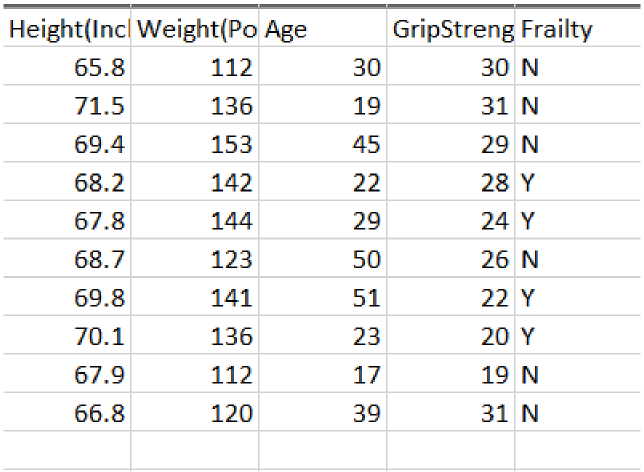
# Assignment 1

## Vishal Reddy Bomma - 16340457

# Frailty Analysis

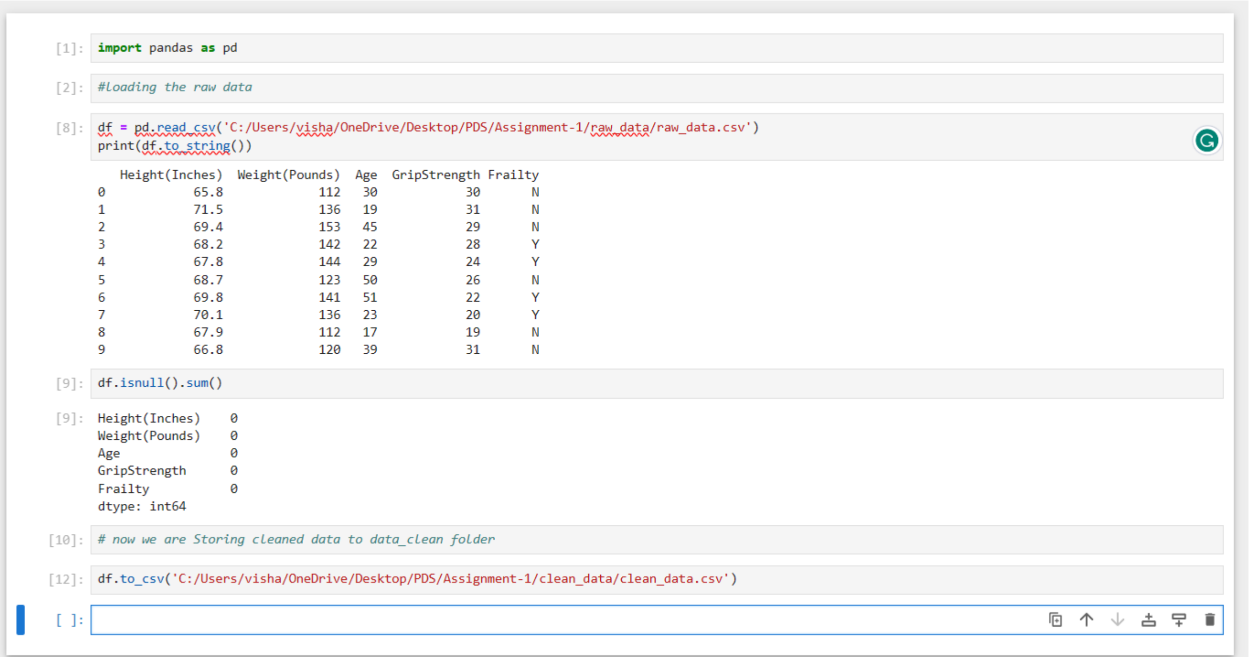
## Data Collection

- **The given data is inputted in excel sheet and saved it as a .csv file**



| Height(Incl | Weight(Po | Age | GripStreng | 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 |
| | | | | |

## Data Processing



```
[1]: import pandas as pd

[2]: #loading the raw data

[8]: df = pd.read_csv('C:/Users/visha/OneDrive/Desktop/PDS/Assignment-1/raw_data/raw_data.csv')
     print(df.to_string())

        Height(Inches)  Weight(Pounds)  Age  GripStrength Frailty
     0            65.8             112   30            30       N
     1            71.5             136   19            31       N
     2            69.4             153   45            29       N
     3            68.2             142   22            28       Y
     4            67.8             144   29            24       Y
     5            68.7             123   50            26       N
     6            69.8             141   51            22       Y
     7            70.1             136   23            20       Y
     8            67.9             112   17            19       N
     9            66.8             120   39            31       N

[9]: df.isnull().sum()

[9]: Height(Inches)    0
     Weight(Pounds)    0
     Age               0
     GripStrength      0
     Frailty           0
     dtype: int64

[10]: # now we are Storing cleaned data to data_clean folder

[12]: df.to_csv('C:/Users/visha/OneDrive/Desktop/PDS/Assignment-1/clean_data/clean_data.csv')

[ ]:
```

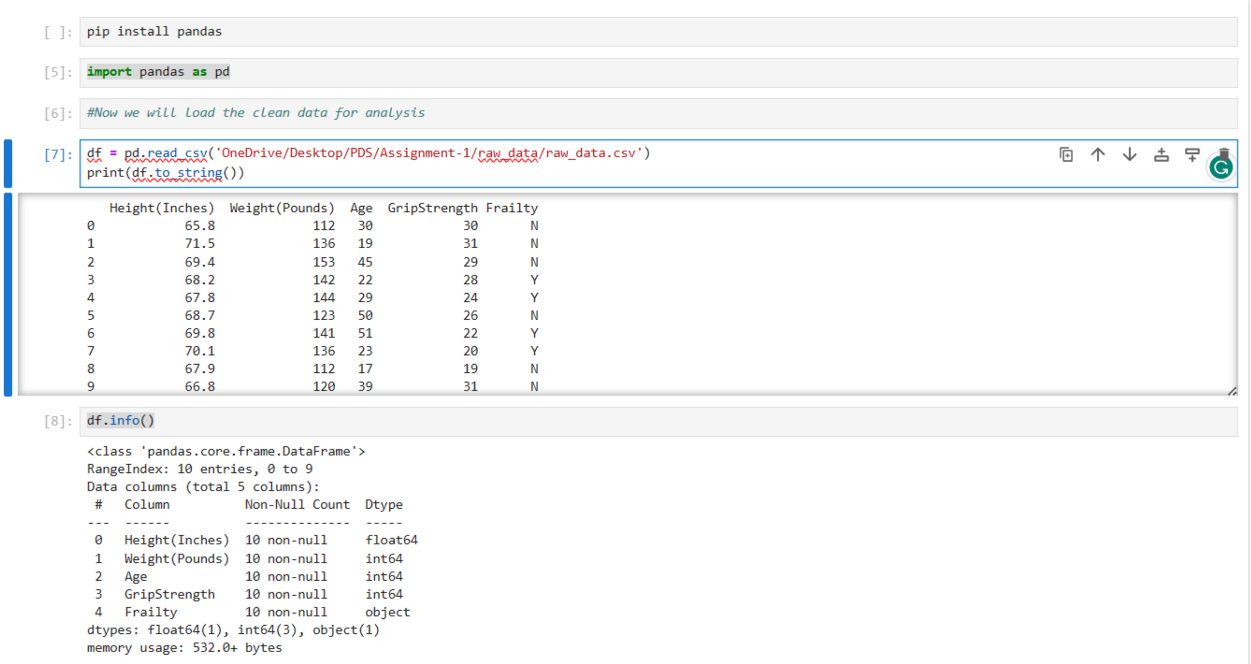
- We use df.isnull().sum() to check for missing values in each column of the DataFrame. This is a crucial data processing step as handling missing data is essential for accurate analysis. In this case, we find that there are no missing values in any of the columns.

## Data Saving:

- Finally, you save the cleaned data to a new CSV file using df.to_csv(). While this step doesn't involve extensive processing, it's part of the data preparation process. You're essentially saving the data in its current cleaned state for future analysis.

## Data Analysis

- First I imported all the required packages such as seaborn , matplotlib etc
- In this step, you perform data visualization tasks using the Matplotlib and Seaborn libraries.
- You create histograms (sns.distplot()) to visualize the distribution of 'Age' and 'GripStrength' columns. The histograms are saved as image files.
- We also create scatter plots (df.plot.scatter()) to explore relationships between 'Age' vs. 'GripStrength,' 'Weight(Pounds)' vs. 'GripStrength,' and 'Height(Inches)' vs. 'GripStrength.' These scatter plots are also saved as image files.



```
[ ]: pip install pandas

[5]: import pandas as pd

[6]: #Now we will load the clean data for analysis

[7]: df = pd.read_csv('OneDrive/Desktop/PDS/Assignment-1/raw_data/raw_data.csv')
     print(df.to_string())

   Height(Inches)  Weight(Pounds)  Age  GripStrength Frailty
0            65.8             112   30            30       N
1            71.5             136   19            31       N
2            69.4             153   45            29       N
3            68.2             142   22            28       Y
4            67.8             144   29            24       Y
5            68.7             123   50            26       N
6            69.8             141   51            22       Y
7            70.1             136   23            20       Y
8            67.9             112   17            19       N
9            66.8             120   39            31       N

[8]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Height(Inches)  10 non-null     float64
 1   Weight(Pounds)  10 non-null     int64  
 2   Age             10 non-null     int64  
 3   GripStrength    10 non-null     int64  
 4   Frailty         10 non-null     object 
dtypes: float64(1), int64(3), object(1)
memory usage: 532.0+ bytes
```

[17]:

