FinalProject_MS1_KanaparthiVenkata

March 3, 2022

[1]:
```python
import pandas as pd
import yellowbrick
```

[2]:
```python
#Step 1:  Load data into a dataframe
addr1 = "StudentsPerformance.csv"
data = pd.read_csv(addr1)
```

[3]:
```python
# Step 2:  check the dimension of the table
print("The dimension of the table is: ", data.shape)
```

```
The dimension of the table is:  (1000, 8)
```

[4]:
```python
#Step 3:  Look at the data
print(data.head(5))
```

```
   gender race/ethnicity parental level of education         lunch  \
0  female        group B           bachelor's degree      standard
1  female        group C                some college      standard
2  female        group B             master's degree      standard
3    male        group A          associate's degree  free/reduced
4    male        group C                some college      standard

  test preparation course  math score  reading score  writing score
0                    none          72             72             74
1               completed          69             90             88
2                    none          90             95             93
3                    none          47             57             44
4                    none          76             78             75
```

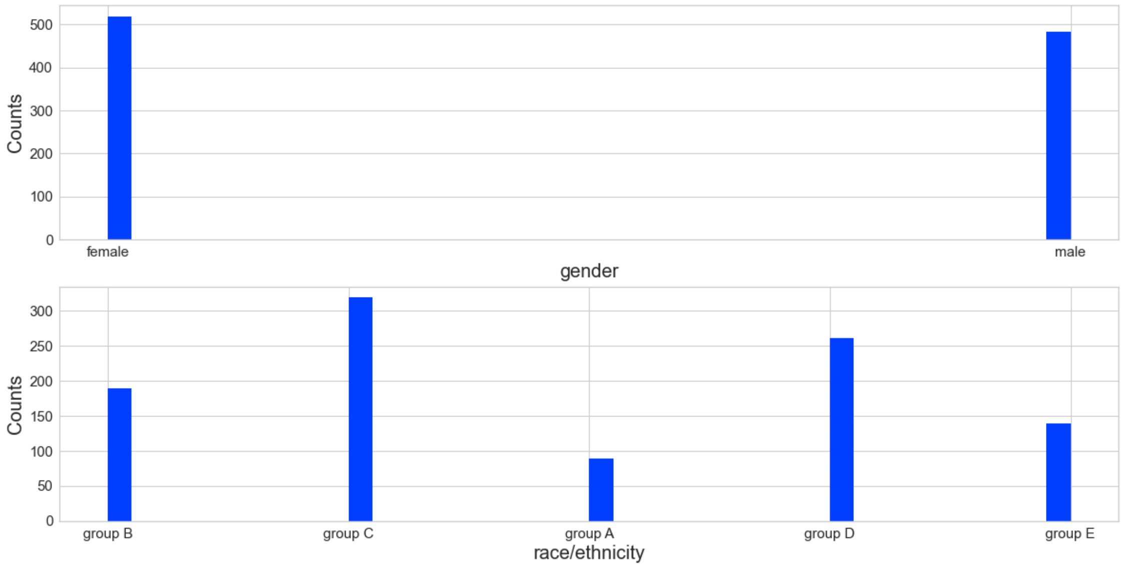
[5]:
```python
#Step 4:  what type of variables are in the table
print("Describe Data")
print(data.describe())
print("Summarized Data")
print(data.describe(include=['O']))
```

```
Describe Data
       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
Summarized Data
        gender race/ethnicity parental level of education     lunch  \
count     1000           1000                        1000      1000
unique       2              5                           6         2
top     female        group C                some college  standard
freq       518            319                         226       645

       test preparation course
count                     1000
unique                       2
top                       none
freq                       642
```

[19]:

