Project 1

In addition to answering the bolded questions on Coursera, also attach your notebook, both as `.ipynb` and `.html`.

In this assignment, we will be using PennGrader, a Python package built by a former TA for autograding Python notebooks. PennGrader was developed to provide students with instant feedback on their answer. You can submit your answer and know whether it's right or wrong instantly. We then record your most recent answer in our backend database. You will have 100 attempts per test case, which should be more than sufficient.

NOTE : Please remember to remove the

```
raise notImplementedError
```

after your implementation, otherwise the cell will not compile.

Getting Setup

Please run the below cells to get setup with the autograder. If you need to install packages, please uncomment and try the following lines; if they do not work, please try running them in the terminal without the `!` sign.

In [29]:
```
# %%capture
# !pip install penngrader --user
# !pip install seaborn --user
```

Let's try PennGrader out! Fill in the cell below with your PennID and then run the following cell to initialize the grader.

In [30]:
```
#PLEASE ENSURE YOUR STUDENT_ID IS ENTERED AS AN INT (NOT A STRING). IF NOT, THE AUTOGRADER W
ON'T KNOW WHO
#TO ASSIGN POINTS TO YOU IN OUR BACKEND

STUDENT_ID = 49731093                    # YOUR 8-DIGIT PENNID GOES HERE
STUDENT_NAME = "Newman Alexander Ilgenfritz"     # YOUR FULL NAME GOES HERE
```

In [31]:
```
import penngrader.grader

grader = penngrader.grader.PennGrader(homework_id = 'ESE542_Online_Spring_2021_HW1', student
_id = STUDENT_ID)
```

Imports

It is important for all (or most) imports to go on the top of a notebook so that other users know which packages need to be installed. In projects that use Anaconda, it is also common to see a file named `requirements.txt` listing all the packages that one has to install.

1. First, import all the necessary modules using the import function. For this exercise, we will be mainly using `pandas`. In the future, we will also be using `numpy`, `seaborn`, and `matplotlib`. To learn more about these packages, you can read through the documentation:
   - https://pandas.pydata.org/
   - https://numpy.org/
   - https://seaborn.pydata.org/
   - https://matplotlib.org/

In [32]:
```
# Let's import the relevant Python packages here
# Feel free to import any other packages for this project

import os
import sys

#Data Wrangling
import numpy as np
import pandas as pd

#Plotting
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

import csv

%matplotlib inline

import distutils
from distutils import util
```

Data

In the following exercise, we will familiarize ourselves with Python by studying the College dataset, which can be found in the file `College.csv`. This dataset contains the following variables from 777 different universities and colleges in the US:

Column	Description
Private	Public/private indicator
Apps	Number of applications received
Accept	Number of applicants accepted
Enroll	Number of new students enrolled
Top10perc	New students from top 10\% of high school class
Top25perc	New students from top 25\% of high school class
F.Undergrad	Number of full-time undergraduates
P.Undergrad	Number of part-time undergraduates
Outstate	Out-of-state tuition
Room.Board	Room and board costs
Books	Estimated book costs
Personal	Estimated personal spending
PhD	Percent of faculty with Ph.D.'s
Terminal	Percent of faculty with terminal degree
S.F.Ratio	Student/faculty ratio
Perc.alumni	Percent of alumni who donate
Expend	Instructional expenditure per student
Grad.Rate	Graduation rate

1. Load the College dataset using `pandas`

In [33]:
```
collegeDF = pd.read_csv('College.csv')
```

1. Use the `head()` function to view the data.

In [34]:
```
print(collegeDF.head(), '\n')
```

```
  Private  Apps  Accept  Enroll  Top10perc  Top25perc  F.Undergrad  \
0     Yes  1660    1232     721         23         52         2885
1     Yes  2186    1924     512         16         29         2683
2     Yes  1428    1097     336         22         50         1036
3     Yes   417     349     137         60         89          510
4     Yes   193     146      55         16         44          249

   P.Undergrad  Outstate  Room.Board  Books  Personal  PhD  Terminal  \
0          537      7440        3300    450      2200   70        78
1         1227     12280        6450    750      1500   29        30
2           99     11250        3750    400      1165   53        66
3           63     12960        5450    450       875   92        97
4          869      7560        4120    800      1500   76        72

   S.F.Ratio  perc.alumni  Expend  Grad.Rate                         Names
0       18.1           12    7041         60  Abilene Christian University
1       12.2           16   10527         56            Adelphi University
2       12.9           30    8735         54                Adrian College
3        7.7           37   19016         59             Agnes Scott College
4       11.9            2   10922         15     Alaska Pacific University
```

1. Notice that there is a column 'Names' of each university's name. As we don't want to use these names as predictors, they are natural candidates to index our data. We can do this using the following function:

```
#college is the dataframe
college.set_index('Names', inplace = True)
```

In [35]:
```
collegeDF.set_index ("Names", inplace = True )
```

1. Use the `head()` function again. You should now see that the indices have been replaced with the name of each university in the data set. This means that Python has given each row a name corresponding to the appropriate university. Python will not try to perform calculations on the row names.

