
29 67,69,75,True,False,True,False,False,False,False,True,False,True,211
30 70,70,65,True,False,True,False,False,False,False,True,True,205
31 62,70,75,True,False,True,False,False,False,False,True,True,207
32 69,74,74,False,False,True,False,False,False,False,True,True,217
33 63,65,61,False,True,False,False,False,False,True,False,True,True,189
34 56,72,65,False,False,True,False,True,False,False,False,True,193
35 40,42,38,True,False,True,False,False,False,False,True,True,120
36 97,87,82,True,False,True,False,False,False,False,True,True,266
37 81,81,79,True,False,False,False,False,False,False,True,False,241
38 74,81,83,False,False,True,False,False,False,False,True,True,238
39 50,64,59,False,False,True,False,False,False,False,True,False,True,173
40 75,90,88,False,False,True,False,False,False,False,False,True,253
41 57,56,57,True,True,False,False,False,False,False,False,True,170
42 55,61,54,True,False,True,False,False,False,False,False,True,170
43 58,73,68,False,True,False,False,False,False,False,True,199
44 53,58,65,False,True,False,False,False,False,False,True,True,176
45 59,65,66,True,True,False,False,False,False,False,True,True,190

```

STEP2 : VISUALIZATIONS:

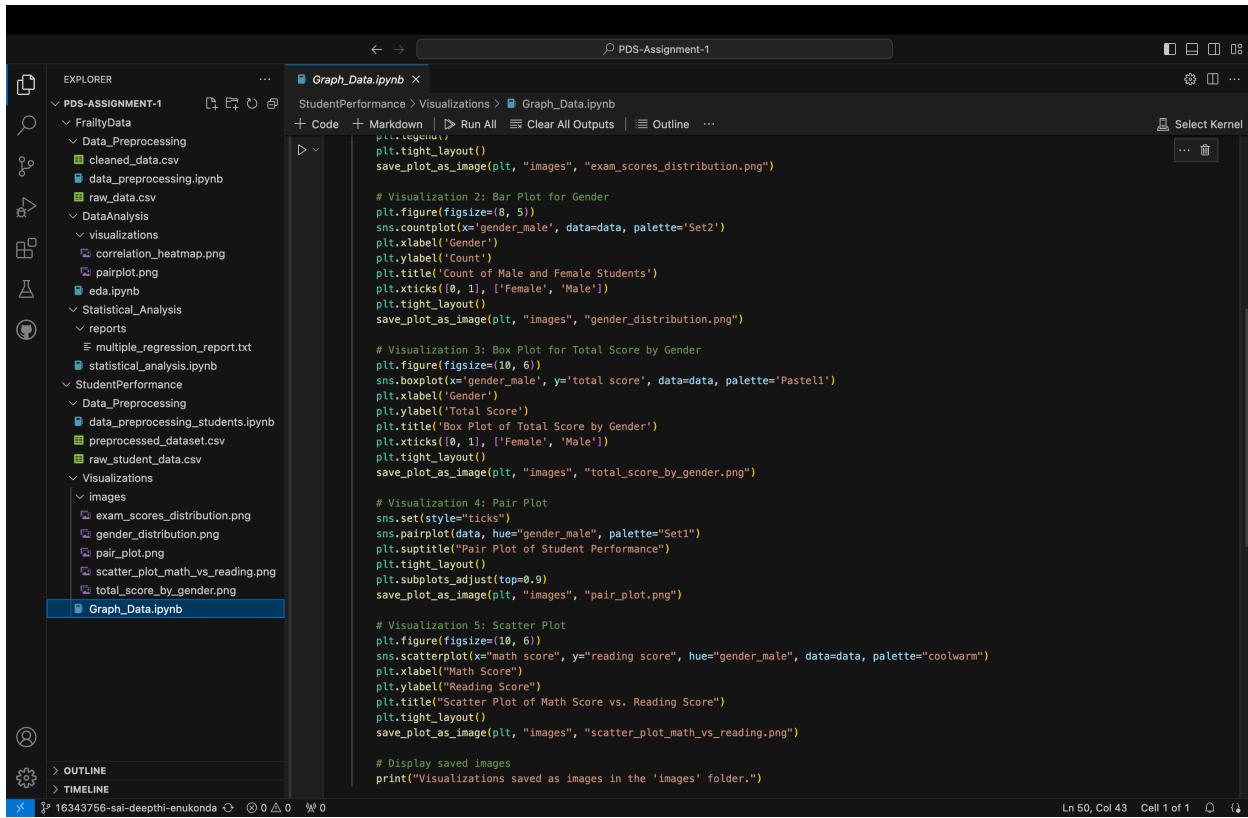
Performed 5 data visualization tasks on the student performance dataset by using these steps:

1. Imports necessary libraries for data manipulation and visualization.
2. Defines functions to create folders and save plots as images.
3. Loaded a preprocessed dataset from a CSV file.
4. Created and saved various data visualizations, including histograms, bar plots, box plots, pair plots, and scatter plots.
