Middle School Data Analysis

Sangwon Baek

May 25th, 2021

Introduction to Data Science

Professor Pascal Wallisch

Data Preprocessing:

In this data analysis project, I imputed each column's NaN values (missing data) by using respective column's mean values. Although I was aware of the fact that data should only come from the measurement, I decided to choose imputation instead of removal to maintain the data points as much as possible. Removing all missing data would result a significant loss of data that the data loses its statistical power. Thus, some of the data was missing systematically that removal method could have led to losing all data for charter schools. During imputation, taking the mean values seemed reasonable for replacing the missing data. The reason was that the variables of the dataset were about the average performance, student ratings, or size of the student body; the mean values for these variables did not significantly harmed the data set's validity. This data cleaning process was necessary to be done prior to running any tests to avoid issues with NaN values.

Dimension Reduction (Principal Component Analysis):

To successfully handle dimension reduction, I performed PCA on the variables. I first excluded school name and dbn because these two variables are string data that serve as identification for each school in a row. Then I separated the data into predictors (columns C, E, F, G-K, L-Q, R, S, T, and U), objective achievement (columns V,W,X), and admission (column D). I made this separation prior to running PCA because it was important to make the distinction between the independent variables (predictors) and dependent variables (outcomes). Prior to the PCA, I did exploratory analysis of the variables through creating a color plot of the correlation matrix, shown in **Appendix A**. The diverse color scheme on the plot represents the existence of variability, and the bright colors such as red and light green signal high correlation between the variables. Through the bright colors in the plot, I could identify potential high correlation among the variables L-Q (the school climate factors), the variables S-T(student condition), and the variables V-X (objective achievement variables). Then, I first obtained the z-score values of the data to normalize the data. After that, I used those z-score values to run PCA. The two scree plots, shown in **Appendix B**, demonstrate the sorted Eigen values, which is the variance accounted for, in the descending order. Through these plots, using the Kaiser criterion, which is to keep the factors with eigenvalue>1, I identified that the 18 variables could be reduced to 4 factors, meaning there exist 4 uncorrelated factors that can interpret dataset meaningfully, the factor plots corresponding to these 4 factors are shown in **Appendix C**. On top of that, I identified that objective achievement could be reduced to 1 factor, which is also shown in **Appendix C**.

1) What is the correlation between the number of applications and admissions to HSPHS?

The correlation between the number of applications and admission to HSPHS is 0.801727. The scatter plot, shown in **Appendix D,** demonstrates how the two variables are highly correlated visually. As you see, the majority of data points (located between range 0-100 for applications, range 0-50 for acceptances) are distributed close to one another and the overall trend show a positive relationship between the two variables.

2) What is a better predictor of admission to HSPHS? Raw number of applications or application *rate*?

First I divided the applicants by the school size to get the application rate. Then, I used the correlation matrix to identify the correlation, and used the correlation to determine which predictor is better than the other predictor. My results show that the better predictor of admission to HSPHS is the raw number of applications. The raw number of applications has a higher correlation with admission to HSPHS, correlation=0.801766, than the correlation of application rate and admission to HSPHS, which is 0.658751.

3) Which school has the best *per student* odds of sending someone to HSPHS?

I calculated the per student odds of admission to HSPHS through dividing the number of acceptances by the number of applications. Through this calculation I was able to identify that Christa Mcauliffe School\I.S 187 had the highest odds of sending a student to HSPHS: the proportion of admission by applicants was 81.67%.

4) Is there a relationship between how students perceive their school (as reported in columns L-Q) and how the school performs on objective measures of achievement (as noted in columns V-X).

I performed PCA to do dimension reduction prior to solve the questions. As a result, I generated the factor plots, shown in **Appendix C**. The factor 1 plot for predictors have high loadings for variables 8-13 that these 6 variables could be reduced to 1 school climate factor. Also, the factor 1 plot for objective achievement show high loadings for all three variables that it could be reduced to 1 factor, which is an objective measure of achievement.

After the dimension reduction, I generated a scatter plot to identify the relationship between the students' perception about the school and the school performance on objective measure of achievement, shown **Appendix E**. This scatter plot shows that there exists a high correlation between the two variables because the data points are distributed close to one another, and there exists a uphill trend that signals a positive relationship between the two variables. Therefore, when the students have a positive perception about their school their objective performance increases, and when they have a negative perception then their objective performance decreases.

5) Test a hypothesis of your choice as to which kind of school (e.g. small schools vs. large schools or charter schools vs. not (or any other classification, such as rich vs. poor school)) performs differently than another kind either on some dependent measure, e.g. objective measures of achievement or admission to HSPHS (pick one).

I suspect that the school size has an impact on the number of admission to HSPHS. To confirm whether this suspicion is true or not, I am going to run a hypothesis testing on the following null hypothesis: the school size (small vs large) has no statistically significant influence on the number of students' admission to HSPHS school. Although there exists an obvious confounding variable that large school is going to send more applications that higher number of applications would likely to increase the odds of admission to HSPHS, I used the number of admissions as the dependent variable due to the pragmatic reasons that we are not able to find admission rate for each school with the unknown population. With the dataset we have, I thought that the number of admissions seemed reasonable for explaining the null hypothesis.

Prior to running the hypothesis testing, I have converted the school size data into categorical variables of small or large. In this process, I used the median of the school size as the determining factor: assigned 0 (representing small) to the school size lower than the median and assigned 1 (representing large) to the school size higher than the median.

After the conversion of the school size variable, I turned the data into two samples based on their school size category (small or large). Then, I ran a two-sample-t-test on these two groups of dataset. Here I set the significance level as 0.01 and ran a two-sided t-test because we are interested in the difference between the two samples' mean. Our sample size was 297 for each sample that the degrees of freedom was 592. The result of the two sample t test, shown in **Appendix F**, has t-statistics of -8.736 and p-value of 1.819*10^-16. Since p-value < significance level, this result shows that there is a statistically significant different between the two samples that we can reject the null hypothesis. Therefore, there exists a significant influence on the admission to HSPHS based on the school size: large school are more likely to have their students admit to HSPHS.

6) Is there any evidence that the availability of material resources (e.g. per student spending or class size) impacts objective measures of achievement or admission to HSPHS?

To identify whether the per student spending or the class size have an impact on admission to HSPHS. I first converted the class size as the categorical variables of small class and large class and converted per student spending as the categorical variables of less spending and more spending. In this process, I used the median of each variable as the determining factor. For instance, I assigned 0 (representing small) to the class size lower than the median and assigned 1 (representing large) to the class size higher than the median. An identical process was done for the per student spending.

After the conversion of the class size and per student spending variables, I turned the data into two samples based on their class size category (small or large) and their spending category (less or more). Then, I ran a two-sample-t-test on these two groups of dataset. Here I set the significance level as 0.01 and ran a two-sided t-test because we are interested in the difference between the two samples' mean. Our sample size was 297 for each sample that the degrees of freedom was 592. The result of the two sample t test, shown in **Appendix G**, shows that t-statistics with -8.736 and p-value of 1.819*10^-16 for both per student spending data and class size data. Since p-value < significance level, this result shows that there is a statistically significant different between the two samples that we can reject the null hypothesis for both cases. Therefore, there exists a significant influence on the admission to HSPHS based on the class size and the per student spending: a large class and a school that spend more on student are likely to have their students admit to HSPHS.

7) What proportion of schools accounts for 90% of all students accepted to HSPHS?

Prior to answering the question, I first visualized the number of acceptances for each school in a decreasing rank order through a bar plot, shown in **Appendix H**. Here the school name does not seem visible because the names are too long that they are not printed correctly. Hence, in **Appendix I,** I prepared a table containing first 10 schools with acceptance numbers to facilitate the understanding of the bar plot.

To identify the proportion of schools accounting for 90% of all students accepted to HSPHS. I first calculated the total number of student accepted to HSPHS, 4461. Then calculated 90% of total acceptance, 4014.9. After that, I counted the number of schools in decreasing that have sent 90% of students to HSPHS. Then divided that number by the total number of schools. As a result, I found that about 20.7% of the school account for 90% of all students accepted to HSPHS

8) Build a model of your choice – clustering, classification or prediction – that includes all factors – as to what school characteristics are most important in terms of a) sending students to HSPHS, b) achieving high scores on objective measures of achievement?

