seaborn-visualization

March 6, 2023

1 Data visualization using Seaborn

[43]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[68]:
```
df = pd.read_csv('/content/drive/MyDrive/Student.csv')
df.dropna(inplace = True)
df.reset_index(drop = True, inplace = True)
```

[69]:
```
df.columns
```

[69]:
```
Index(['Gender', 'Ethnicity', 'PLE', 'Lunch', 'TPE', 'Math score',
       'Reading score', 'Writing score'],
      dtype='object')
```

[46]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Ethnicity      1000 non-null   object
 1   PLE            1000 non-null   object
 2   Lunch          1000 non-null   object
 3   TPE            1000 non-null   object
 4   Math score     1000 non-null   int64
 5   Reading score  1000 non-null   int64
 6   Writing score  1000 non-null   int64
dtypes: int64(3), object(4)
memory usage: 54.8+ KB
```

[47]:
```
df.describe()
```

[47]:
```
       Math score  Reading score  Writing score
count  1000.00000    1000.000000    1000.000000
mean     66.08900      69.169000      68.054000
std      15.16308      14.600192      15.195657
min       0.00000      17.000000      10.000000
25%      57.00000      59.000000      57.750000
50%      66.00000      70.000000      69.000000
75%      77.00000      79.000000      79.000000
max     100.00000     100.000000     100.000000
```

[48]:
```
df.head()
```

[48]:
```
  Ethnicity                 PLE         Lunch        TPE  Math score  \
0   group B   bachelor's degree      standard       none          72
1   group C        some college      standard  completed          69
2   group B     master's degree      standard       none          90
3   group A  associate's degree  free/reduced       none          47
4   group C        some college      standard       none          76

   Reading score  Writing score
0             72             74
1             90             88
2             95             93
3             57             44
4             78             75
```

1. Scatter plot

[ ]:
```
ax = sns.scatterplot(data = df, x = 'Reading score', y = 'Writing score', color␣
↪= 'indigo', alpha = 0.5)
ax.set_title('Reading score vs Writing score')

# The scatter plot below shows the variation of reading score with respect to
# writing score. The variation is linear and depicts that as the reading score
# increases the writing score increases as well.
```

[ ]:
```
Text(0.5, 1.0, 'Reading score vs Writing score')
```

1.1 Scatter plot for more than 2 variables

[ ]:
```
ax = sns.scatterplot(data = df, x = 'Reading score', y = 'Writing score', hue =␣
↪'Gender')
ax.set_title('Reading score vs Writing score')

# The partitioning using hue shows that there is almost same number of male and
# female student having their respective scores which varies linearly.
```

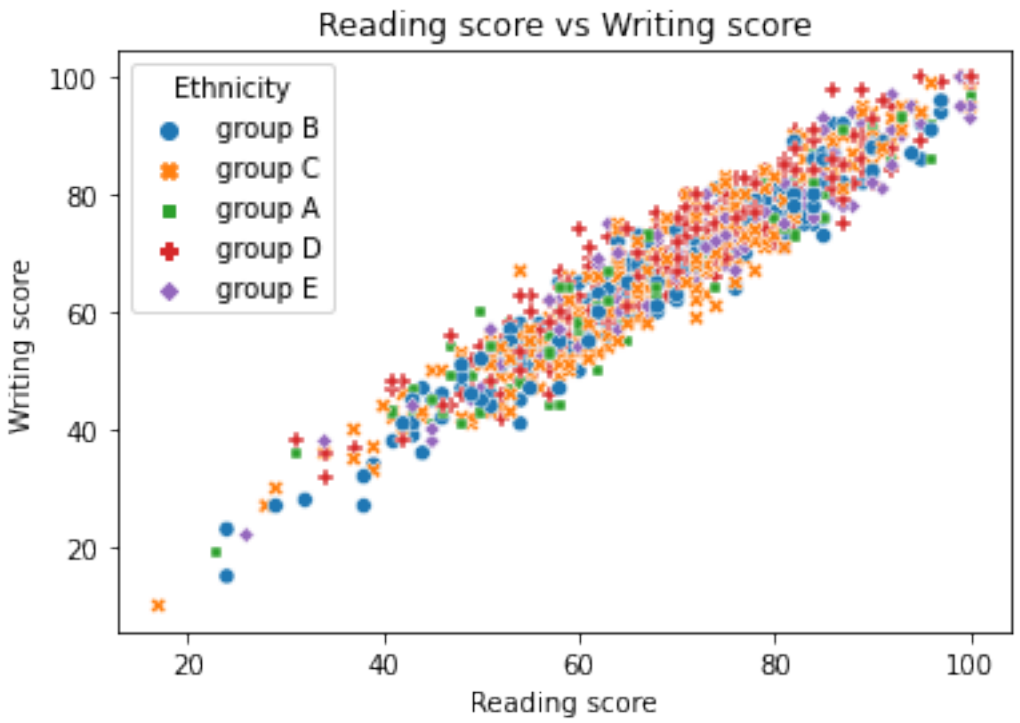
[ ]:
```
Text(0.5, 1.0, 'Reading score vs Writing score')
```

1.2 Scatter plot using marker style

[ ]:
```
ax = sns.scatterplot(data = df, x = 'Reading score', y = 'Writing score', hue =␣
↪'Ethnicity', style = 'Ethnicity')
ax.set_title('Reading score vs Writing score')