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

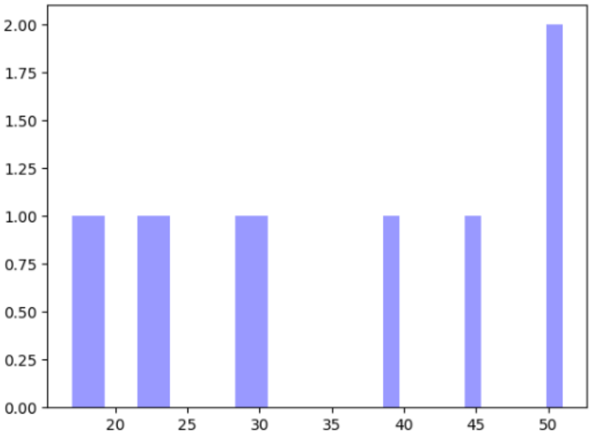
[18]:

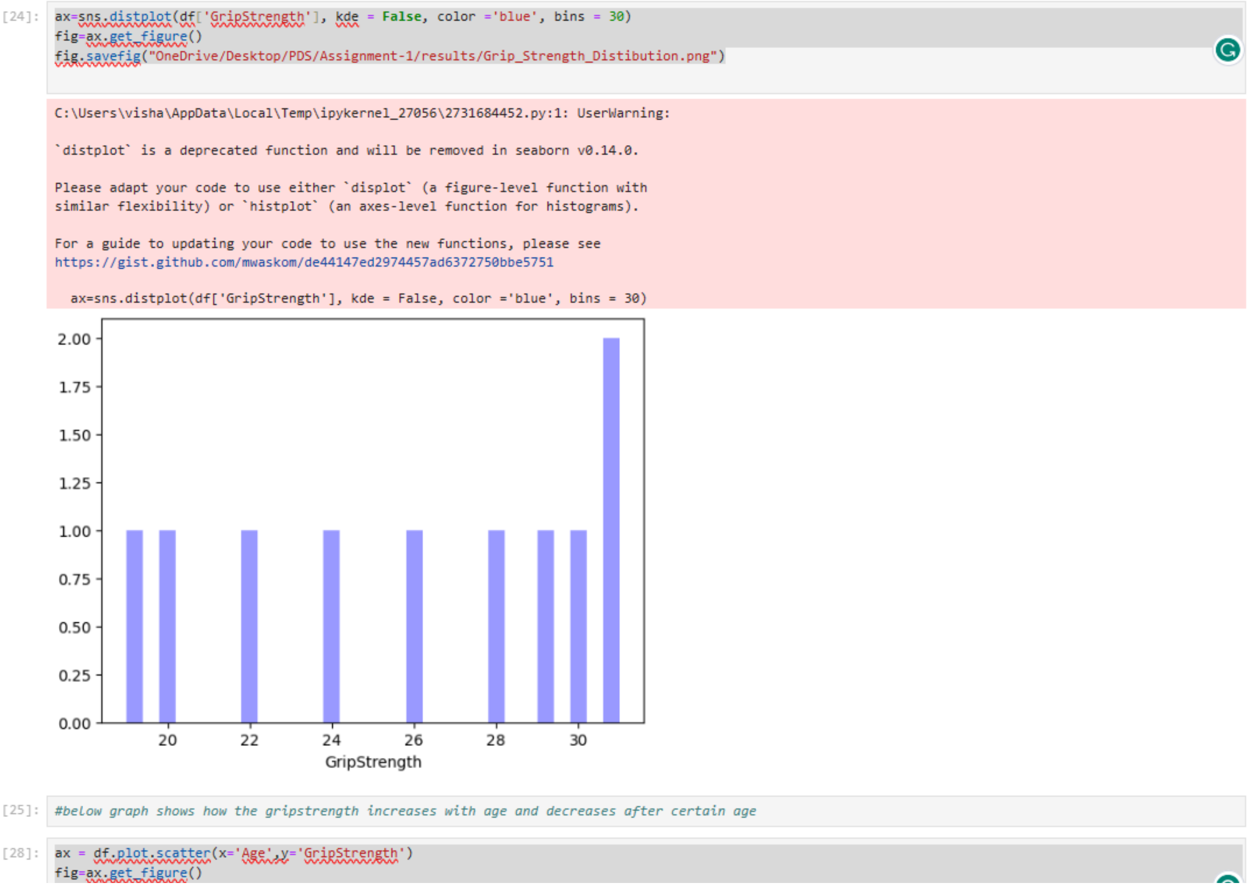
```
# below graph shows the distribution of Age in the given data
```

[22]:

```
ax=sns.distplot(df['Age'], kde = False, color ='blue', bins = 30)
fig=ax.get_figure()
fig.savefig("OneDrive/Desktop/PDS/Assignment-1/results/Age_Distibution.png")
```

```
C:\Users\visha\AppData\Local\Temp\ipykernel_27056\1172856943.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  ax=sns.distplot(df['Age'], kde = False, color ='blue', bins = 30)
```



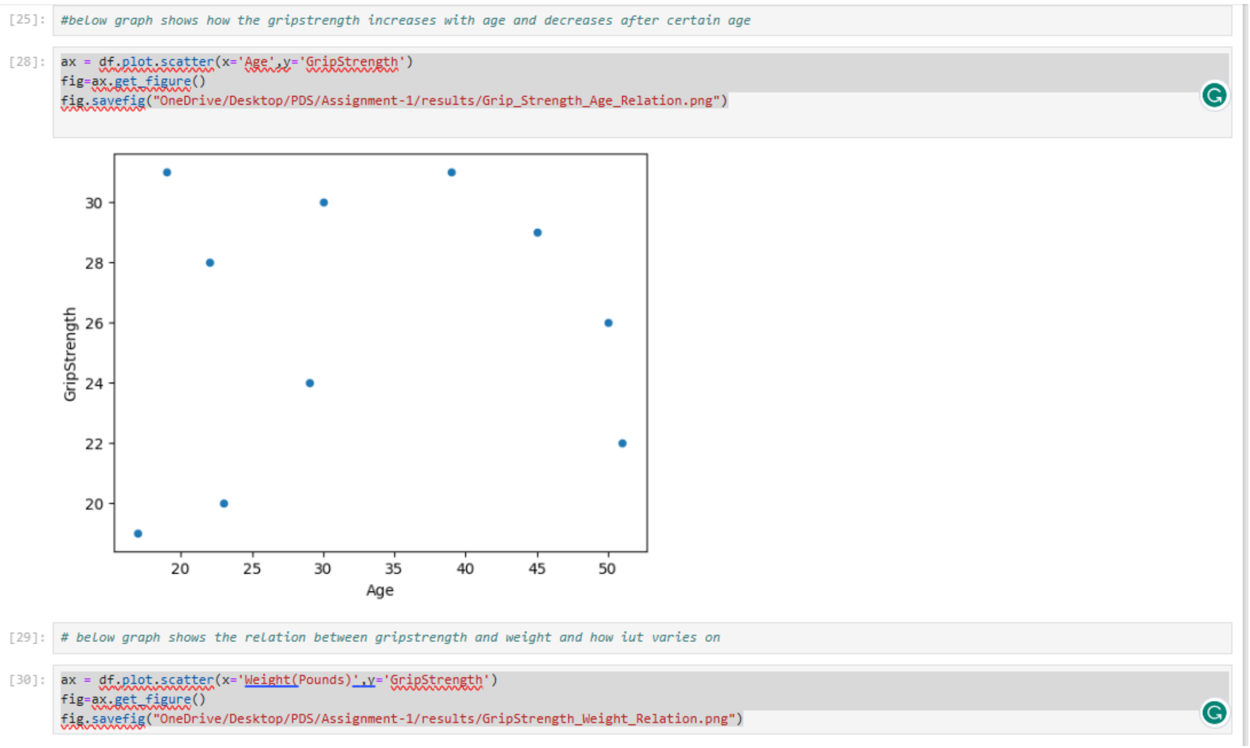


```
[24]: ax=sns.distplot(df['GripStrength'], kde = False, color ='blue', bins = 30)
      fig=ax.get_figure()
      fig.savefig("OneDrive/Desktop/PDS/Assignment-1/results/Grip_Strength_Distibution.png")
```

```
C:\Users\visha\AppData\Local\Temp\ipykernel_27056\2731684452.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  ax=sns.distplot(df['GripStrength'], kde = False, color ='blue', bins = 30)
```

```
[25]: #below graph shows how the gripstrength increases with age and decreases after certain age
```

```
[28]: ax = df.plot.scatter(x='Age',y='GripStrength')
      fig=ax.get_figure()
```



```
[25]: #below graph shows how the gripstrength increases with age and decreases after certain age
```

```
[28]: ax = df.plot.scatter(x='Age',y='GripStrength')
      fig=ax.get_figure()
      fig.savefig("OneDrive/Desktop/PDS/Assignment-1/results/Grip_Strength_Age_Relation.png")
```

```
[29]: # below graph shows the relation between gripstrength and weight and how iut varies on
```

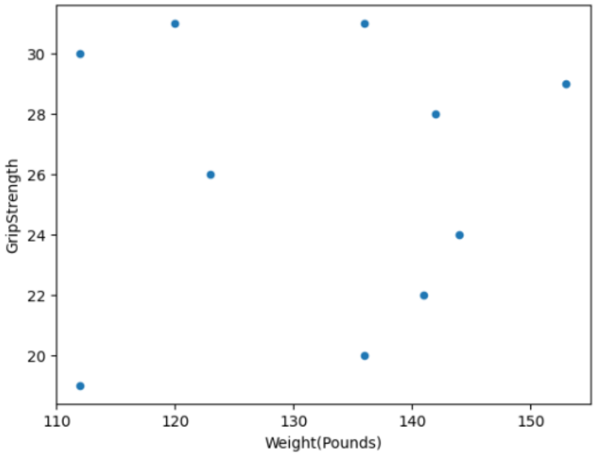
```
[30]: ax = df.plot.scatter(x='Weight(Pounds)',y='GripStrength')
      fig=ax.get_figure()
      fig.savefig("OneDrive/Desktop/PDS/Assignment-1/results/GripStrength_Weight_Relation.png")
```