I chose to run clustering methods through the use of K-means clustering to identify the number of clusters within the variables. I ran two different clustering using the factor 1 predictirons **(Appendix C)** that include school climate variable, number of applications variable, class & school size variable as the significant factors onto admission to HSPHS and objective measures of achievement.

I ran K-means clustering on predictors and admission, I obtained the following silhouette score expressed in histograms, shown in **Appendix J**. I then made the plot to analyze the corresponding number of clusters for obtaining the optimal silhouette score: 2 clusters. After that I plotted the data in a color to show each cluster. Through this cluster analysis, I identified that the Factor 1 variables only have positive correlation with the admission to HSPHS in the extreme cases that majority of the time, these variables do not significantly impact the admission, shown in **Appendix L**.

Then, I ran K-means clustering on predictors and admission, I obtained the following silhouette score expressed in histograms, shown in **Appendix M**. I then made the plot to analyze the corresponding number of clusters for obtaining the optimal silhouette score: 2

clusters. After that I plotted the data in a color to show each cluster. Through this cluster analysis, I identified that school climate variable, number of applications variables, class & school size variable indeed have a highly correlated positive relationship with the objective measures of achievement, shown in **Appendix N**.

9) Write an overall summary of your findings – what school characteristics seem to be most relevant in determining acceptance of their students to HSPHS?

The overall finding of my analysis of the dataset indicates that the most relevant determinants to the students' admission to HSPHS rely on the large school, large class sizes, and more per student spending. This argument is statistically strengthened through the results of the two-sample t-test, shown **in Appendix F&G**. There was a statistically significant relation that show the students from large school, large class size, received more spending had higher odds of getting accepted to HSPHS.

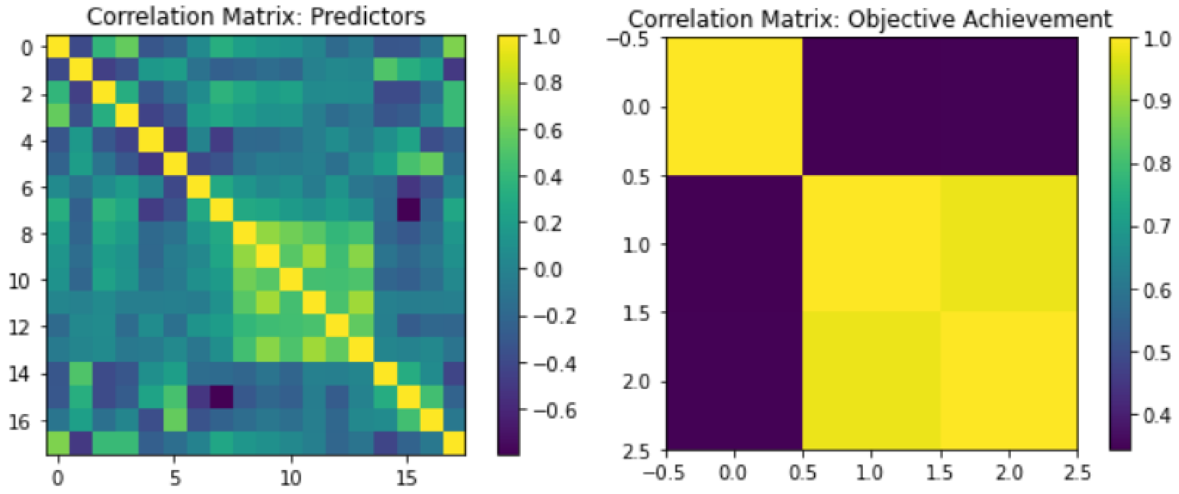
In contrast, the objective measures of achievement relied on the quality of the school, which was indicated by the school climate factors (student's general perception about the school). The scatter plot, shown in **Appendix E**, certifies that student who had positive perception about their school also had a good performance on their academic achievements.

10) Imagine that you are working for the New York City Department of Education as a data scientist (like one of my former students). What actionable recommendations would you make on how to improve schools so that they a) send more students to HSPHS and b) improve objective measures or achievement.

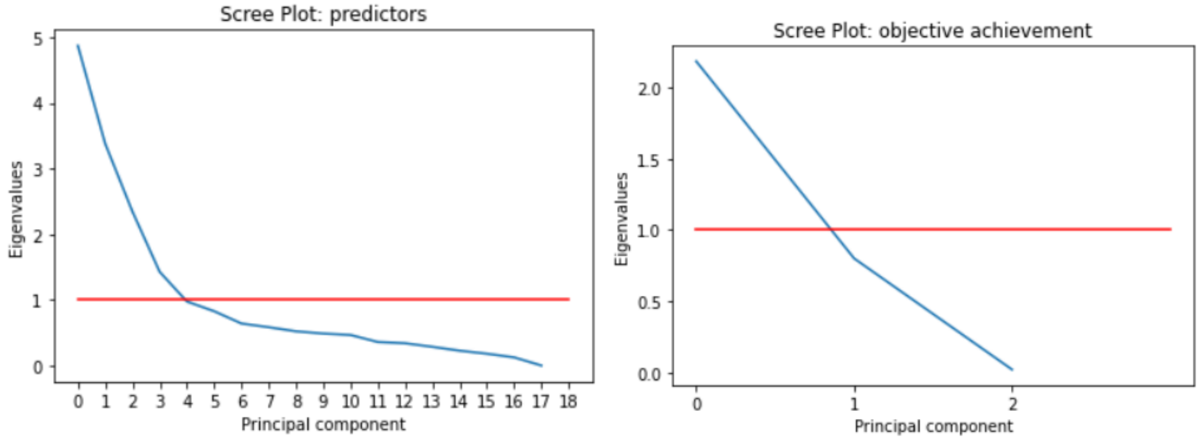
Based on my research result, I would recommend the New York City Department of Education to spend more time on improving the school climate factors by providing adequate learning environment for the students. Such improvement could be done through holding more extra-help sessions for the students or creating more collaborative activities to build the trust among the school community. These effort will significantly improve the objective measures of student achievement that will also impact admission rate to HSPHS positively. Also, sending more applications to the HSPHS is also necessary to increase the chance of sending more students. My research shows that large school tended to have more student to be sent to HSPHS, and the prime factor that allowed them to do so was sending a lot of applications. If the students give up, they will automatically have 0 chance to get in. So encourage the students to send in applications with the confidence, do not discourage them!

These are the data-based recommendations I could make for improving the schools.

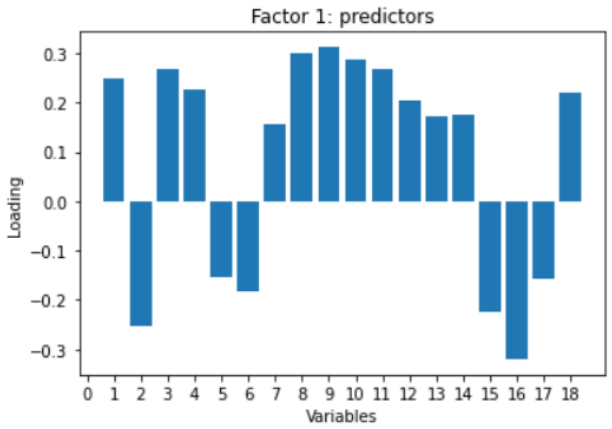
Appendix A: Correlation Matrix shown in Color Plot - Predictors & Objective Achievement