5. The saved images are organized in an 'images' folder.
6. A message is printed to indicate that the visualizations have been saved as images.



The screenshot shows a Jupyter Notebook interface with the following details:

- EXPLORER** sidebar: Shows a tree view of the project structure under "PDS-ASSIGNMENT-1". It includes "FrailtyData", "Data_Preprocessing", "DataAnalysis", "Statistical_Analysis", "StudentPerformance", and "Visualizations" sections. "Visualizations" contains several image files: "exam_scores_distribution.png", "gender_distribution.png", "pairplot.png", "scatter_plot_math_vs_reading.png", and "total_score_by_gender.png".
- Code Editor**: The active cell is "Graph_Data.ipynb". The code implements a series of data visualization tasks:
 - Imports os, pandas, seaborn, and matplotlib.pyplot.
 - Defines a function "create_folder" to create a folder for images if it doesn't exist.
 - Defines a function "save_plot_as_image" to save a plot to a specific folder.
 - Reads a CSV file "preprocessed_dataset.csv" into a pandas DataFrame "data".
 - Creates three histograms for "math score", "reading score", and "writing score" with alpha=0.7 and edgecolor='k'.
 - Creates a bar plot for gender distribution using sns.countplot(x='gender_male', data=data, palette='Set2').
- Output**: The notebook displays the generated plots as images in the "images" folder.



The screenshot shows a Jupyter Notebook interface with the following details:

- EXPLORER** sidebar on the left listing project files:
 - PDS-ASSIGNMENT-1
 - FrailtyData
 - Data_Preprocessing
 - DataAnalysis
 - EDA
 - Statistical_Analysis
 - StudentPerformance
 - Data_Preprocessing
 - Visualizations
 - images
- Graph_Data.ipynb** notebook tab is active.
- Code** pane contains Python code for data visualization:


```

plt.tight_layout()
save_plot_as_image(plt, "images", "exam_scores_distribution.png")

# Visualization 2: Bar Plot for Gender
plt.figure(figsize=(8, 5))