# By using different styles for marker it is evident that most of the scores
# achieved in reading and writing are scored by students who fall in group C.
```

[ ]:
```
Text(0.5, 1.0, 'Reading score vs Writing score')
```

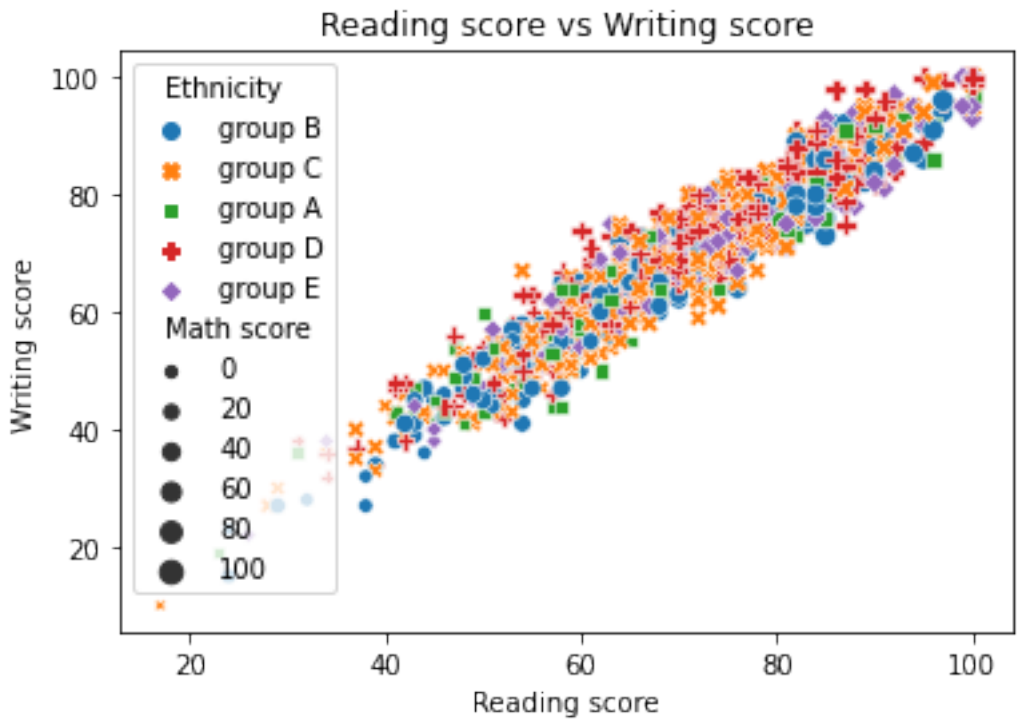


1.3 Scatter plot using marker size

[ ]:
```
ax = sns.scatterplot(data = df, x = 'Reading score', y = 'Writing score', hue =␣
↪'Ethnicity', style = 'Ethnicity', size = 'Math score')
ax.set_title('Reading score vs Writing score')