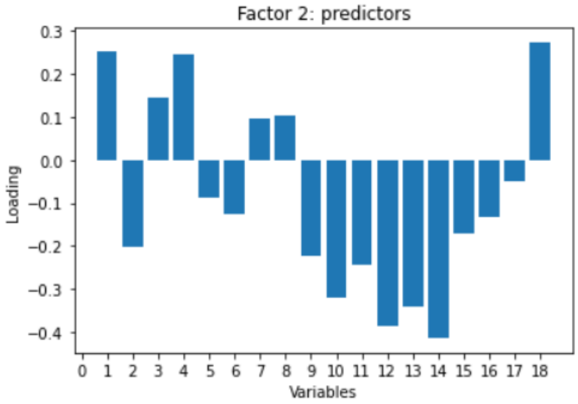


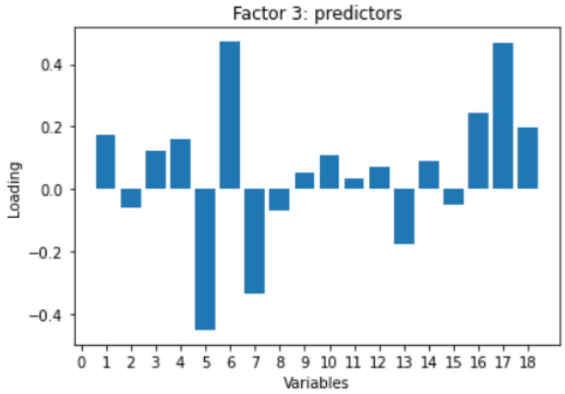
Appendix B: Scree Plot - Predictors & Objective Achievement

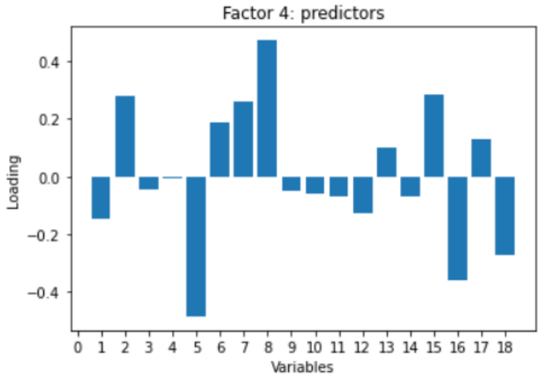


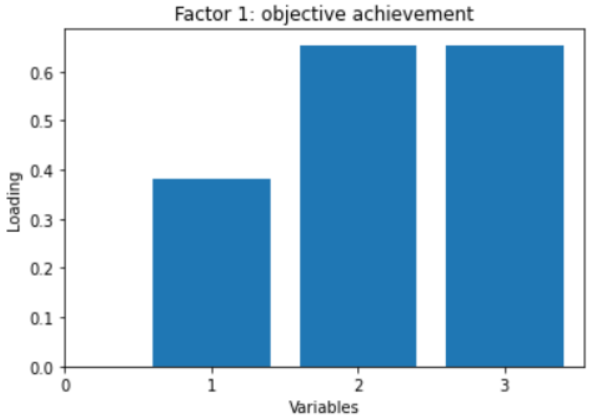
Appendix C: Factor bar plot - Predictors & Objective Achievement



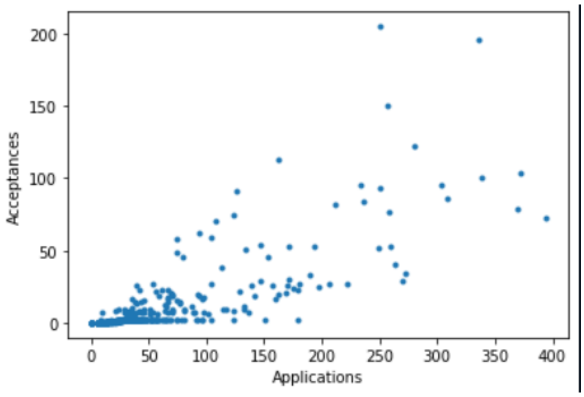




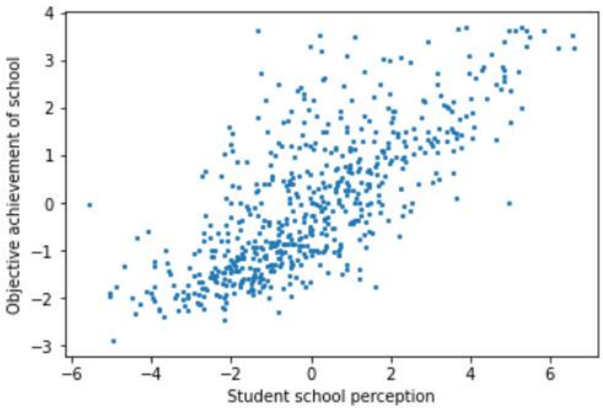




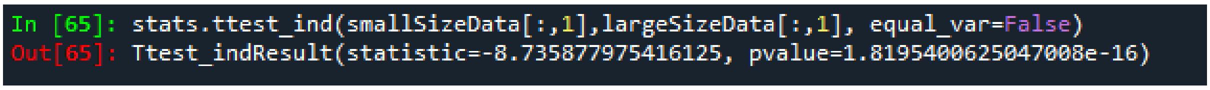
Appendix D: Scatter plot (acceptances vs. applications)



Appendix E: Scatter plot (student school perception vs. objective achievement of school)

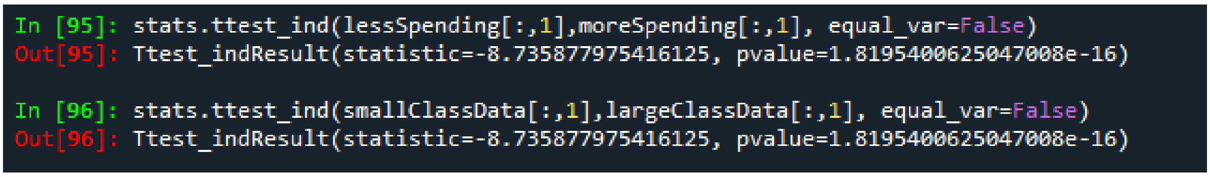


Appendix F: Two Sample T Test Results (School Size - small/large)



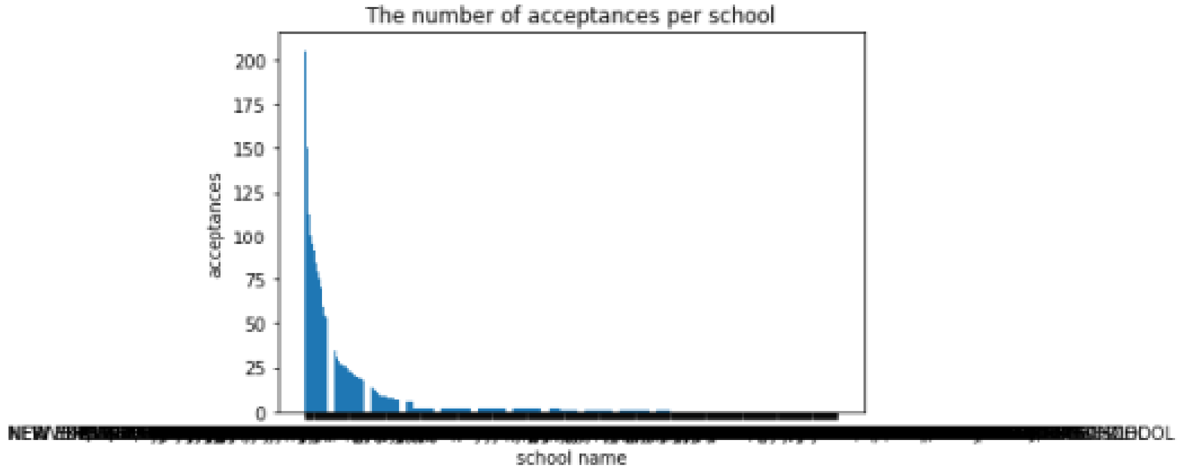
```
In [65]: stats.ttest_ind(smallSizeData[:,1],largeSizeData[:,1], equal_var=False)
Out[65]: Ttest_indResult(statistic=-8.735877975416125, pvalue=1.8195400625047008e-16)
```

Appendix G: Two Sample T Test Results (Per student spenidng, Class Size)