[29]:

```
# below graph shows the relation between gripstrength and weight and how iut varies on
```

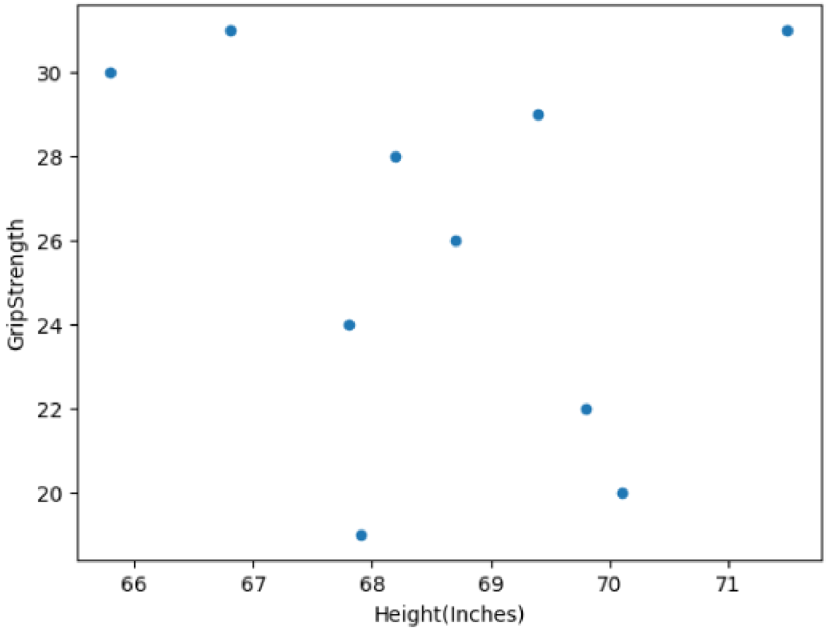
[30]:

```
ax = df.plot.scatter(x='Weight(Pounds)',y='GripStrength')
fig=ax.get_figure()
fig.savefig("OneDrive/Desktop/PDS/Assignment-1/results/GripStrength_Weight_Relation.png")
```



[31]:

```
ax = df.plot.scatter(x='Height(Inches)',y='GripStrength')
fig=ax.get_figure()
fig.savefig("OneDrive/Desktop/PDS/Assignment-1/results/GripStrength_Height_Relation.png")
```

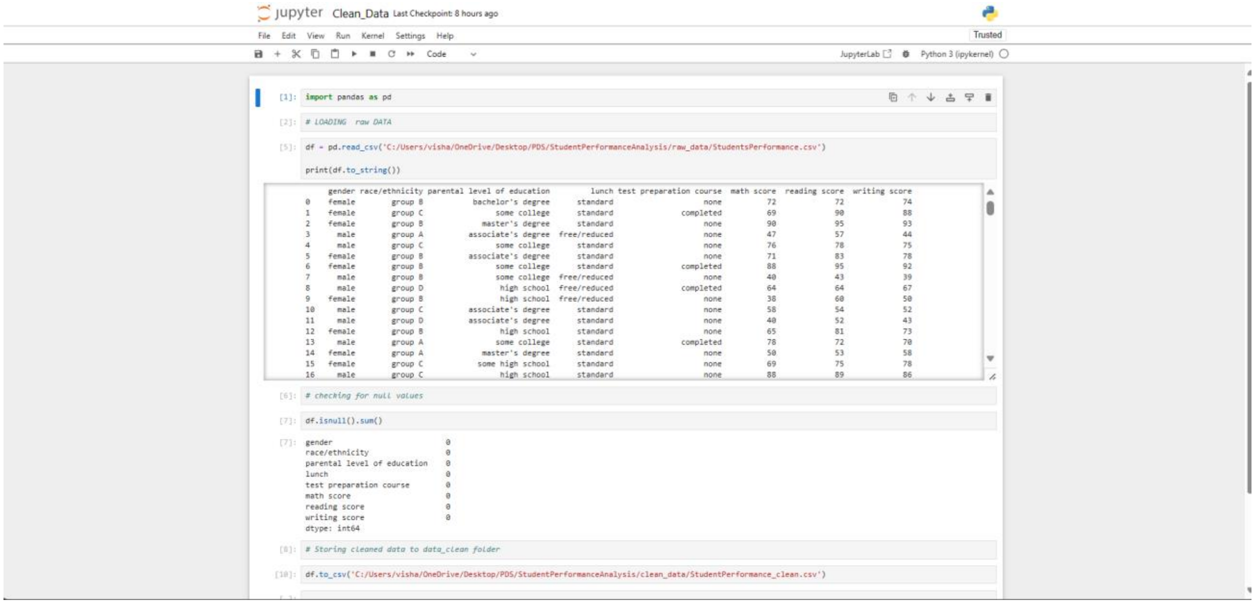


]:

# StudentPerformanceAnalysis

## Data Collection

- In this section, we load the raw data from the "StudentsPerformance.csv" file, which likely contains student performance information.


jupyter Clean_Data Last Checkpoint: 8 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

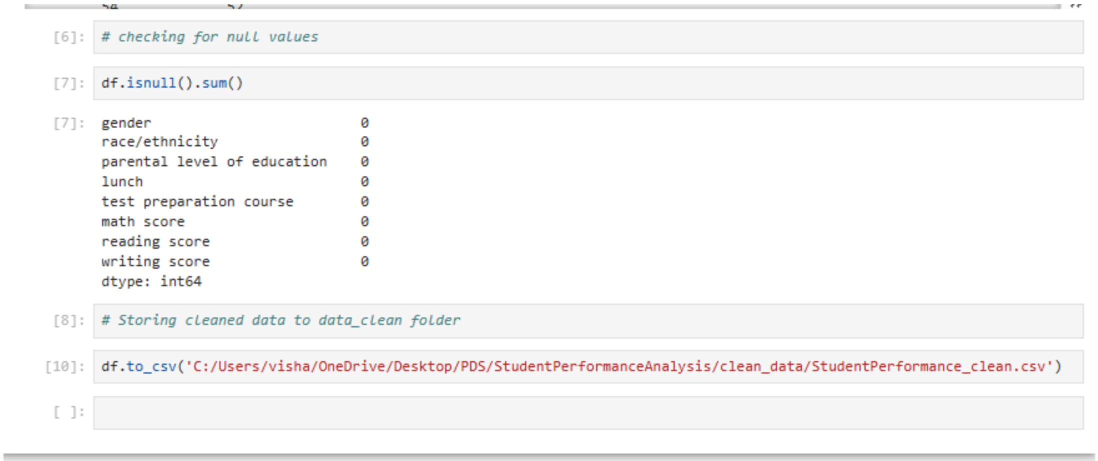
```
[1]: import pandas as pd

[2]: # LOADING  raw DATA

[5]: df = pd.read_csv('C:/Users/visha/OneDrive/Desktop/PDS/StudentPerformanceAnalysis/raw_data/StudentsPerformance.csv')

     print(df.to_string())
```

```
   gender race/ethnicity parental level of education        lunch test preparation course  math score  reading score  writing score
0  female        group B           bachelor's degree     standard                    none          72             72             74
1  female        group C                some college     standard               completed          69             90             88
2  female        group B             master's degree     standard                    none          90             95             93
3    male        group A          associate's degree free/reduced                    none          47             57             44
4    male        group C                some college     standard                    none          76             78             75
5  female        group B          associate's degree     standard                    none          71             83             78
6  female        group B                some college     standard               completed          88             95             92
7    male        group B                some college free/reduced                    none          40             43             39
8    male        group D                 high school free/reduced               completed          64             64             67
9  female        group B                 high school free/reduced                    none          38             60             50
10   male        group C          associate's degree     standard                    none          58             54             52
11   male        group D          associate's degree     standard                    none          40             52             43
12 female        group B                 high school     standard                    none          65             81             73
13   male        group A                some college     standard               completed          78             72             70
14 female        group A             master's degree     standard                    none          50             53             58
15 female        group C            some high school     standard                    none          69             75             78
16   male        group C                 high school     standard                    none          88             89             86
```

```
[6]: # checking for null values

[7]: df.isnull().sum()

[7]: 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

[8]: # Storing cleaned data to data_clean folder

[10]: df.to_csv('C:/Users/visha/OneDrive/Desktop/PDS/StudentPerformanceAnalysis/clean_data/StudentPerformance_clean.csv')
```

## Step 2: Data Preprocessing

- We use df.isnull().sum() to check for missing values in each column of the DataFrame. This is a crucial data processing step as handling missing data is essential for accurate analysis. In this case, we find that there are no missing values in any of the columns.



```
[6]: # checking for null values

[7]: df.isnull().sum()

[7]: 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

[8]: # Storing cleaned data to data_clean folder

[10]: df.to_csv('C:/Users/visha/OneDrive/Desktop/PDS/StudentPerformanceAnalysis/clean_data/StudentPerformance_clean.csv')

[ ]:
```