# By using different size of marker it can be infered that the highest score in
# maths are obtained by students who fall in group B which can be interospected
# through the mapping table.
```

[ ]:
```
Text(0.5, 1.0, 'Reading score vs Writing score')
```

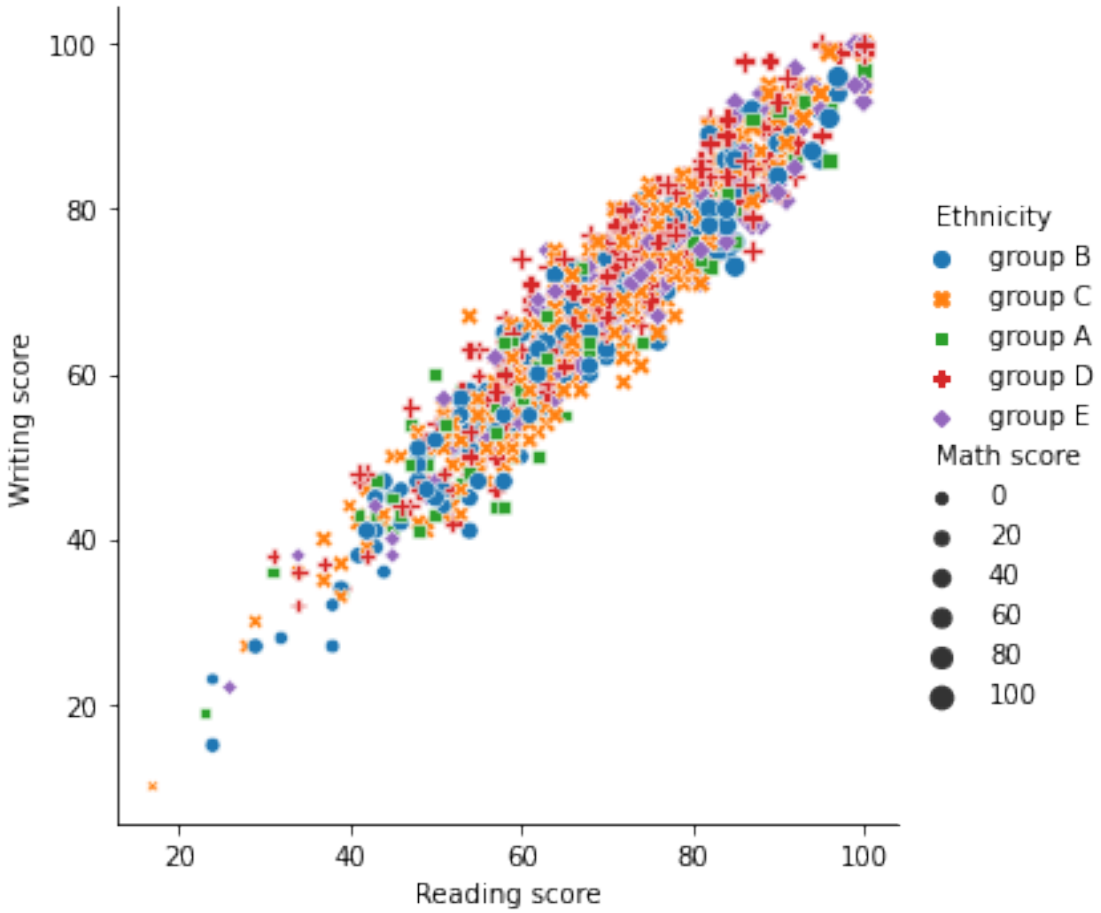


2. Relplot

[76]:
```
ax = sns.relplot(data = df, x = 'Reading score', y = 'Writing score', hue =␣
↪'Ethnicity', style = 'Ethnicity', size = 'Math score')
ax.set_titles('Reading score vs Writing score')