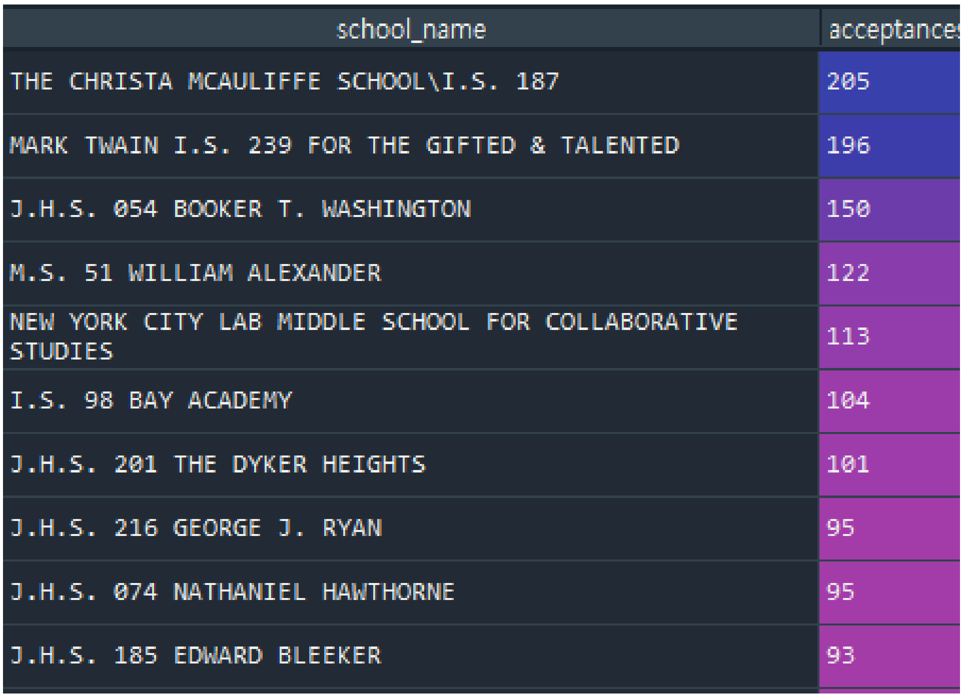


```
In [95]: stats.ttest_ind(lessSpending[:,1],moreSpending[:,1], equal_var=False)
Out[95]: Ttest_indResult(statistic=-8.735877975416125, pvalue=1.8195400625047008e-16)

In [96]: stats.ttest_ind(smallClassData[:,1],largeClassData[:,1], equal_var=False)
Out[96]: Ttest_indResult(statistic=-8.735877975416125, pvalue=1.8195400625047008e-16)
```

Appendix H: A bar plot of schools, rank-ordered by decreasing number of acceptances

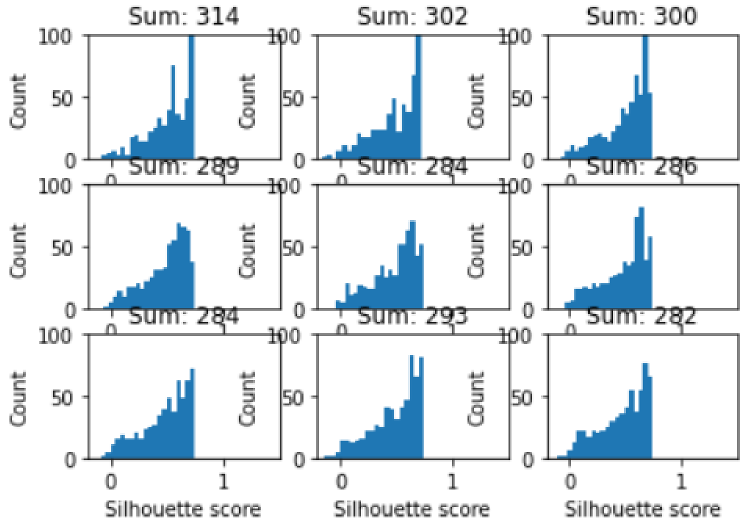


Appendix I: A table containing first 10 rank of schools to visualize school name of bar plot in **Appendix H.**

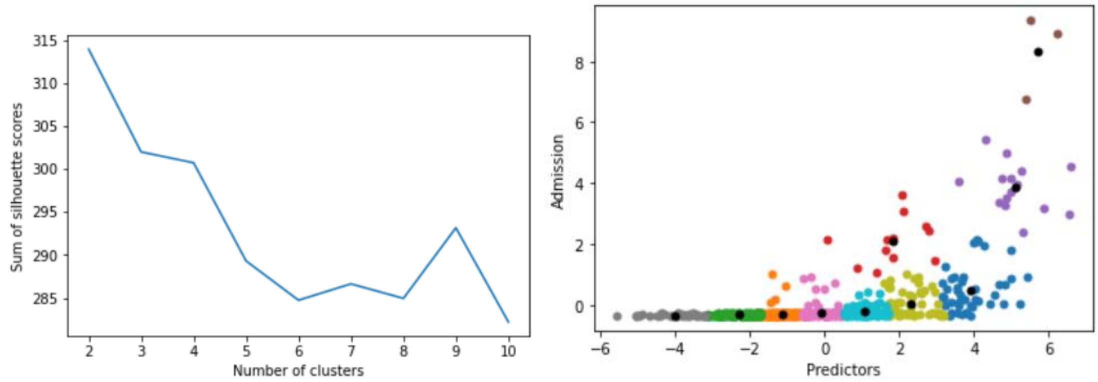


school_name	acceptance
THE CHRISTA MCAULIFFE SCHOOL\I.S. 187	205
MARK TWAIN I.S. 239 FOR THE GIFTED & TALENTED	196
J.H.S. 054 BOOKER T. WASHINGTON	150
M.S. 51 WILLIAM ALEXANDER	122
NEW YORK CITY LAB MIDDLE SCHOOL FOR COLLABORATIVE STUDIES	113
I.S. 98 BAY ACADEMY	104
J.H.S. 201 THE DYKER HEIGHTS	101
J.H.S. 216 GEORGE J. RYAN	95
J.H.S. 074 NATHANIEL HAWTHORNE	95
J.H.S. 185 EDWARD BLEEKER	93

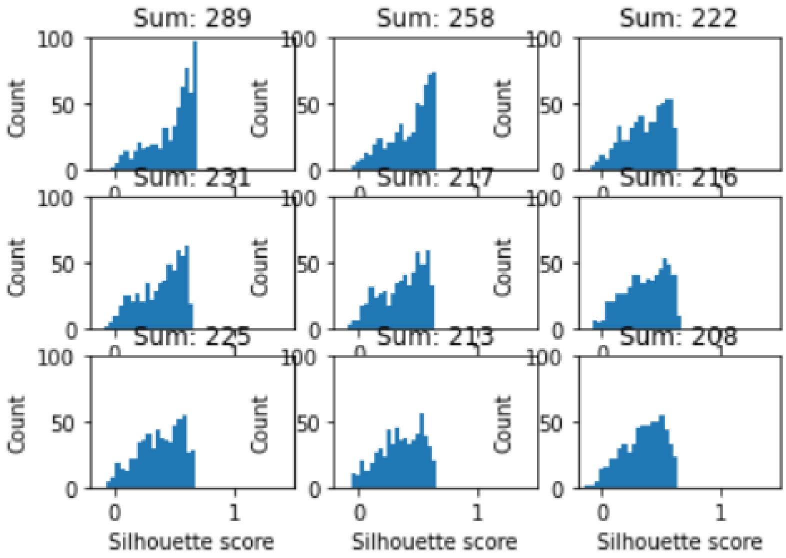
Appendix J: histograms showing the silhouette score of Kmeans clustering: predictors & admission