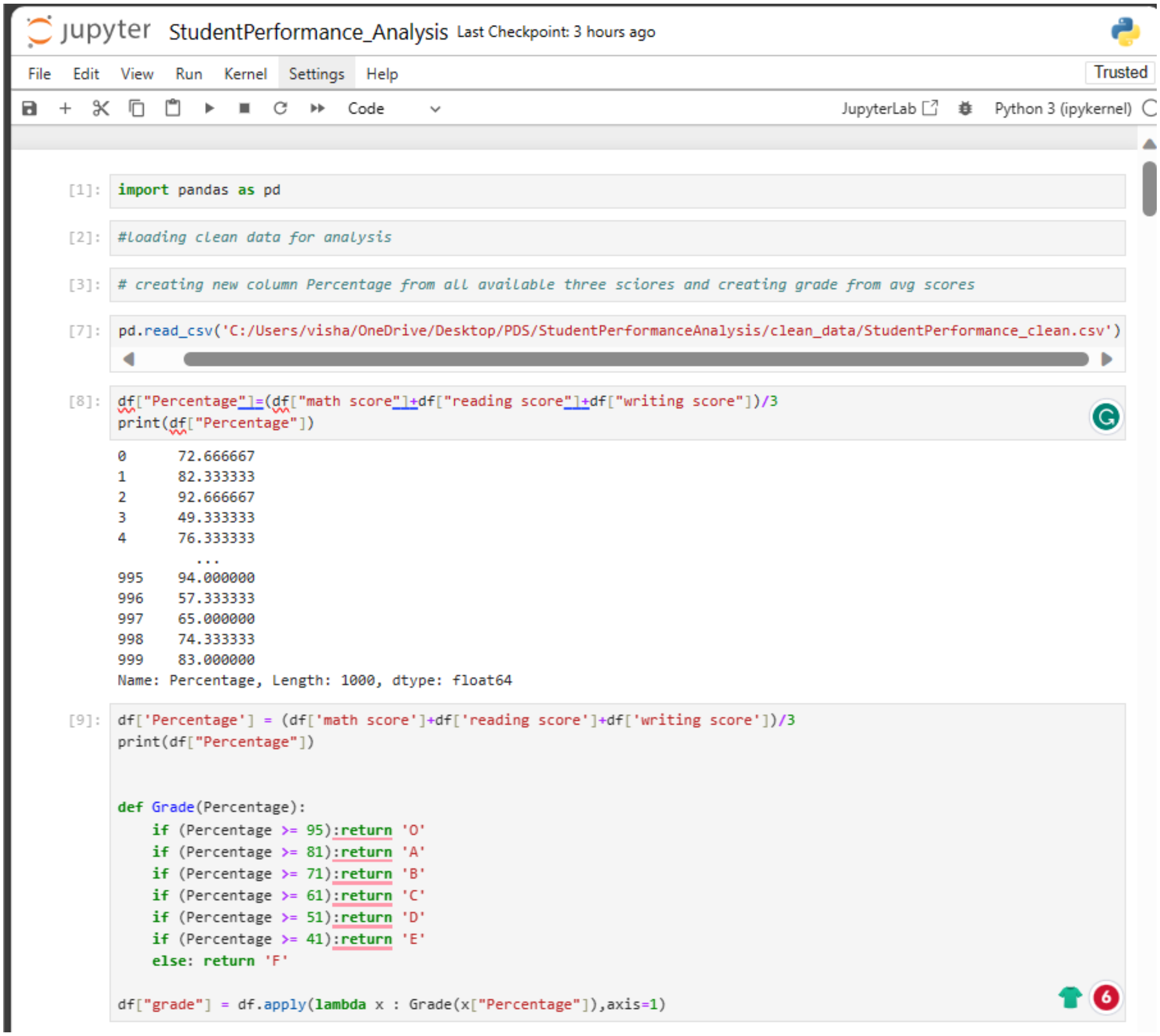
- Then, we print the entire DataFrame to inspect the data.
- print(df.to_string())
- we check for missing values in the DataFrame and print the count of missing values for each column.
- df.isnull().sum()
- Finally, we store the cleaned data into a new CSV file in the "clean_data" folder.
- df.to_csv('C:/Users/visha/OneDrive/Desktop/PDS/StudentPerformanceAnalysis/clean_data/StudentPerformance_clean.csv')

## Data Analysis (studentperformance_analysis.py):

- In this section, you load the cleaned data for analysis.
- df = pd.read_csv('C:/Users/visha/OneDrive/Desktop/PDS/StudentPerformanceAnalysis/clean_data/StudentPerformance_clean.csv')
- we calculate the percentage score for each student based on their math, reading, and writing scores and create a new column named "Percentage."

- df["Percentage"] = (df["math score"] + df["reading score"] + df["writing score"]) / 3

- we define a function called "Grade" that assigns a grade based on the percentage score. This function is then applied to each row in the DataFrame to create a new column named "grade."

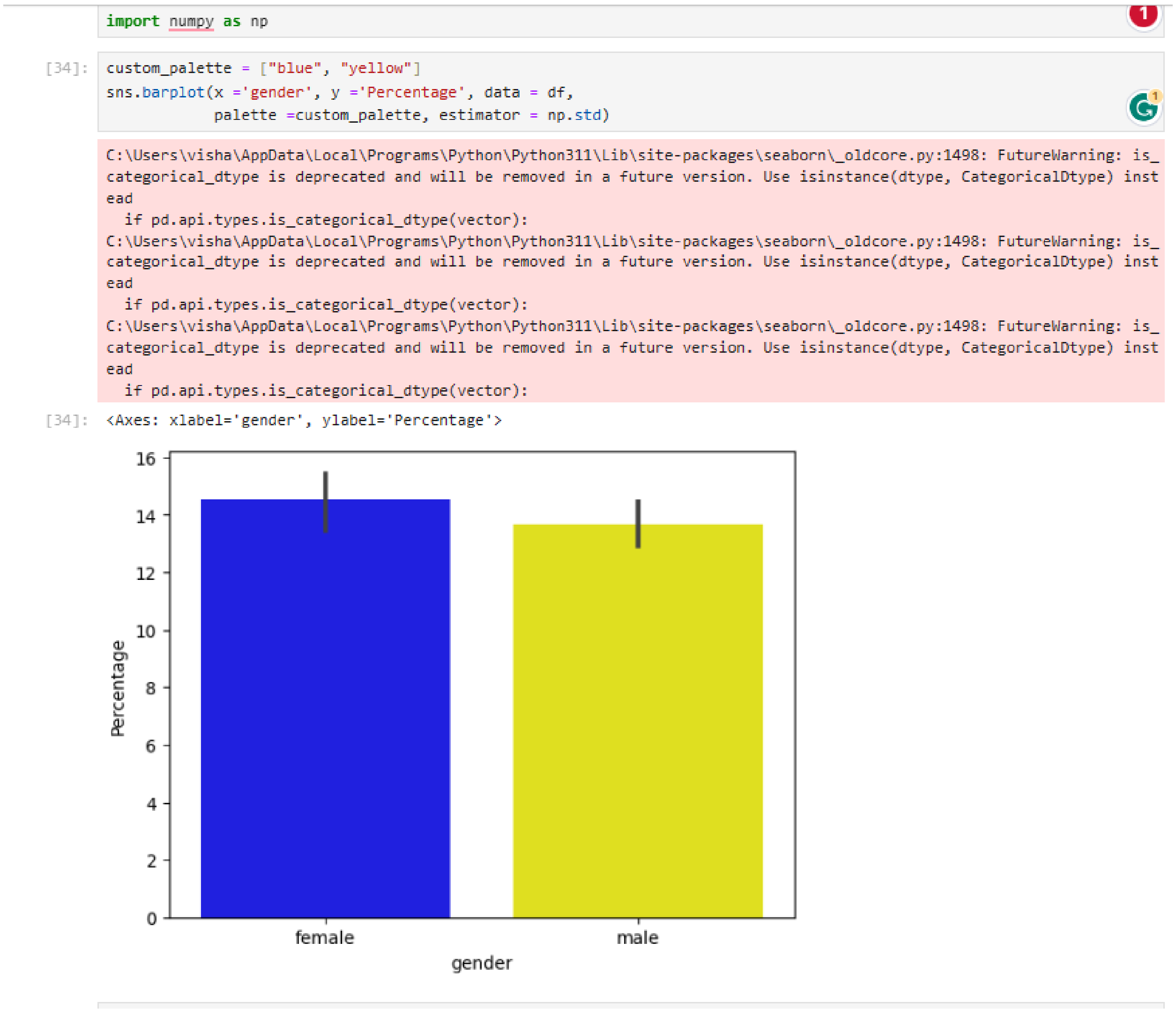

jupyter
StudentPerformance_Analysis Last Checkpoint: 3 hours ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

```
[1]: import pandas as pd

[2]: #loading clean data for analysis

[3]: # creating new column Percentage from all available three sciores and creating grade from avg scores

[7]: pd.read_csv('C:/Users/visha/OneDrive/Desktop/PDS/StudentPerformanceAnalysis/clean_data/StudentPerformance_clean.csv')

[8]: df["Percentage"]=(df["math score"]+df["reading score"]+df["writing score"])/3
     print(df["Percentage"])

0      72.666667
1      82.333333
2      92.666667
3      49.333333
4      76.333333
         ...
995    94.000000
996    57.333333
997    65.000000
998    74.333333
999    83.000000
Name: Percentage, Length: 1000, dtype: float64

[9]: df['Percentage'] = (df['math score']+df['reading score']+df['writing score'])/3
     print(df["Percentage"])


     def Grade(Percentage):
         if (Percentage >= 95):return 'O'
         if (Percentage >= 81):return 'A'
         if (Percentage >= 71):return 'B'
         if (Percentage >= 61):return 'C'
         if (Percentage >= 51):return 'D'
         if (Percentage >= 41):return 'E'
         else: return 'F'

     df["grade"] = df.apply(lambda x : Grade(x["Percentage"]),axis=1)
```

```
0      72.666667
1      82.333333
2      92.666667
3      49.333333
4      76.333333
         ...
995    94.000000
996    57.333333
997    65.000000
998    74.333333
999    83.000000
Name: Percentage, Length: 1000, dtype: float64
```

```
print(df)
```

```
     Unnamed: 0  gender race/ethnicity parental level of education  \
0             0  female        group B           bachelor's degree
1             1  female        group C                some college
2             2  female        group B             master's degree
3             3    male        group A          associate's degree
4             4    male        group C                some college
..          ...     ...            ...                         ...
995         995  female        group E             master's degree
996         996    male        group C                 high school
997         997  female        group C                 high school
998         998  female        group D                some college
999         999  female        group D                some college

            lunch test preparation course  math score  reading score  \
0        standard                    none          72             72
1        standard               completed          69             90
2        standard                    none          90             95
3    free/reduced                    none          47             57
4        standard                    none          76             78
..            ...                     ...         ...            ...
995      standard               completed          88             99
996  free/reduced                    none          62             55
997  free/reduced               completed          59             71
998      standard               completed          68             78
999  free/reduced                    none          77             86

     writing score  Percentage grade
0               74   72.666667     B
1               88   82.333333     A
2               93   92.666667     A
3               44   49.333333     E
4               75   76.333333     B
..             ...         ...   ...
995             95   94.000000     A
```