# The below visualization obtained through relplot results the same output as
# scatterplot. The advantage of relplot is that one can visualize multiple
# plots using this function by just changing the 'kind' parameter.
```

[76]:
```
<seaborn.axisgrid.FacetGrid at 0x7fde7fe0c220>
```



3. Line plot

[79]:
```
ax = sns.lineplot(data = df, x = 'Reading score', y = 'Writing score', hue =␣
↪'Gender')
ax.set_title('Reading score vs Writing score')

# As evident from earlier the writing score is linearly proportional to reading
# score. Therefore more the reading score more will be the writing score. From
# legend it can be seen that there is abrupt change in scores of female.
```

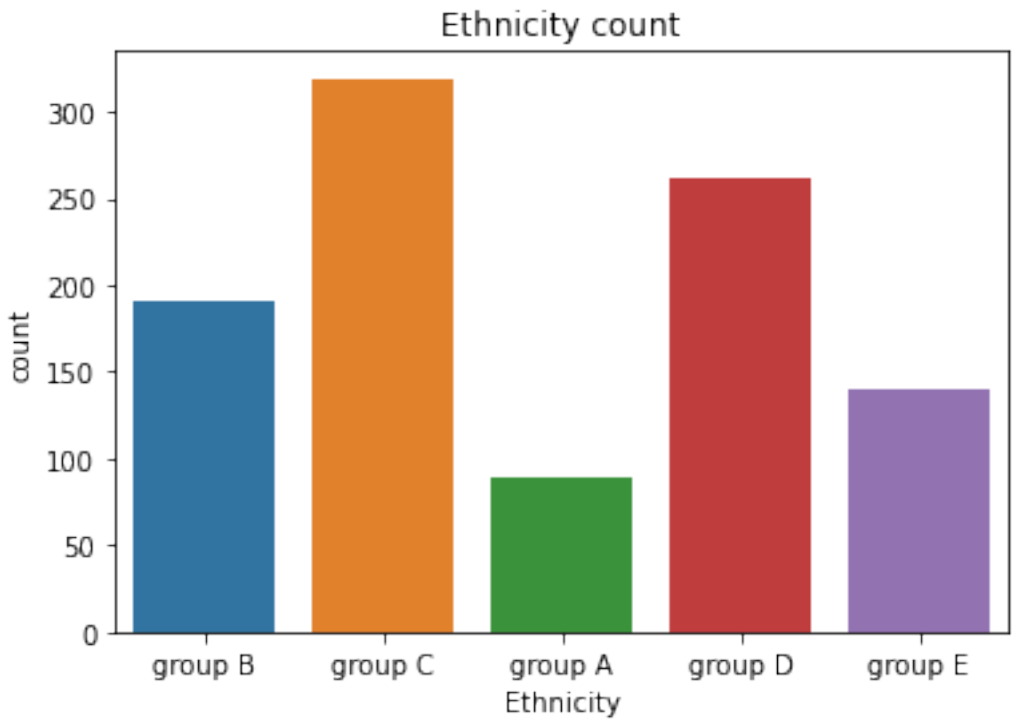
[79]:
```
Text(0.5, 1.0, 'Reading score vs Writing score')
```

4. Count plot

[52]:
```
ax = sns.countplot(data = df, x = 'Ethnicity')
ax.set_title('Ethnicity count')