sns.countplot(x='gender_male', data=data, palette='Set2')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.title('Count of Male and Female Students')
plt.xticks([0, 1], ['Female', 'Male'])
plt.tight_layout()
save_plot_as_image(plt, "images", "gender_distribution.png")

# Visualization 3: Box Plot for Total Score by Gender
plt.figure(figsize=(10, 6))
sns.boxplot(x='gender_male', y='total score', data=data, palette='Pastel1')
plt.xlabel('Gender')
plt.ylabel('Total Score')
plt.title('Box Plot of Total Score by Gender')
plt.xticks([0, 1], ['Female', 'Male'])
plt.tight_layout()
save_plot_as_image(plt, "images", "total_score_by_gender.png")

# Visualization 4: Pair Plot
sns.set(style="ticks")
sns.pairplot(data, hue="gender_male", palette="Set1")
plt.subtitle("Pair Plot of Student Performance")
plt.tight_layout()
plt.subplots_adjust(top=0.9)
save_plot_as_image(plt, "images", "pair_plot.png")

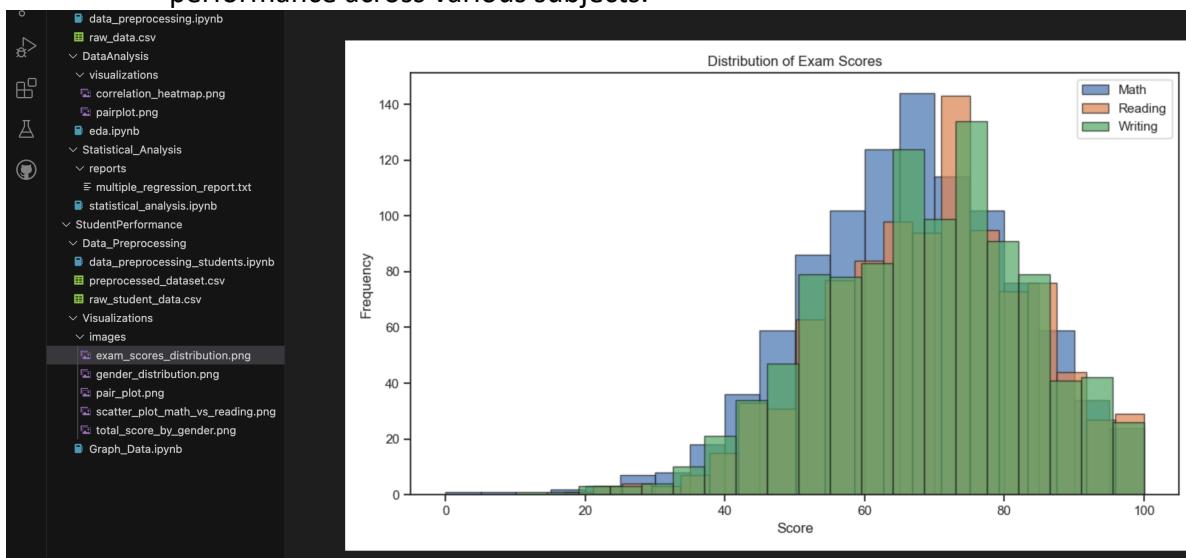
# Visualization 5: Scatter Plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x="math score", y="reading score", hue="gender_male", data=data, palette="coolwarm")
plt.xlabel("Math Score")
plt.ylabel("Reading Score")
plt.title("Scatter Plot of Math Score vs. Reading Score")
plt.tight_layout()
save_plot_as_image(plt, "images", "scatter_plot_math_vs_reading.png")

# Display saved images
print("Visualizations saved as images in the 'images' folder.")

```
- OUTPUT** and **TIMELINE** sections are visible at the bottom.
- Bottom status bar shows: Ln 50, Col 43 Cell 1 of 1

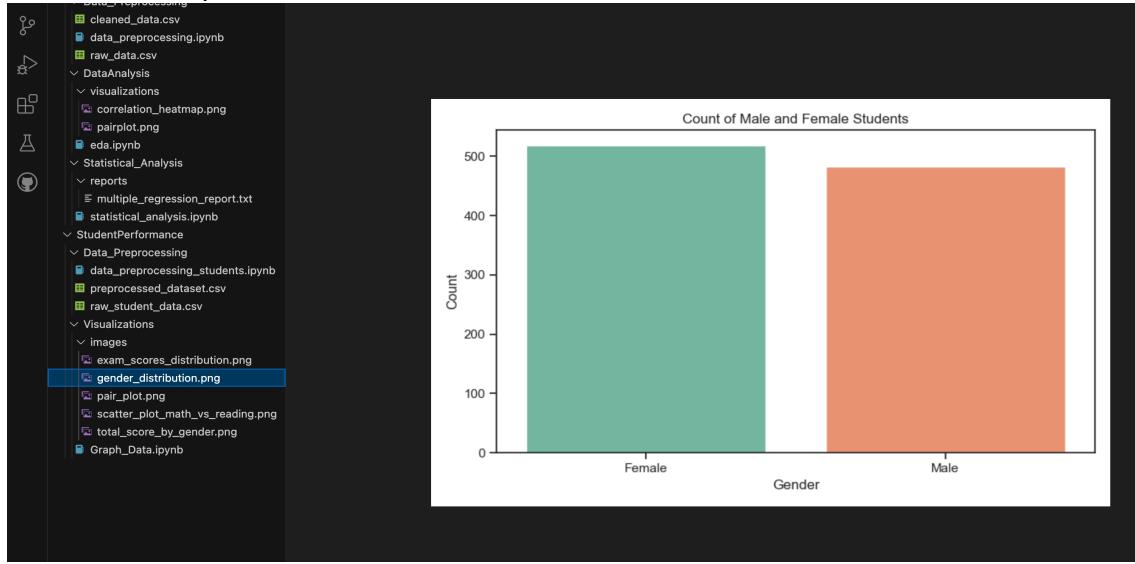
1. Visualization 1: Histograms for Scores:

- Analysis Made Easier: This visualization simplifies the examination of score distributions (math, reading, and writing) by presenting them as histograms. It facilitates the assessment of score patterns, such as whether they follow a normal distribution or exhibit skewness. This aids in gaining insights into student performance across various subjects.