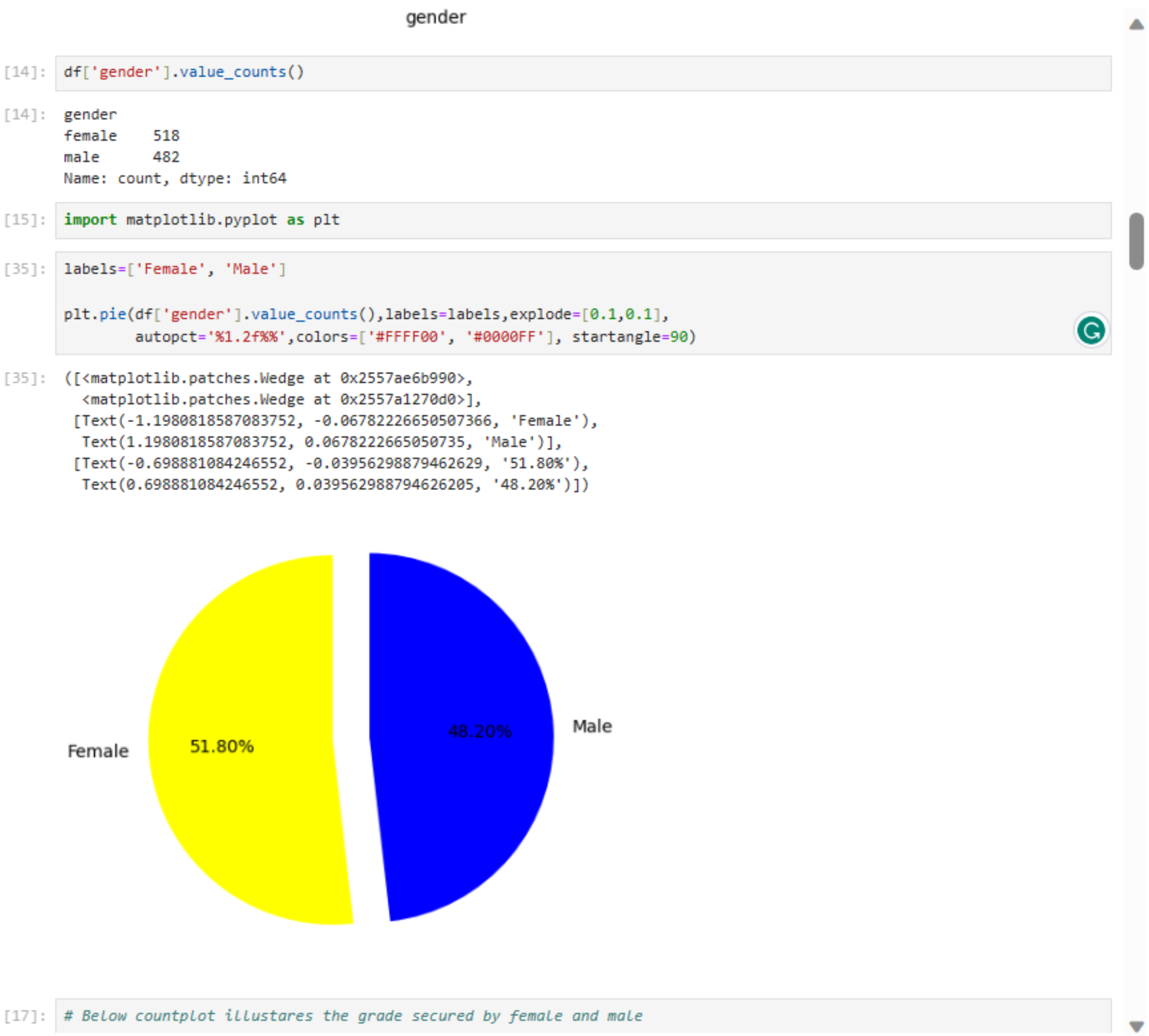


```
import numpy as np
```

```
[34]: custom_palette = ["blue", "yellow"]
      sns.barplot(x ='gender', y ='Percentage', data = df,
                  palette =custom_palette, estimator = np.std)
```

```
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
```

```
[34]: <Axes: xlabel='gender', ylabel='Percentage'>
```



```
[14]: df['gender'].value_counts()
```

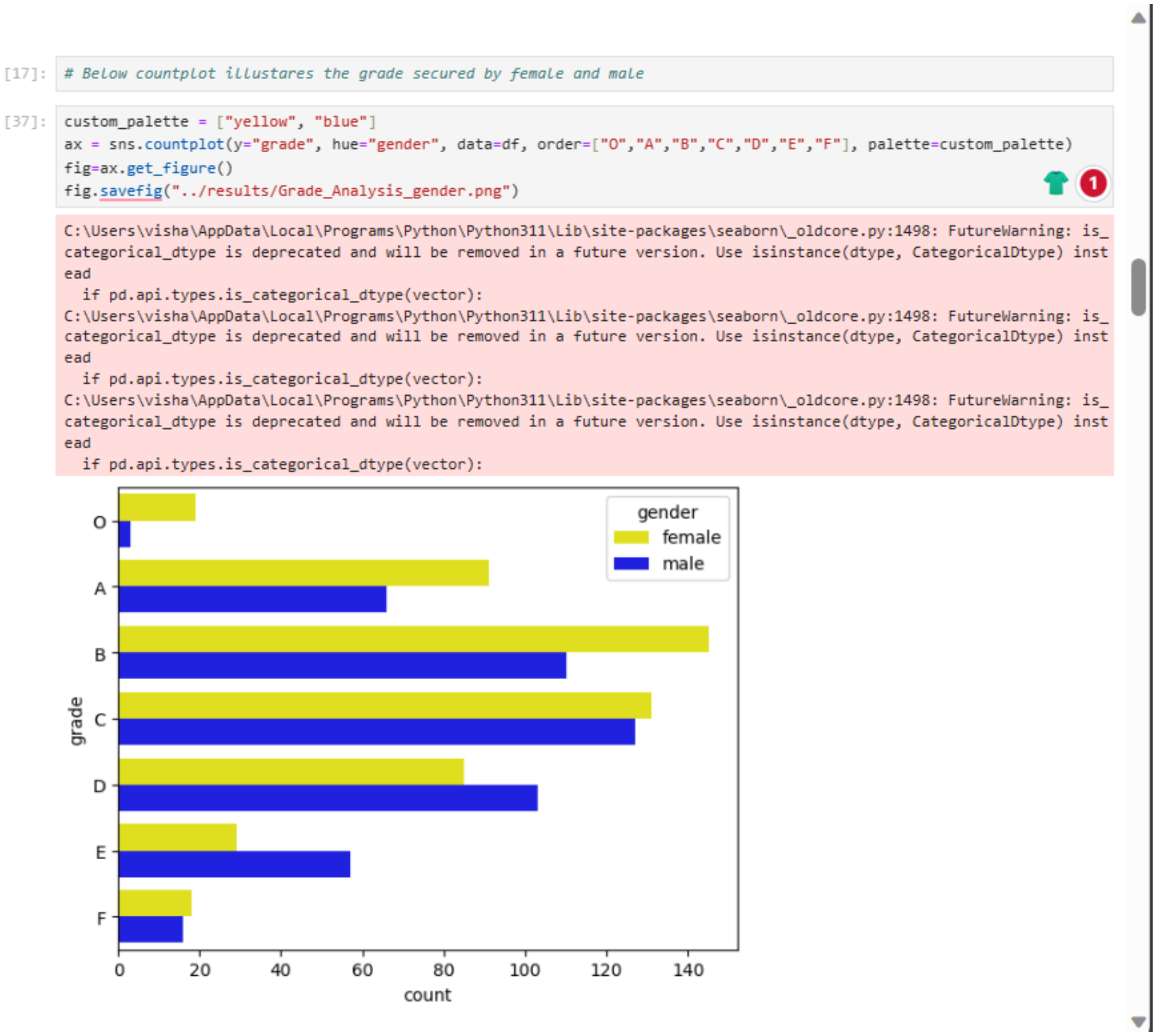
```
[14]: gender
      female    518
      male      482
      Name: count, dtype: int64
```

```
[15]: import matplotlib.pyplot as plt
```

```
[35]: labels=['Female', 'Male']

      plt.pie(df['gender'].value_counts(),labels=labels,explode=[0.1,0.1],
              autopct='%1.2f%%',colors=['#FFFF00', '#0000FF'], startangle=90)
```

```
[35]: ([<matplotlib.patches.Wedge at 0x2557ae6b990>,
        <matplotlib.patches.Wedge at 0x2557a1270d0>],
       [Text(-1.1980818587083752, -0.06782226650507366, 'Female'),
        Text(1.1980818587083752, 0.0678222665050735, 'Male')],
       [Text(-0.698881084246552, -0.03956298879462629, '51.80%'),
        Text(0.698881084246552, 0.039562988794626205, '48.20%')])
```