# The plot below shows the unique count for each group of ethnicity. It can be
# concluded that most of the students belong to group C while group A has the
# least number of studnets.
```

[52]:
```
Text(0.5, 1.0, 'Ethnicity count')
```

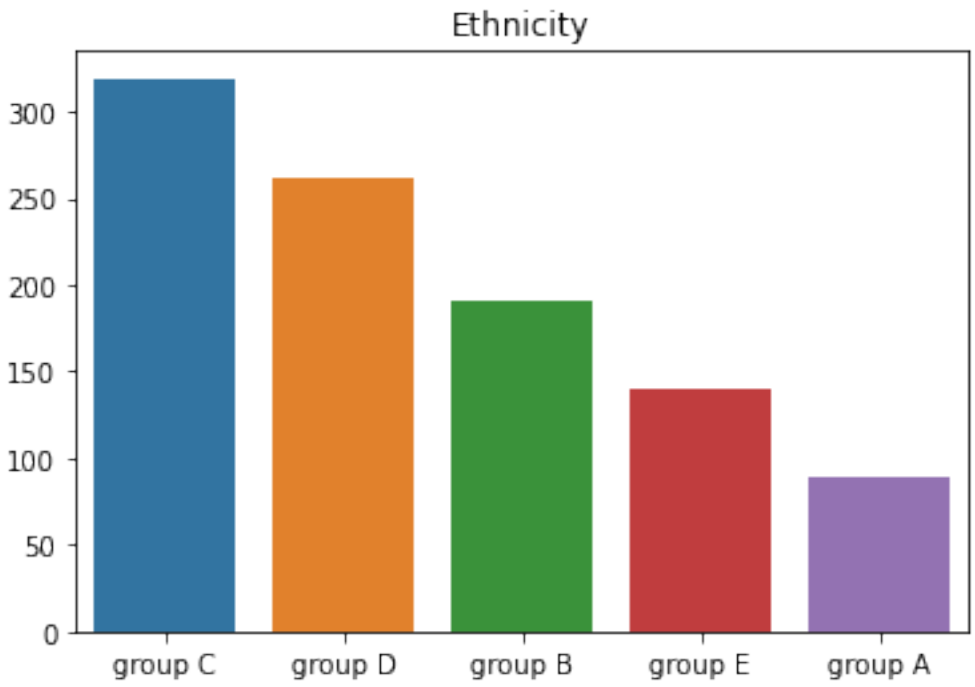


5. Bar plot

[53]:
```
data = df['Ethnicity'].value_counts()
m = data.index
n = data.values
ax = sns.barplot(data = df, x = m, y = n)
ax.set_title('Ethnicity')

# The bar plot function achieves the same output as above. The only thing that
# varies is that the frequency is in sorted order and hence it is visually
# appealing. Also unlike matplotlib and pandas the bar stripe is automatically
# colored differently, hence making visualization more intuitive.
```

[53]:
```
Text(0.5, 1.0, 'Ethnicity')
```

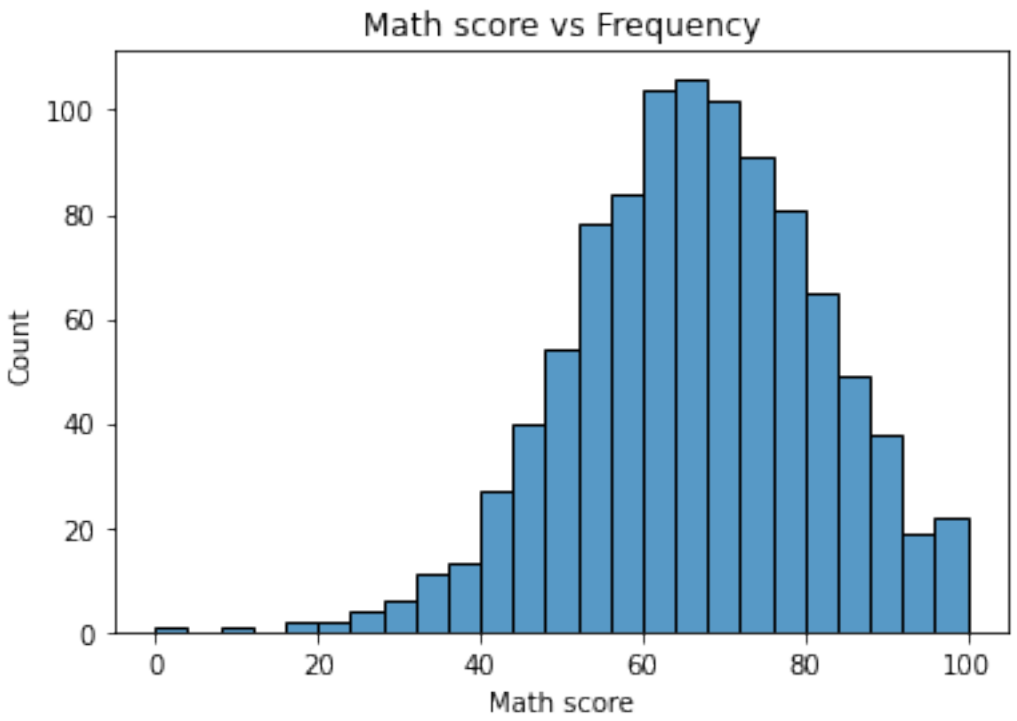


6. Historgram

[54]:
```
ax = sns.histplot(data = df, x = 'Math score')
ax.set_title('Math score vs Frequency')