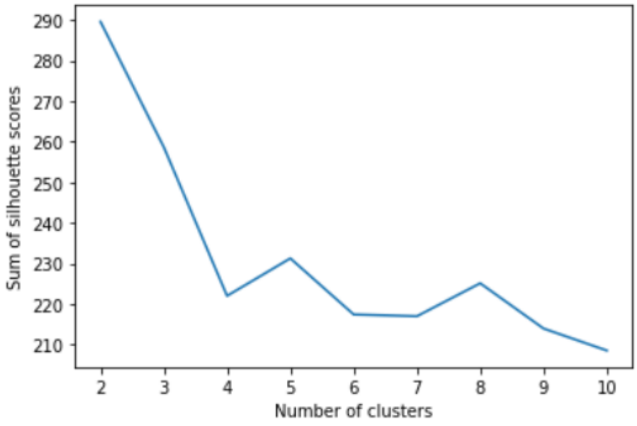
Appendix L: silhouette coefficient and scatter plot: predictors & admission

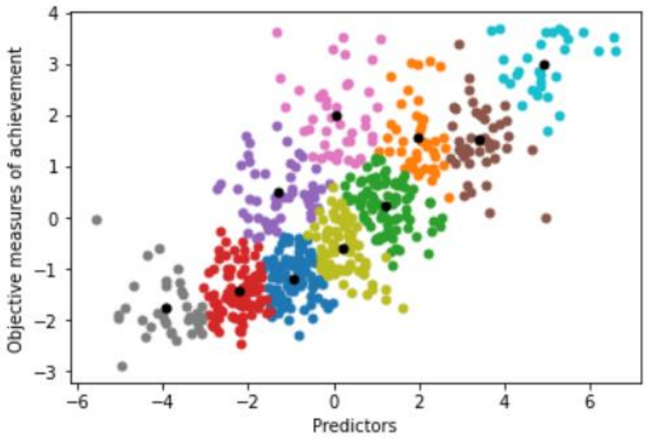


Appendix M: histograms showing the silhouette score of Kmeans clustering: predictors & objective measures of achievement

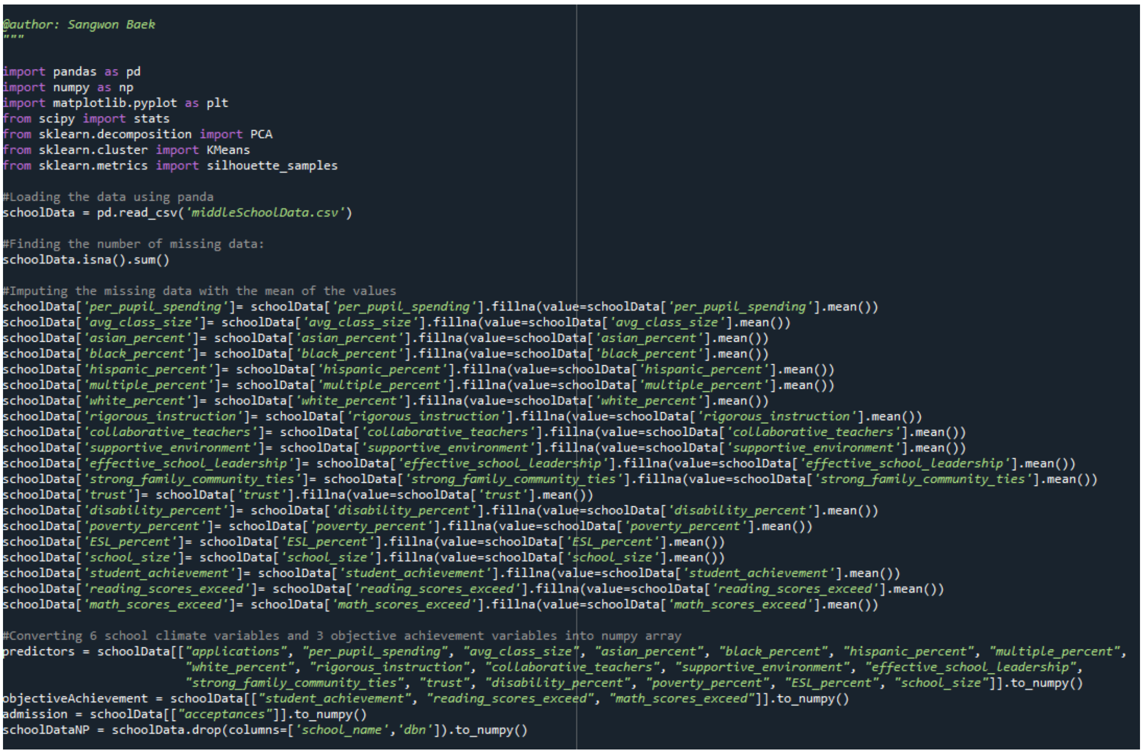


Appendix N: silhouette coefficient plot: predictors & objective measures of achievement