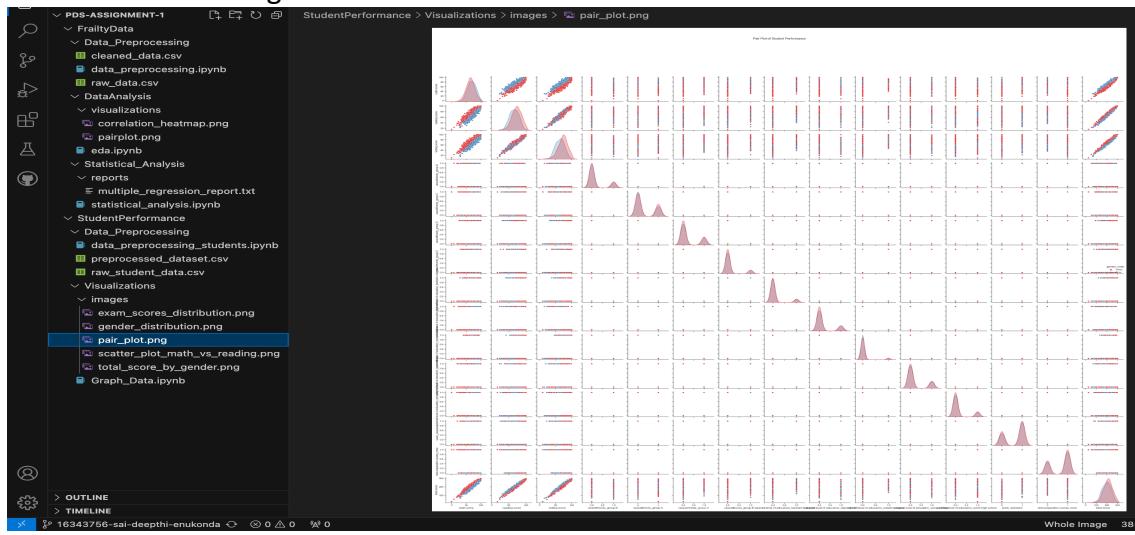
2. Visualization 2: Bar Plot for Gender:

- Analysis Made Easier: By using a bar plot, it becomes straightforward to compare the counts of male and female students. This visual representation clarifies which gender is predominant in the dataset, making it convenient for gender-related analyses or considerations.



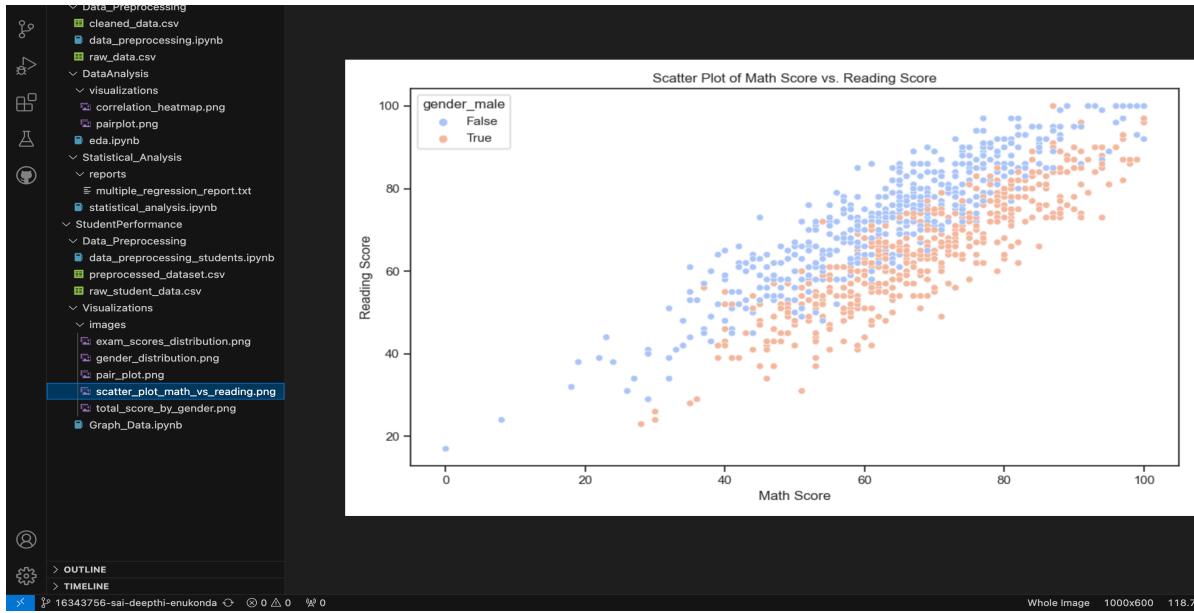
3. Visualization 3: Box Plot for Total Score by Gender:

- Analysis Made Easier: The box plot offers a concise visual summary of total score distributions for each gender (male and female). It simplifies the identification of disparities in score distributions, including the presence of outliers and quartiles, between genders.