# The histogram below shows the peak of math score using frequency by taking
# score as continuous interval. It can be seen that the math score is achieved
# maximum for range 65-70 and minimum for range 0-15 i.e., most of the students
# achieved math score in the range 65-70. This plot is more accurate to analyze
# when compared to histogram plotted using pandas and matplotlib.
```

[54]:
```
Text(0.5, 1.0, 'Math score vs Frequency')
```

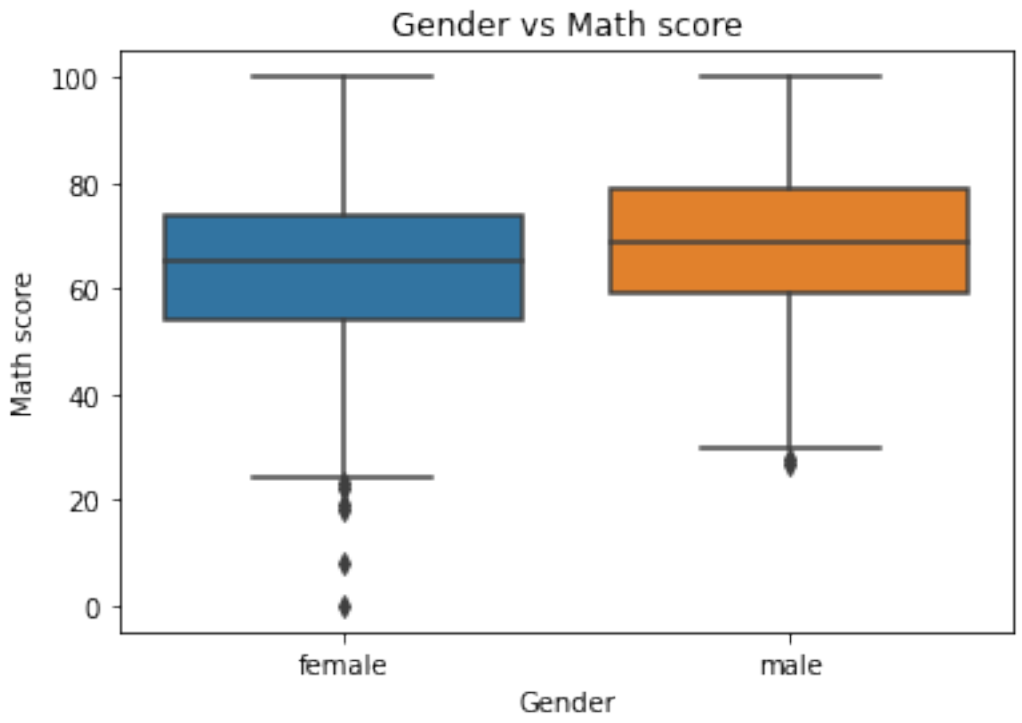


7. Box plot

[71]:
```
ax = sns.boxplot(data = df, x = 'Gender', y = 'Math score')
ax.set_title('Gender vs Math score')