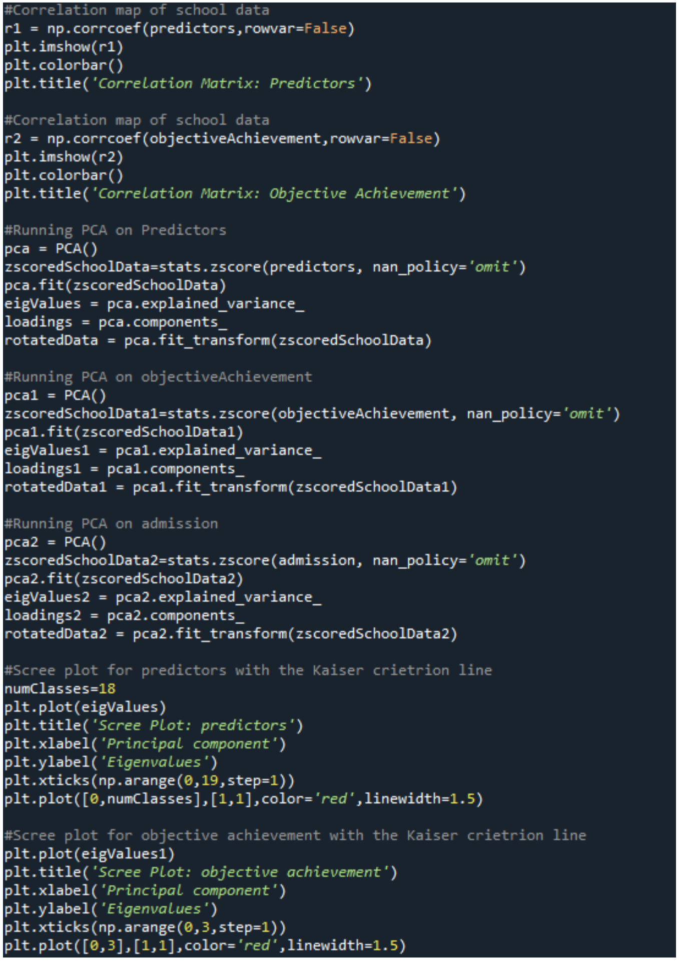




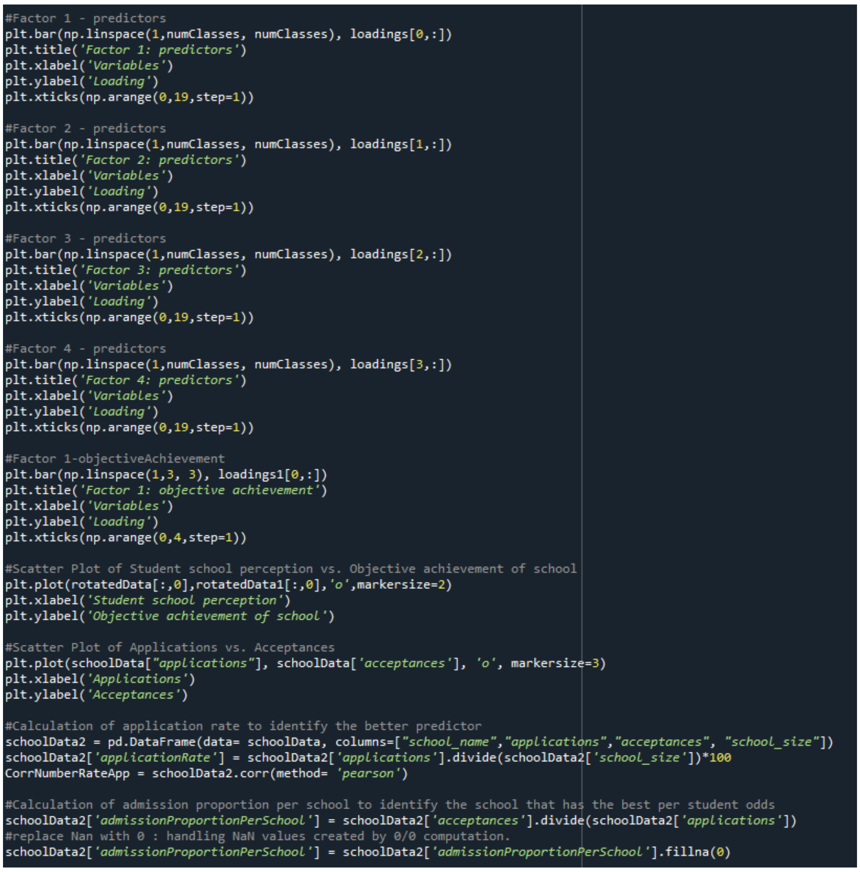
Appendix O: Code used for the project



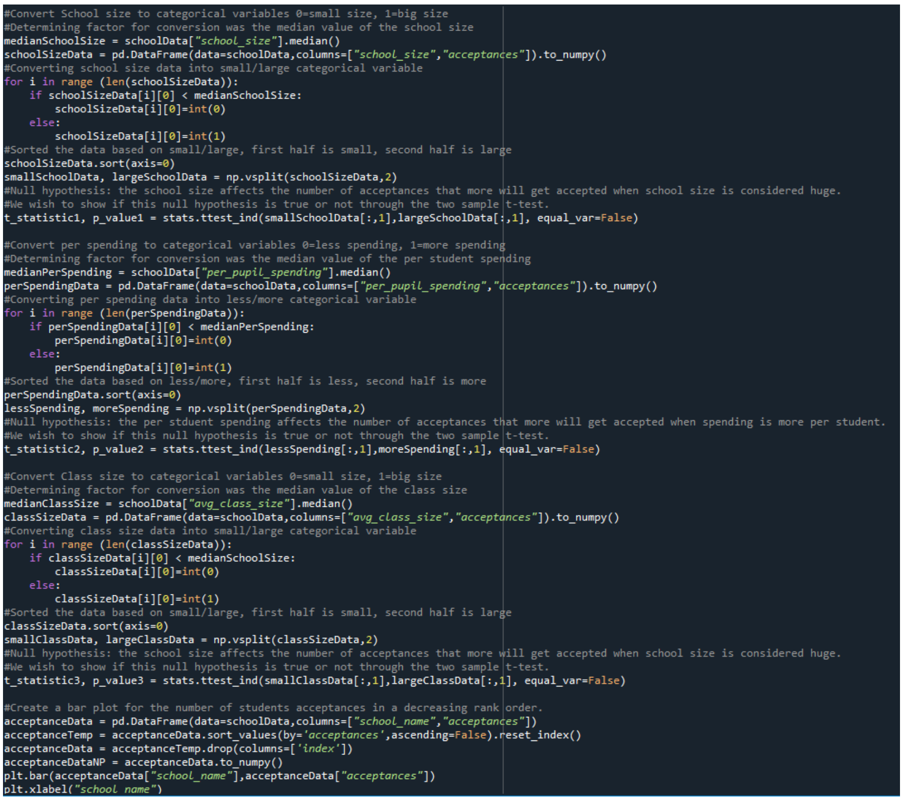
```
@author: Sangwon Baek
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_samples

#Loading the data using panda
schoolData = pd.read_csv('middleSchoolData.csv')

#Finding the number of missing data:
schoolData.isna().sum()

#Imputing the missing data with the mean of the values
schoolData['per_pupil_spending']= schoolData['per_pupil_spending'].fillna(value=schoolData['per_pupil_spending'].mean())
schoolData['avg_class_size']= schoolData['avg_class_size'].fillna(value=schoolData['avg_class_size'].mean())
schoolData['asian_percent']= schoolData['asian_percent'].fillna(value=schoolData['asian_percent'].mean())
schoolData['black_percent']= schoolData['black_percent'].fillna(value=schoolData['black_percent'].mean())
schoolData['hispanic_percent']= schoolData['hispanic_percent'].fillna(value=schoolData['hispanic_percent'].mean())
schoolData['multiple_percent']= schoolData['multiple_percent'].fillna(value=schoolData['multiple_percent'].mean())
schoolData['white_percent']= schoolData['white_percent'].fillna(value=schoolData['white_percent'].mean())
schoolData['rigorous_instruction']= schoolData['rigorous_instruction'].fillna(value=schoolData['rigorous_instruction'].mean())
schoolData['collaborative_teachers']= schoolData['collaborative_teachers'].fillna(value=schoolData['collaborative_teachers'].mean())
schoolData['supportive_environment']= schoolData['supportive_environment'].fillna(value=schoolData['supportive_environment'].mean())
schoolData['effective_school_leadership']= schoolData['effective_school_leadership'].fillna(value=schoolData['effective_school_leadership'].mean())
schoolData['strong_family_community_ties']= schoolData['strong_family_community_ties'].fillna(value=schoolData['strong_family_community_ties'].mean())
schoolData['trust']= schoolData['trust'].fillna(value=schoolData['trust'].mean())
schoolData['disability_percent']= schoolData['disability_percent'].fillna(value=schoolData['disability_percent'].mean())
schoolData['poverty_percent']= schoolData['poverty_percent'].fillna(value=schoolData['poverty_percent'].mean())
schoolData['ESL_percent']= schoolData['ESL_percent'].fillna(value=schoolData['ESL_percent'].mean())
schoolData['school_size']= schoolData['school_size'].fillna(value=schoolData['school_size'].mean())
schoolData['student_achievement']= schoolData['student_achievement'].fillna(value=schoolData['student_achievement'].mean())
schoolData['reading_scores_exceed']= schoolData['reading_scores_exceed'].fillna(value=schoolData['reading_scores_exceed'].mean())
schoolData['math_scores_exceed']= schoolData['math_scores_exceed'].fillna(value=schoolData['math_scores_exceed'].mean())

#Converting 6 school climate variables and 3 objective achievement variables into numpy array
predictors = schoolData[["applications", "per_pupil_spending", "avg_class_size", "asian_percent", "black_percent", "hispanic_percent", "multiple_percent",
                         "white_percent", "rigorous_instruction", "collaborative_teachers", "supportive_environment", "effective_school_leadership",
                         "strong_family_community_ties", "trust", "disability_percent", "poverty_percent", "ESL_percent", "school_size"]].to_numpy()
objectiveAchievement = schoolData[["student_achievement", "reading_scores_exceed", "math_scores_exceed"]].to_numpy()
admission = schoolData[["acceptances"]].to_numpy()
schoolDataNP = schoolData.drop(columns=['school_name','dbn']).to_numpy()
```