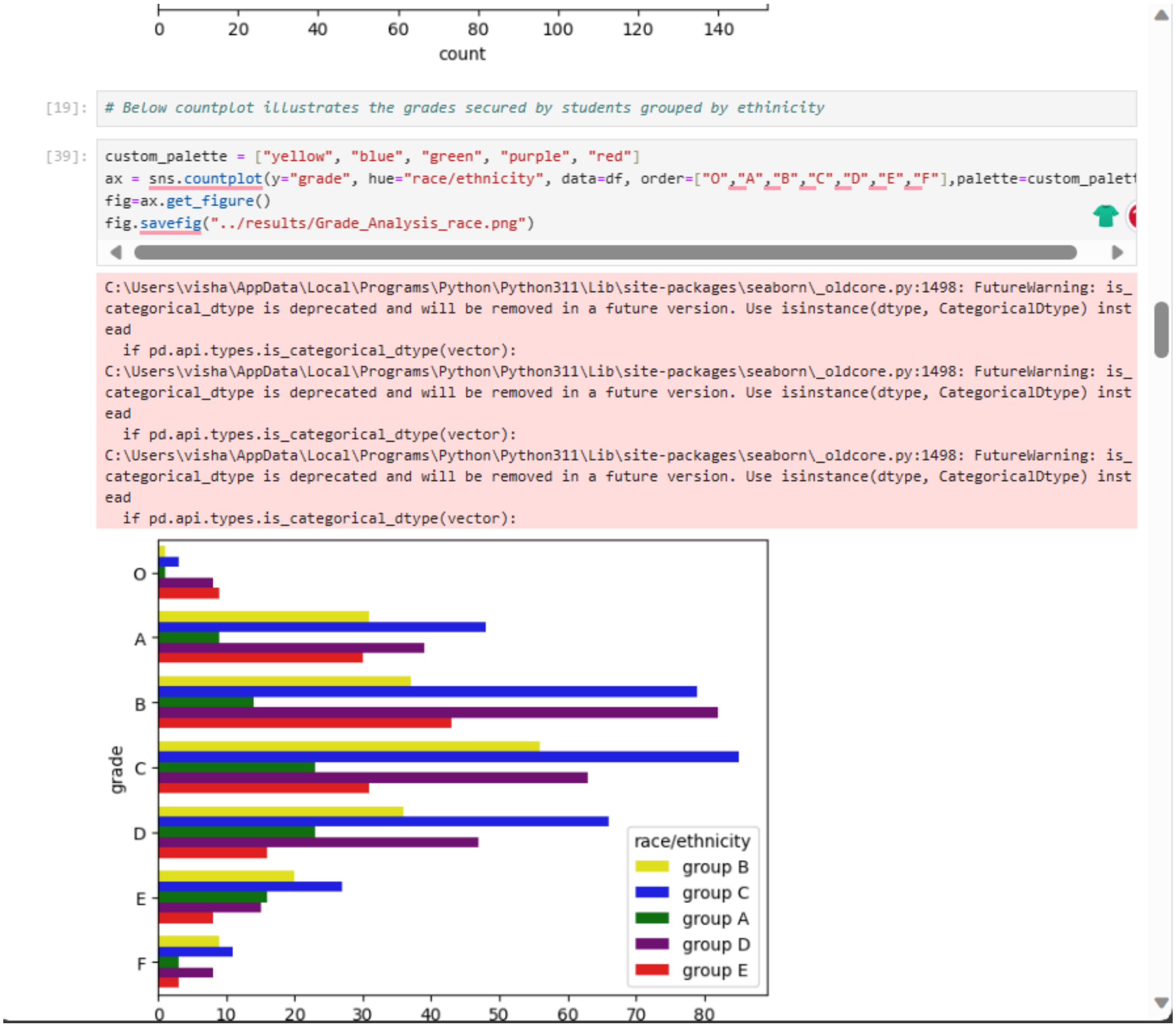
```
[17]: # Below countplot illustares the grade secured by female and male
```



```
[17]: # Below countplot illustares the grade secured by female and male
```

```
[37]: custom_palette = ["yellow", "blue"]
      ax = sns.countplot(y="grade", hue="gender", data=df, order=["O","A","B","C","D","E","F"], palette=custom_palette)
      fig=ax.get_figure()
      fig.savefig("../results/Grade_Analysis_gender.png")
```

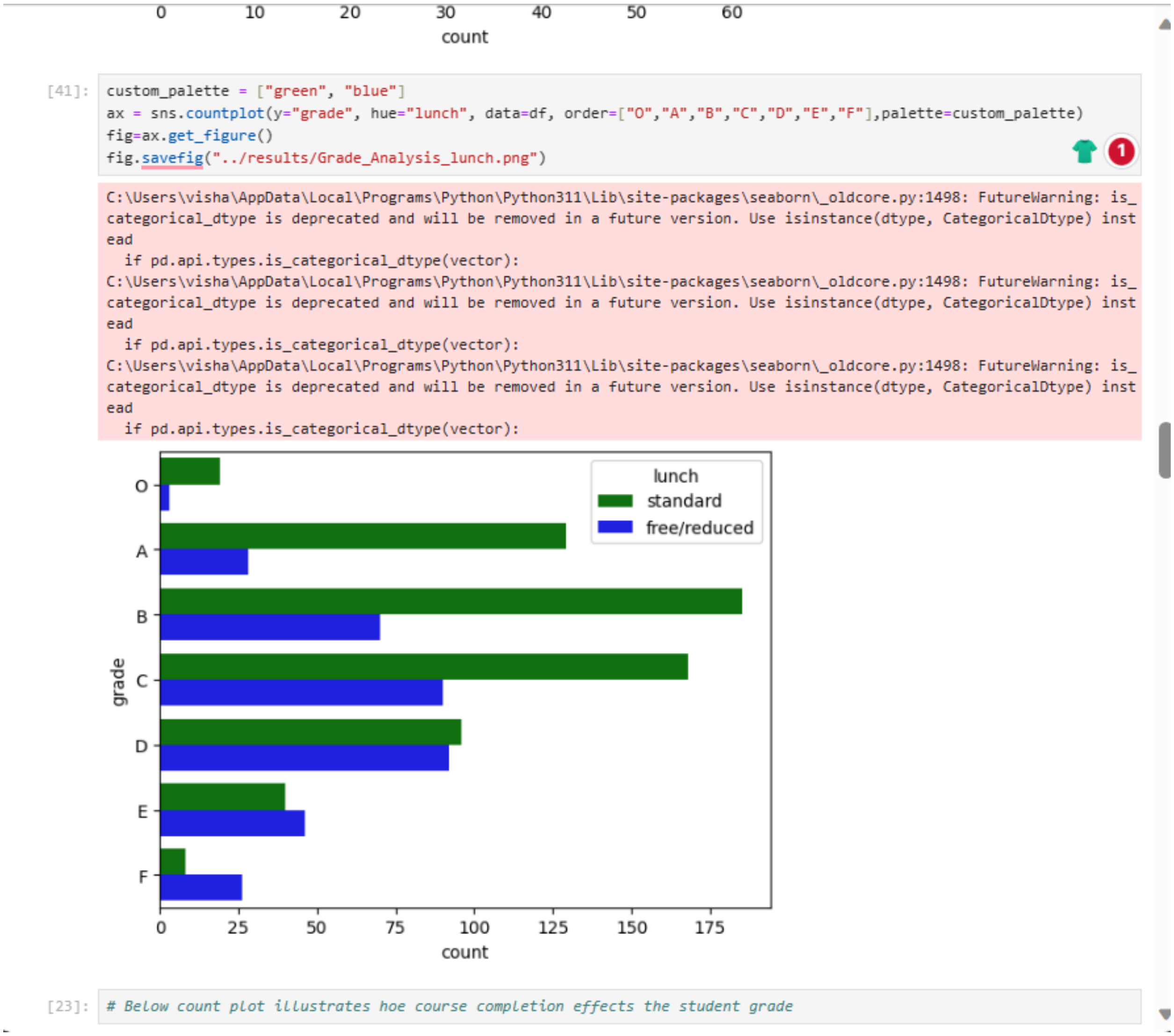
```
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
```



[19]:
```
# Below countplot illustrates the grades secured by students grouped by ethinicity
```

[39]:
```
custom_palette = ["yellow", "blue", "green", "purple", "red"]
ax = sns.countplot(y="grade", hue="race/ethnicity", data=df, order=["O","A","B","C","D","E","F"],palette=custom_palett
fig=ax.get_figure()
fig.savefig("../results/Grade_Analysis_race.png")
```

```
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
```