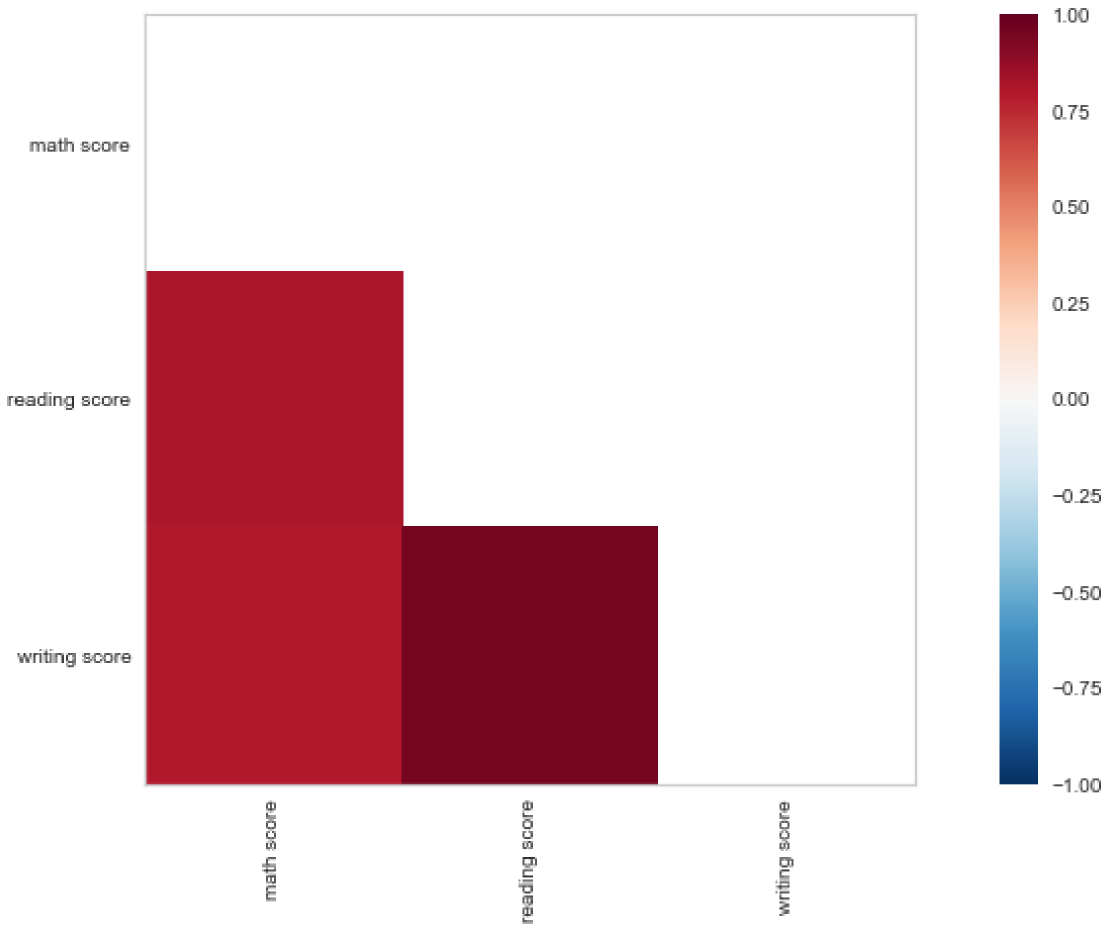
```python
#Step 5: import visulization packages
import matplotlib.pyplot as plt

# set up the figure size
plt.rcParams['figure.figsize'] = (20, 10)

# make subplots
fig, axes = plt.subplots(nrows = 2, ncols = 1)

# Specify the features of interest
num_features = ['gender', 'race/ethnicity']
xaxes = num_features
yaxes = ['Counts', 'Counts']

# draw histograms
axes = axes.ravel()
for idx, ax in enumerate(axes):
    ax.hist(data[num_features[idx]].dropna(), bins=40)
    ax.set_xlabel(xaxes[idx], fontsize=20)
    ax.set_ylabel(yaxes[idx], fontsize=20)
    ax.tick_params(axis='both', labelsize=15)
plt.show()
```