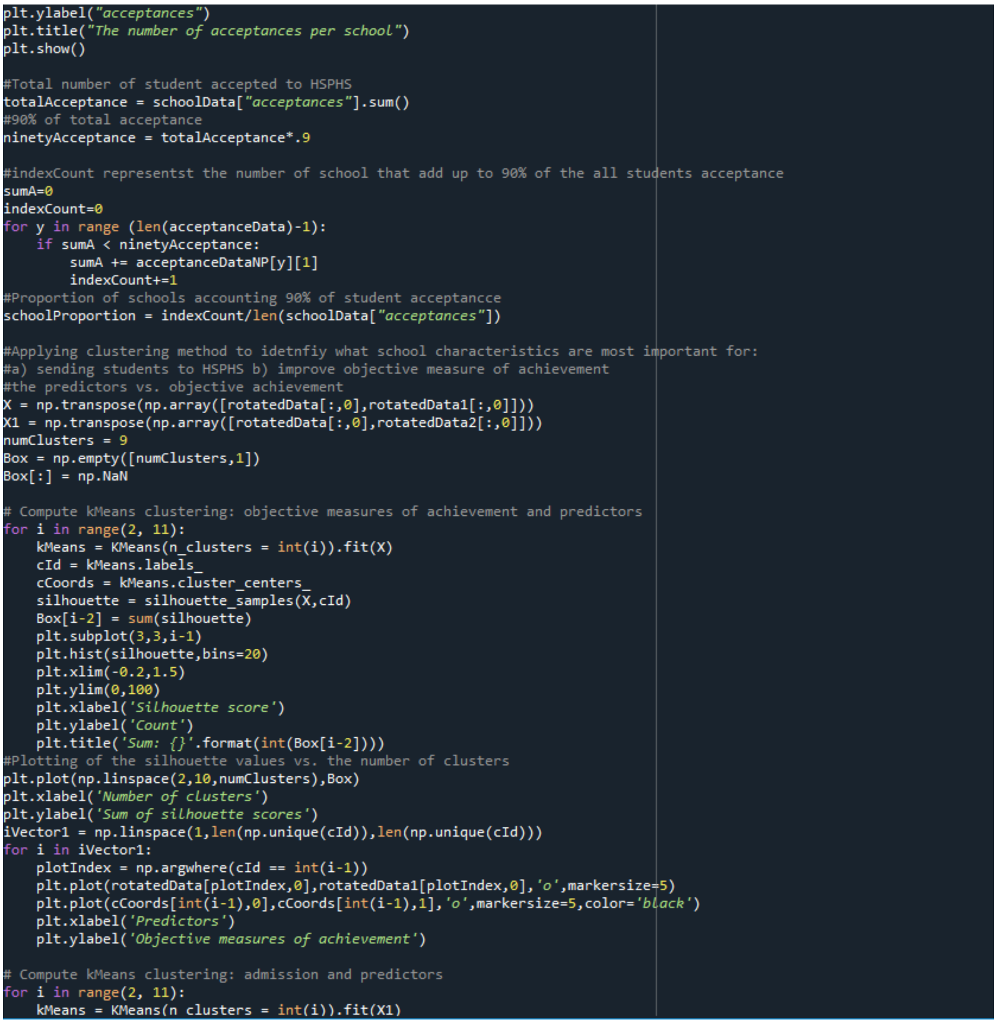
```
#Correlation map of school data
r1 = np.corrcoef(predictors,rowvar=False)
plt.imshow(r1)
plt.colorbar()
plt.title('Correlation Matrix: Predictors')

#Correlation map of school data
r2 = np.corrcoef(objectiveAchievement,rowvar=False)
plt.imshow(r2)
plt.colorbar()
plt.title('Correlation Matrix: Objective Achievement')

#Running PCA on Predictors
pca = PCA()
zscoredSchoolData=stats.zscore(predictors, nan_policy='omit')
pca.fit(zscoredSchoolData)
eigValues = pca.explained_variance_
loadings = pca.components_
rotatedData = pca.fit_transform(zscoredSchoolData)

#Running PCA on objectiveAchievement
pca1 = PCA()
zscoredSchoolData1=stats.zscore(objectiveAchievement, nan_policy='omit')
pca1.fit(zscoredSchoolData1)
eigValues1 = pca1.explained_variance_
loadings1 = pca1.components_
rotatedData1 = pca1.fit_transform(zscoredSchoolData1)

#Running PCA on admission
pca2 = PCA()
zscoredSchoolData2=stats.zscore(admission, nan_policy='omit')
pca2.fit(zscoredSchoolData2)
eigValues2 = pca2.explained_variance_
loadings2 = pca2.components_
rotatedData2 = pca2.fit_transform(zscoredSchoolData2)

#Scree plot for predictors with the Kaiser crietrion line
numClasses=18
plt.plot(eigValues)
plt.title('Scree Plot: predictors')
plt.xlabel('Principal component')
plt.ylabel('Eigenvalues')
plt.xticks(np.arange(0,19,step=1))
plt.plot([0,numClasses],[1,1],color='red',linewidth=1.5)

#Scree plot for objective achievement with the Kaiser crietrion line
plt.plot(eigValues1)
plt.title('Scree Plot: objective achievement')
plt.xlabel('Principal component')
plt.ylabel('Eigenvalues')
plt.xticks(np.arange(0,3,step=1))
plt.plot([0,3],[1,1],color='red',linewidth=1.5)
```