# From the box plot below it can be seen that the median value of math score for
# male is above female which shows that highest score in maths is achieved by
# male. Also the whiskers or outliers extends more in female than male which
# shows that the score obtained by female is less compared to majority.
```

[71]:
```
Text(0.5, 1.0, 'Gender vs Math score')
```

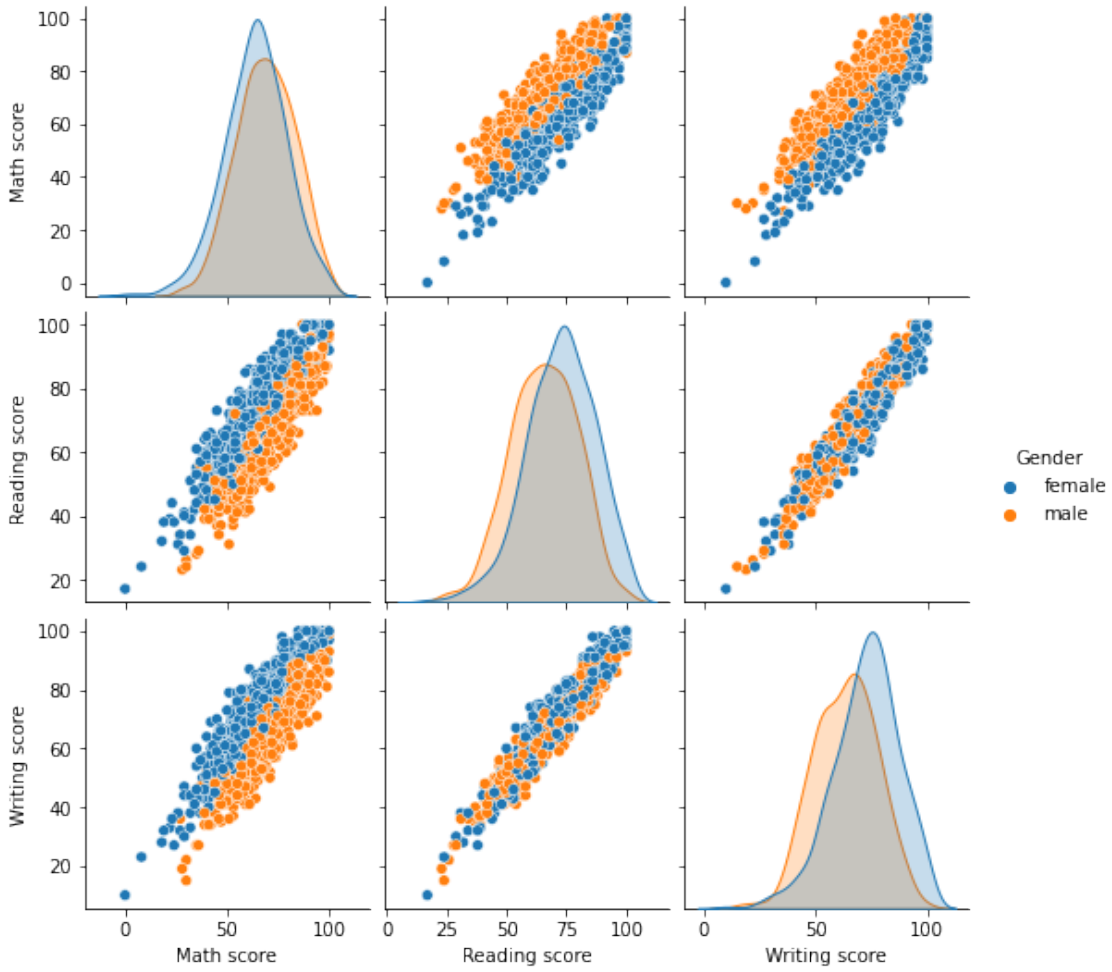


8. Pair plot

[73]:
```
sns.pairplot(data = df, hue = 'Gender')