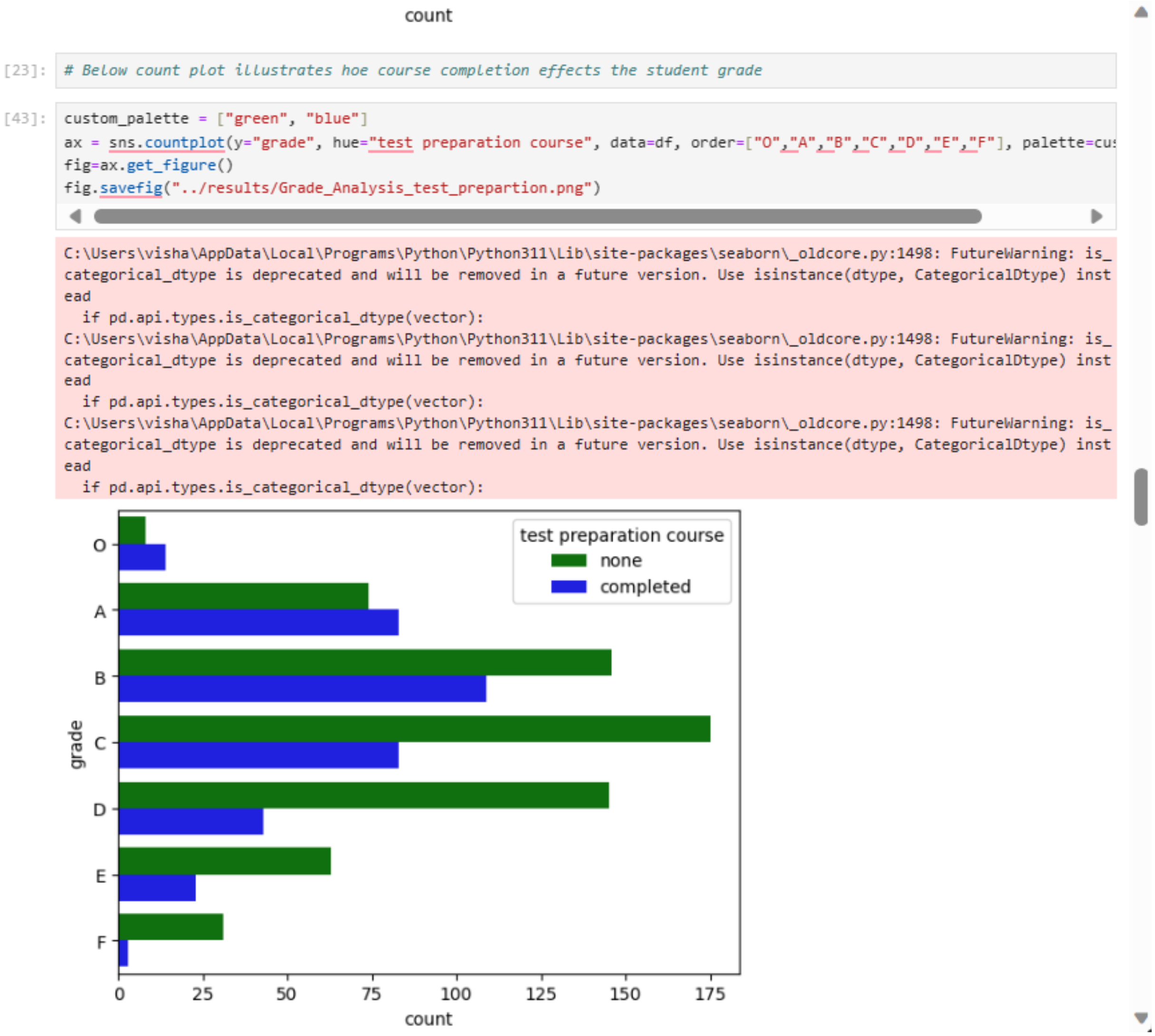
[41]:

```
custom_palette = ["green", "blue"]
ax = sns.countplot(y="grade", hue="lunch", data=df, order=["O","A","B","C","D","E","F"],palette=custom_palette)
fig=ax.get_figure()
fig.savefig("../results/Grade_Analysis_lunch.png")
```

```
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
```

[23]:

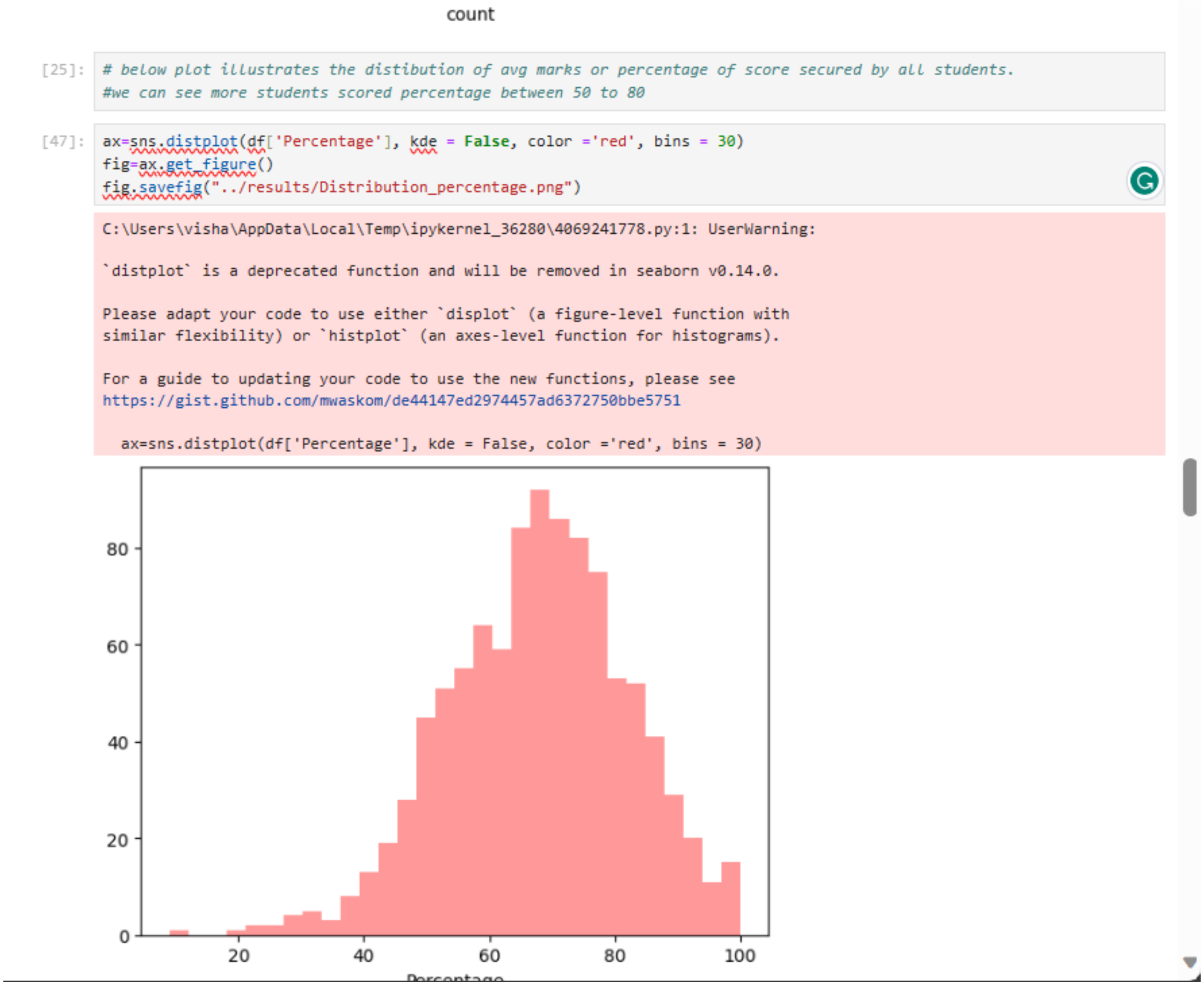
```
# Below count plot illustrates hoe course completion effects the student grade
```



```
[23]: # Below count plot illustrates hoe course completion effects the student grade
```

```
[43]: custom_palette = ["green", "blue"]
      ax = sns.countplot(y="grade", hue="test preparation course", data=df, order=["O","A","B","C","D","E","F"], palette=cu
      fig=ax.get_figure()
      fig.savefig("../results/Grade_Analysis_test_prepartion.png")
```

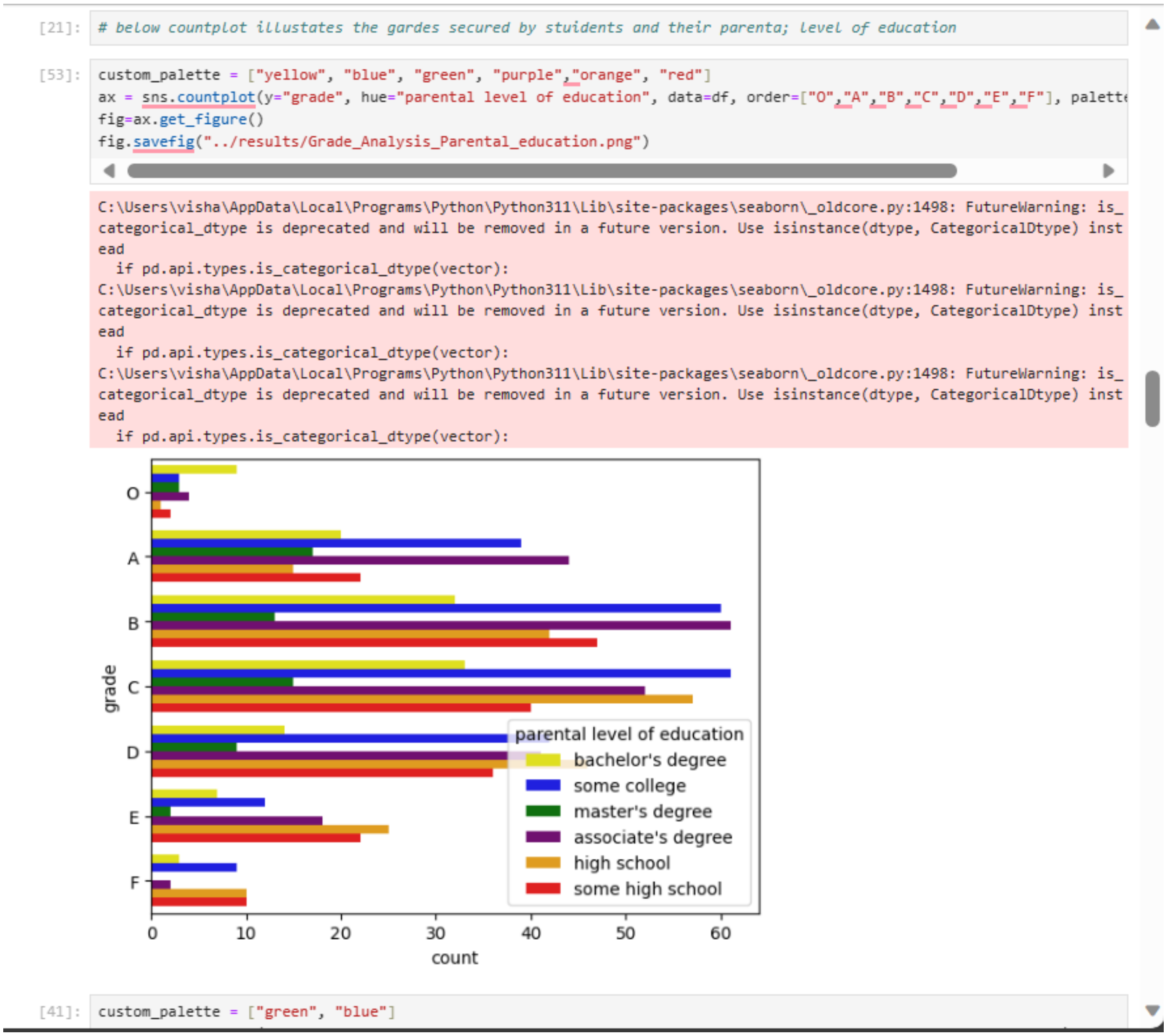
```
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
```



```
[25]: # below plot illustrates the distibution of avg marks or percentage of score secured by all students.
      #we can see more students scored percentage between 50 to 80
```

```
[47]: ax=sns.distplot(df['Percentage'], kde = False, color ='red', bins = 30)
      fig=ax.get_figure()
      fig.savefig("../results/Distribution_percentage.png")
```

```
C:\Users\visha\AppData\Local\Temp\ipykernel_36280\4069241778.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  ax=sns.distplot(df['Percentage'], kde = False, color ='red', bins = 30)
```