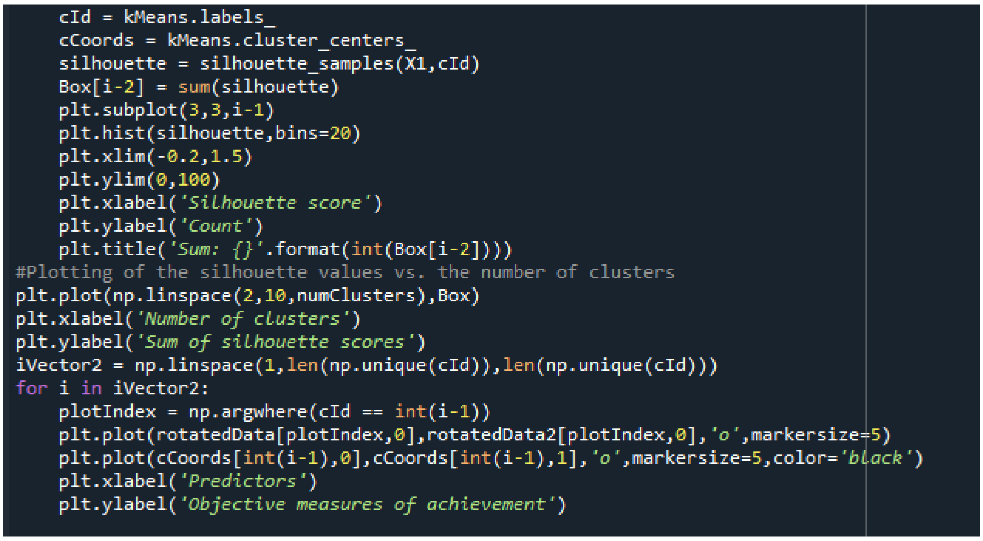
```
#Factor 1 - predictors
plt.bar(np.linspace(1,numClasses, numClasses), loadings[0,:])
plt.title('Factor 1: predictors')
plt.xlabel('Variables')
plt.ylabel('Loading')
plt.xticks(np.arange(0,19,step=1))

#Factor 2 - predictors
plt.bar(np.linspace(1,numClasses, numClasses), loadings[1,:])
plt.title('Factor 2: predictors')
plt.xlabel('Variables')
plt.ylabel('Loading')
plt.xticks(np.arange(0,19,step=1))

#Factor 3 - predictors
plt.bar(np.linspace(1,numClasses, numClasses), loadings[2,:])
plt.title('Factor 3: predictors')
plt.xlabel('Variables')
plt.ylabel('Loading')
plt.xticks(np.arange(0,19,step=1))

#Factor 4 - predictors
plt.bar(np.linspace(1,numClasses, numClasses), loadings[3,:])
plt.title('Factor 4: predictors')
plt.xlabel('Variables')
plt.ylabel('Loading')
plt.xticks(np.arange(0,19,step=1))

#Factor 1-objectiveAchievement
plt.bar(np.linspace(1,3, 3), loadings1[0,:])
plt.title('Factor 1: objective achievement')
plt.xlabel('Variables')
plt.ylabel('Loading')
plt.xticks(np.arange(0,4,step=1))

#Scatter Plot of Student school perception vs. Objective achievement of school
plt.plot(rotatedData[:,0],rotatedData1[:,0],'o',markersize=2)
plt.xlabel('Student school perception')
plt.ylabel('Objective achievement of school')

#Scatter Plot of Applications vs. Acceptances
plt.plot(schoolData["applications"], schoolData['acceptances'], 'o', markersize=3)
plt.xlabel('Applications')
plt.ylabel('Acceptances')

#Calculation of application rate to identify the better predictor
schoolData2 = pd.DataFrame(data= schoolData, columns=["school_name","applications","acceptances", "school_size"])
schoolData2['applicationRate'] = schoolData2['applications'].divide(schoolData2['school_size'])*100
CorrNumberRateApp = schoolData2.corr(method= 'pearson')

#Calculation of admission proportion per school to identify the school that has the best per student odds
schoolData2['admissionProportionPerSchool'] = schoolData2['acceptances'].divide(schoolData2['applications'])
#replace Nan with 0 : handling NaN values created by 0/0 computation.
schoolData2['admissionProportionPerSchool'] = schoolData2['admissionProportionPerSchool'].fillna(0)
```



```
#Convert School size to categorical variables 0=small size, 1=big size
#Determining factor for conversion was the median value of the school size
medianSchoolSize = schoolData["school_size"].median()
schoolSizeData = pd.DataFrame(data=schoolData,columns=["school_size","acceptances"]).to_numpy()
#Converting school size data into small/large categorical variable
for i in range (len(schoolSizeData)):
    if schoolSizeData[i][0] < medianSchoolSize:
        schoolSizeData[i][0]=int(0)
    else:
        schoolSizeData[i][0]=int(1)
#Sorted the data based on small/large, first half is small, second half is large
schoolSizeData.sort(axis=0)
smallSchoolData, largeSchoolData = np.vsplit(schoolSizeData,2)
#Null hypothesis: the school size affects the number of acceptances that more will get accepted when school size is considered huge.
#We wish to show if this null hypothesis is true or not through the two sample t-test.
t_statistic1, p_value1 = stats.ttest_ind(smallSchoolData[:,1],largeSchoolData[:,1], equal_var=False)

#Convert per spending to categorical variables 0=less spending, 1=more spending
#Determining factor for conversion was the median value of the per student spending
medianPerSpending = schoolData["per_pupil_spending"].median()
perSpendingData = pd.DataFrame(data=schoolData,columns=["per_pupil_spending","acceptances"]).to_numpy()
#Converting per spending data into less/more categorical variable
for i in range (len(perSpendingData)):
    if perSpendingData[i][0] < medianPerSpending:
        perSpendingData[i][0]=int(0)
    else:
        perSpendingData[i][0]=int(1)
#Sorted the data based on less/more, first half is less, second half is more
perSpendingData.sort(axis=0)
lessSpending, moreSpending = np.vsplit(perSpendingData,2)
#Null hypothesis: the per stduent spending affects the number of acceptances that more will get accepted when spending is more per student.
#We wish to show if this null hypothesis is true or not through the two sample t-test.
t_statistic2, p_value2 = stats.ttest_ind(lessSpending[:,1],moreSpending[:,1], equal_var=False)

#Convert Class size to categorical variables 0=small size, 1=big size
#Determining factor for conversion was the median value of the class size
medianClassSize = schoolData["avg_class_size"].median()
classSizeData = pd.DataFrame(data=schoolData,columns=["avg_class_size","acceptances"]).to_numpy()
#Converting class size data into small/large categorical variable
for i in range (len(classSizeData)):
    if classSizeData[i][0] < medianSchoolSize:
        classSizeData[i][0]=int(0)
    else:
        classSizeData[i][0]=int(1)
#Sorted the data based on small/large, first half is small, second half is large
classSizeData.sort(axis=0)
smallClassData, largeClassData = np.vsplit(classSizeData,2)
#Null hypothesis: the school size affects the number of acceptances that more will get accepted when school size is considered huge.
#We wish to show if this null hypothesis is true or not through the two sample t-test.
t_statistic3, p_value3 = stats.ttest_ind(smallClassData[:,1],largeClassData[:,1], equal_var=False)

#Create a bar plot for the number of students acceptances in a decreasing rank order.
acceptanceData = pd.DataFrame(data=schoolData,columns=["school_name","acceptances"])
acceptanceTemp = acceptanceData.sort_values(by='acceptances',ascending=False).reset_index()
acceptanceData = acceptanceTemp.drop(columns=['index'])
acceptanceDataNP = acceptanceData.to_numpy()
plt.bar(acceptanceData["school_name"],acceptanceData["acceptances"])
plt.xlabel("school_name")
```



```
plt.ylabel("acceptances")
plt.title("The number of acceptances per school")
plt.show()

#Total number of student accepted to HSPHS
totalAcceptance = schoolData["acceptances"].sum()
#90% of total acceptance
ninetyAcceptance = totalAcceptance*.9

#indexCount representst the number of school that add up to 90% of the all students acceptance
sumA=0
indexCount=0
for y in range (len(acceptanceData)-1):
    if sumA < ninetyAcceptance:
        sumA += acceptanceDataNP[y][1]
        indexCount+=1
#Proportion of schools accounting 90% of student acceptancce
schoolProportion = indexCount/len(schoolData["acceptances"])

#Applying clustering method to idetnfiy what school characteristics are most important for:
#a) sending students to HSPHS b) improve objective measure of achievement
#the predictors vs. objective achievement
X = np.transpose(np.array([rotatedData[:,0],rotatedData1[:,0]]))
X1 = np.transpose(np.array([rotatedData[:,0],rotatedData2[:,0]]))
numClusters = 9
Box = np.empty([numClusters,1])
Box[:] = np.NaN

# Compute kMeans clustering: objective measures of achievement and predictors
for i in range(2, 11):
    kMeans = KMeans(n_clusters = int(i)).fit(X)
    cId = kMeans.labels_
    cCoords = kMeans.cluster_centers_
    silhouette = silhouette_samples(X,cId)
    Box[i-2] = sum(silhouette)
    plt.subplot(3,3,i-1)
    plt.hist(silhouette,bins=20)
    plt.xlim(-0.2,1.5)
    plt.ylim(0,100)
    plt.xlabel('Silhouette score')
    plt.ylabel('Count')
    plt.title('Sum: {}'.format(int(Box[i-2])))
#Plotting of the silhouette values vs. the number of clusters
plt.plot(np.linspace(2,10,numClusters),Box)
plt.xlabel('Number of clusters')
plt.ylabel('Sum of silhouette scores')
iVector1 = np.linspace(1,len(np.unique(cId)),len(np.unique(cId)))
for i in iVector1:
    plotIndex = np.argwhere(cId == int(i-1))
    plt.plot(rotatedData[plotIndex,0],rotatedData1[plotIndex,0],'o',markersize=5)
    plt.plot(cCoords[int(i-1),0],cCoords[int(i-1),1],'o',markersize=5,color='black')
    plt.xlabel('Predictors')
    plt.ylabel('Objective measures of achievement')

# Compute kMeans clustering: admission and predictors
for i in range(2, 11):
    kMeans = KMeans(n_clusters = int(i)).fit(X1)
```



```
    cId = kMeans.labels_
    cCoords = kMeans.cluster_centers_
    silhouette = silhouette_samples(X1,cId)
    Box[i-2] = sum(silhouette)
    plt.subplot(3,3,i-1)
    plt.hist(silhouette,bins=20)
    plt.xlim(-0.2,1.5)
    plt.ylim(0,100)
    plt.xlabel('Silhouette score')
    plt.ylabel('Count')
    plt.title('Sum: {}'.format(int(Box[i-2])))
#Plotting of the silhouette values vs. the number of clusters
plt.plot(np.linspace(2,10,numClusters),Box)
plt.xlabel('Number of clusters')
plt.ylabel('Sum of silhouette scores')
iVector2 = np.linspace(1,len(np.unique(cId)),len(np.unique(cId)))
for i in iVector2:
    plotIndex = np.argwhere(cId == int(i-1))
    plt.plot(rotatedData[plotIndex,0],rotatedData2[plotIndex,0],'o',markersize=5)
    plt.plot(cCoords[int(i-1),0],cCoords[int(i-1),1],'o',markersize=5,color='black')
    plt.xlabel('Predictors')
    plt.ylabel('Objective measures of achievement')
```