4. Visualization 4: Pair Plot:

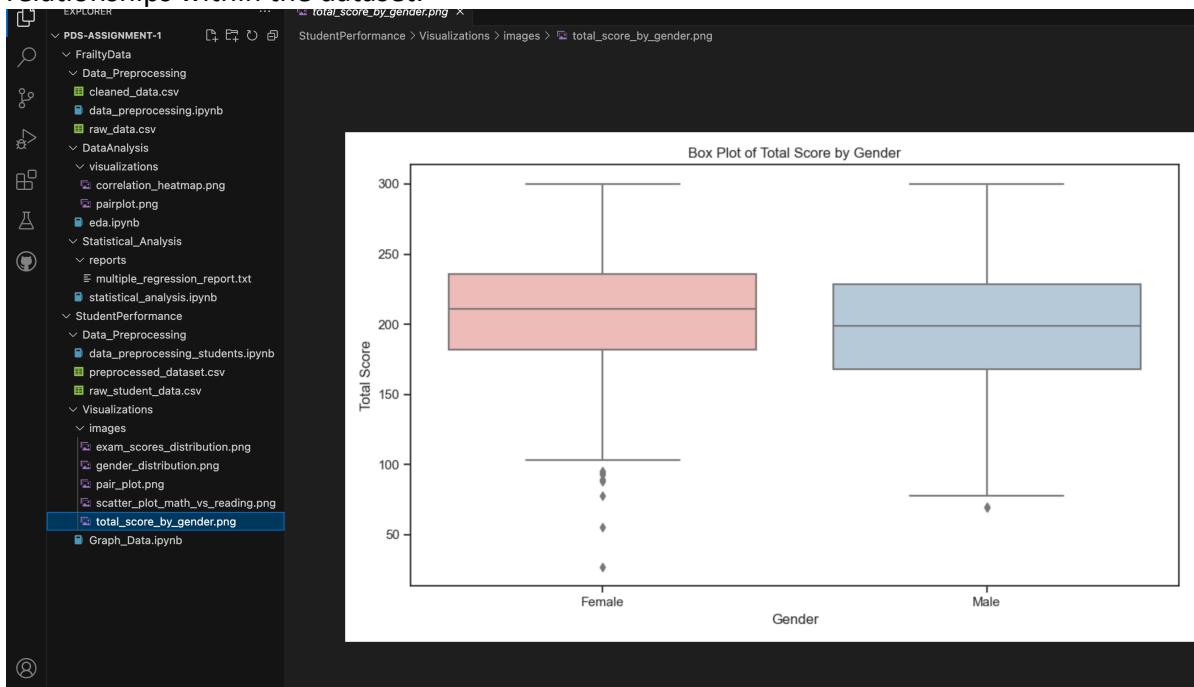
- Analysis Made Easier: This comprehensive visualization simplifies the exploration of relationships between various variables. It streamlines the process of identifying correlations and patterns across multiple variables while also allowing for an examination of how gender relates to these patterns.



5. Visualization 5: Scatter Plot:

- Analysis Made Easier: The scatter plot illustrates the connection between math scores and reading scores, with data points categorized by gender. It aids in visualizing the relationship between these two specific variables and highlights any gender-based differences in performance within these subjects.

In summary, each visualization in the code serves a distinct purpose and enhances different aspects of data analysis, making it more accessible and insightful. These visualizations provide valuable insights into score distributions, gender-related trends, correlations, and variable relationships within the dataset.



Submission:

<https://github.com/deepthi978/PDS-Assignment-1>

Repository Details

The screenshot shows the GitHub 'Create a new repository' interface. The repository name is 'PDS-Assignment-1'. It is set to be public. Other settings include a README file and no .gitignore template.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner * deepthi978 / **Repository name *** PDS-Assignment-1
PDS-Assignment-1 is available.

Great repository names are short and memorable. Need inspiration? How about [supreme-memory](#) ?

Description (optional)

Public Anyone on the internet can see this repository. You choose who can commit.
Private You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license

License: None