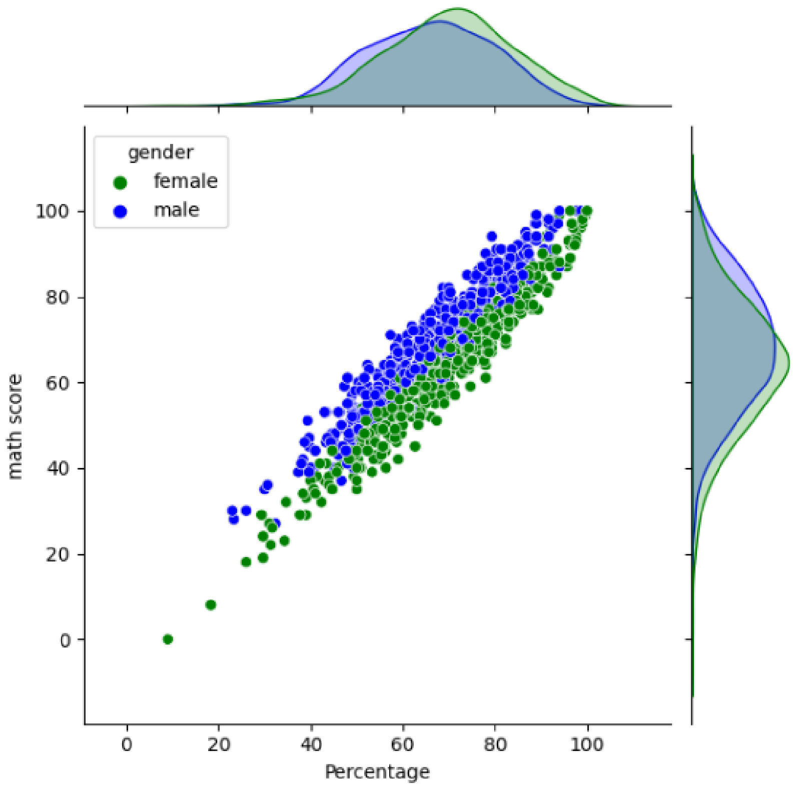


```
[21]: # below countplot illustates the gardes secured by stuidents and their parenta; level of education
```

```
[53]: custom_palette = ["yellow", "blue", "green", "purple","orange", "red"]
      ax = sns.countplot(y="grade", hue="parental level of education", data=df, order=["O","A","B","C","D","E","F"], palette
      fig=ax.get_figure()
      fig.savefig("../results/Grade_Analysis_Parental_education.png")
```

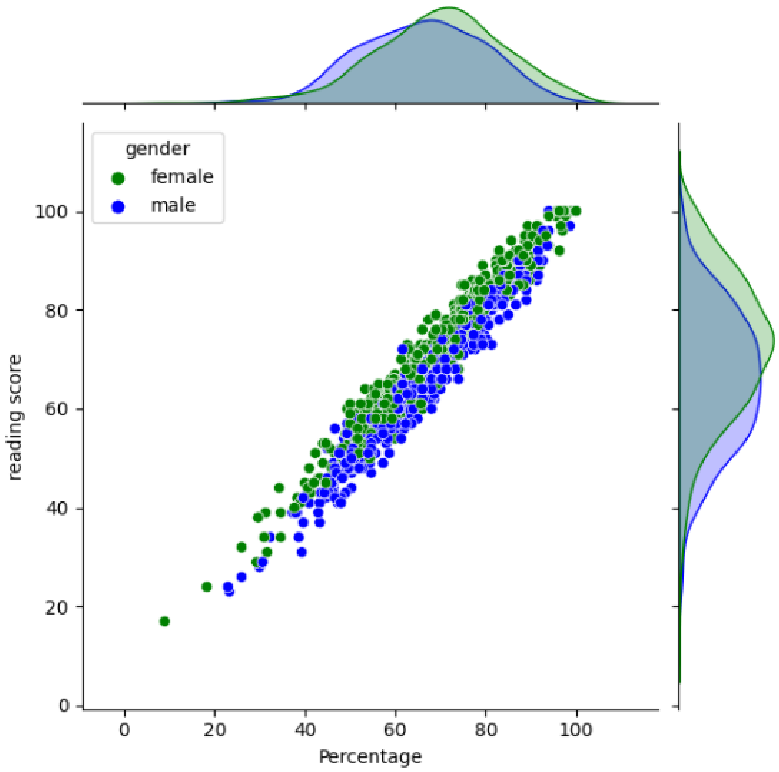
```
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
C:\Users\visha\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_
categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) inst
ead
  if pd.api.types.is_categorical_dtype(vector):
```

```
[41]: custom_palette = ["green", "blue"]
```

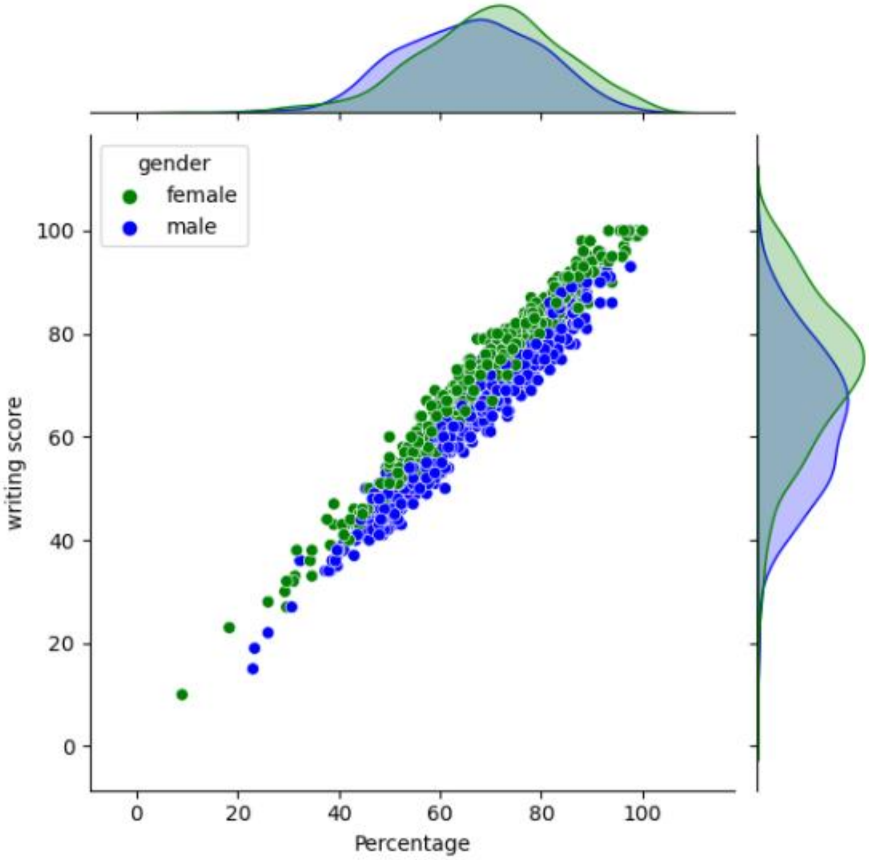


[50]:

```
custom_palette = ["green", "blue"]
ax=sns.jointplot(x ='Percentage', y ='reading score',hue="gender", data = df, palette=custom_palette)
```



```
[51]: custom_palette = ["green", "blue"]
      ax=sns.jointplot(x ='Percentage', y ='writing score',hue="gender", data = df, palette=custom_palette)
```



## Conclusion:

The visualizations presented in the analysis make it easier to compare gender-based differences in academic performance, assess disparities among racial and ethnic groups, explore the impact of parental education, lunch type, and test preparation on academic performance. These visualizations offer clear and intuitive insights, but their selection should align with specific research questions and the need for data exploration. Ultimately, they serve as valuable tools for hypothesis generation and initial insights.