As per the above graphs, the second graph shows it doesnt have the symmetrical data for the groups

[8]:

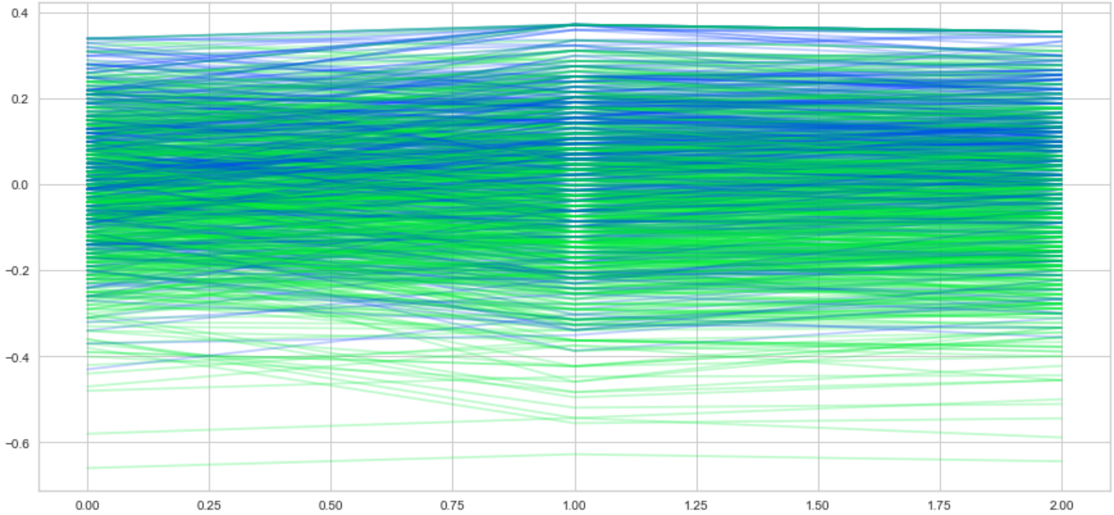
```
#Step 6: Pearson Ranking
#set up the figure size
#%matplotlib inline
plt.rcParams['figure.figsize'] = (15, 7)

# import the package for visulization of the correlation
from yellowbrick.features import Rank2D
num_features=['math score','reading score','writing score']

# extract the numpy arrays from the data frame
X = data[num_features].values

# instantiate the visualizer with the Covariance ranking algorithm
visualizer = Rank2D(features=num_features, algorithm='pearson')
visualizer.fit(X)                # Fit the data to the visualizer
visualizer.transform(X)             # Transform the data
plt.show()
```



[11]:

```
# Step 7:  Compare variables against Course completed ot not
#set up the figure size
#%matplotlib inline
plt.rcParams['figure.figsize'] = (15, 7)
plt.rcParams['font.size'] = 50

# setup the color for yellowbrick visulizer
from yellowbrick.style import set_palette
set_palette('sns_bright')

# import packages
from yellowbrick.features import ParallelCoordinates
# Specify the features of interest and the classes of the target
classes = ['none', 'completed']
num_features=['math score','reading score','writing score']

# copy data to a new dataframe
data_norm = data.copy()
```

```
# normalize data to 0-1 range
for feature in num_features:
    data_norm[feature] = (data[feature] - data[feature].mean(skipna=True)) /␣
 ↪(data[feature].max(skipna=True) - data[feature].min(skipna=True))

# Extract the numpy arrays from the data frame
X = data_norm[num_features].values
y = data['test preparation course'].values

# Instantiate the visualizer
visualizer = ParallelCoordinates(classes=classes, features=num_features)
visualizer.fit(X, y)      # Fit the data to the visualizer
visualizer.transform(X)   # Transform the data
plt.show();
```


0.4
0.2
0.0
−0.2
−0.4
−0.6
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00