# The pairplot helps in visualizing all the quantitative attribute in a dataset
# and having density or histogram plot in the diagonal. This helps in analyzing
# multivariate data in the form of matrix. As seen from below all the␣
 ↪attributes
# vary linearly with respect to each other. The diagonal density plot shows that
# the female density has more peak than male indicating that there is more
# number of female compared to male.
```

[73]:
```
<seaborn.axisgrid.PairGrid at 0x7fde80395760>
```

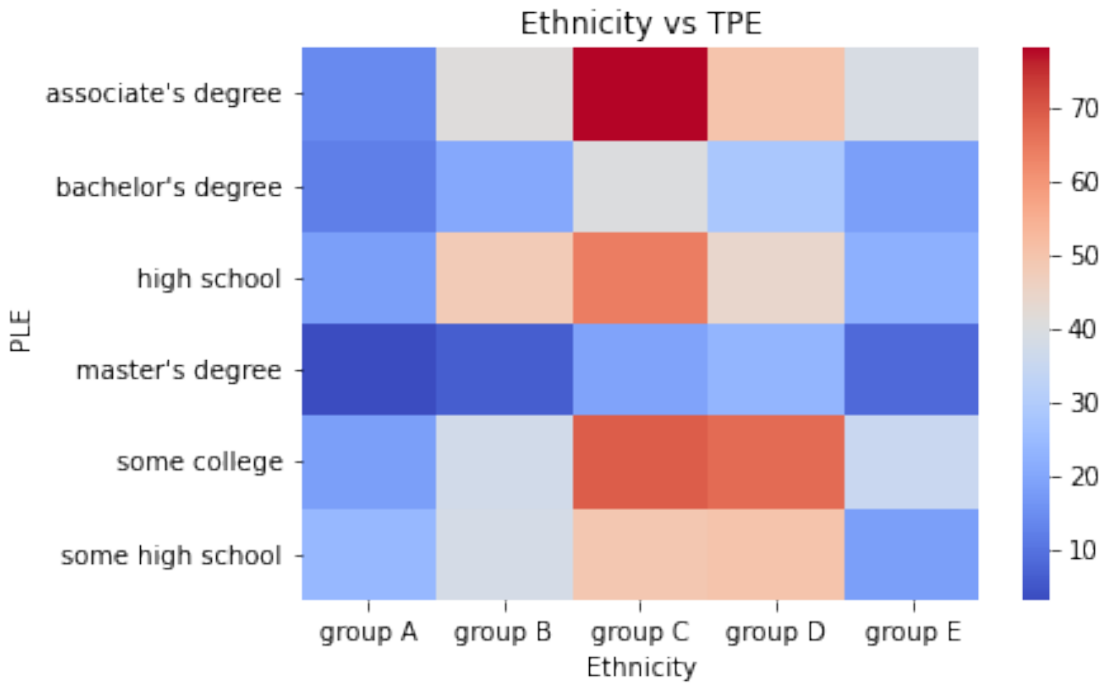


9. Heatmap

[74]:
```
newdf = pd.crosstab(df['PLE'], df['Ethnicity'])
ax = sns.heatmap(newdf, cmap = 'coolwarm')
ax.set_title('Ethnicity vs TPE')

# The heatmap below shows the coorelation of PLE with respect to Ethnicity.
# The dark red portion corresponding to associate degree and group C shows that
# they are highly correlated while the dark blue portion corresponding to
# master's degree and group A are least correlated.
```

[74]:
```
Text(0.5, 1.0, 'Ethnicity vs TPE')
```



10. Swarmplot

[81]:
```
ax = sns.swarmplot(data = df, x = 'Ethnicity', y = 'Math score')
ax.set_title('Ethnicity vs Math score')

# The swarmplot below helps in visualizing data by taking categorical data as
# axis. The points are arranged such that they do not overlap and hence reduces
# the cognitive load. The thin distribution of score among group A signifies
# that there is a strong relation among categorical variable. Group E data is
# assymetrical indicating that some other factor is influncing the data. Some
# of the datapoints in group A are bit seperated from other showing that it
# needs further investigation to visualize data.
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/categorical.py:1296: UserWarning:
20.0% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/categorical.py:1296: UserWarning:
36.4% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/categorical.py:1296: UserWarning:
32.1% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

[81]: Text(0.5, 1.0, 'Ethnicity vs Math score')