In [36]:
```
print('df head:')
print(collegeDF.head(), '\n')
```

```
df head:
                             Private  Apps  Accept  Enroll  Top10perc  \
Names
Abilene Christian University     Yes  1660    1232     721         23
Adelphi University               Yes  2186    1924     512         16
Adrian College                   Yes  1428    1097     336         22
Agnes Scott College              Yes   417     349     137         60
Alaska Pacific University        Yes   193     146      55         16

                              Top25perc  F.Undergrad  P.Undergrad  Outstate  \
Names
Abilene Christian University         52         2885          537      7440
Adelphi University                   29         2683         1227     12280
Adrian College                       50         1036           99     11250
Agnes Scott College                  89          510           63     12960
Alaska Pacific University            44          249          869      7560

                              Room.Board  Books  Personal  PhD  Terminal  \
Names
Abilene Christian University        3300    450      2200   70        78
Adelphi University                  6450    750      1500   29        30
Adrian College                      3750    400      1165   53        66
Agnes Scott College                 5450    450       875   92        97
Alaska Pacific University           4120    800      1500   76        72

                              S.F.Ratio  perc.alumni  Expend  Grad.Rate
Names
Abilene Christian University       18.1           12    7041         60
Adelphi University                 12.2           16   10527         56
Adrian College                     12.9           30    8735         54
Agnes Scott College                 7.7           37   19016         59
Alaska Pacific University          11.9            2   10922         15
```

1. Use the `info()` function to check and produce a numerical summary of your variables.

In [37]:
```
print('df.info()')
print(collegeDF.info(), '\n')
```

```
df.info()
<class 'pandas.core.frame.DataFrame'>
Index: 777 entries, Abilene Christian University to York College of Pennsylvania
Data columns (total 18 columns):
Private        777 non-null object
Apps           777 non-null int64
Accept         777 non-null int64
Enroll         777 non-null int64
Top10perc      777 non-null int64
Top25perc      777 non-null int64
F.Undergrad    777 non-null int64
P.Undergrad    777 non-null int64
Outstate       777 non-null int64
Room.Board     777 non-null int64
Books          777 non-null int64
Personal       777 non-null int64
PhD            777 non-null int64
Terminal       777 non-null int64
S.F.Ratio      777 non-null float64
perc.alumni    777 non-null int64
Expend         777 non-null int64
Grad.Rate      777 non-null int64
dtypes: float64(1), int64(16), object(1)
memory usage: 115.3+ KB
None
```

1. Examine if there are any duplicates and drop them if needed. Hint: It is perfectly fine to observe no duplicates in the dataset, but it is good practice to check anyhow. Before dropping duplicates, it is also good practice to check whether the duplicates contained any different data, since data entry errors frequently occur.

In [38]:
```
collegeDF.drop_duplicates(keep = False, inplace = True)
print('after dropping duplicate rows, df head:')
print(collegeDF.head(), '\n')
print('after dropping duplicate rows, df.info()')
print(collegeDF.info(), '\n')
```

```
after dropping duplicate rows, df head:
                             Private  Apps  Accept  Enroll  Top10perc  \
Names
Abilene Christian University     Yes  1660    1232     721         23
Adelphi University               Yes  2186    1924     512         16
Adrian College                   Yes  1428    1097     336         22
Agnes Scott College              Yes   417     349     137         60
Alaska Pacific University        Yes   193     146      55         16

                              Top25perc  F.Undergrad  P.Undergrad  Outstate  \
Names
Abilene Christian University         52         2885          537      7440
Adelphi University                   29         2683         1227     12280
Adrian College                       50         1036           99     11250
Agnes Scott College                  89          510           63     12960
Alaska Pacific University            44          249          869      7560

                              Room.Board  Books  Personal  PhD  Terminal  \
Names
Abilene Christian University        3300    450      2200   70        78
Adelphi University                  6450    750      1500   29        30
Adrian College                      3750    400      1165   53        66
Agnes Scott College                 5450    450       875   92        97
Alaska Pacific University           4120    800      1500   76        72

                              S.F.Ratio  perc.alumni  Expend  Grad.Rate
Names
Abilene Christian University       18.1           12    7041         60
Adelphi University                 12.2           16   10527         56
Adrian College                     12.9           30    8735         54
Agnes Scott College                 7.7           37   19016         59
Alaska Pacific University          11.9            2   10922         15

after dropping duplicate rows, df.info()
<class 'pandas.core.frame.DataFrame'>
Index: 777 entries, Abilene Christian University to York College of Pennsylvania
Data columns (total 18 columns):
Private        777 non-null object
Apps           777 non-null int64
Accept         777 non-null int64
Enroll         777 non-null int64
Top10perc      777 non-null int64
Top25perc      777 non-null int64
F.Undergrad    777 non-null int64
P.Undergrad    777 non-null int64
Outstate       777 non-null int64
Room.Board     777 non-null int64
Books          777 non-null int64
Personal       777 non-null int64
PhD            777 non-null int64
Terminal       777 non-null int64
S.F.Ratio      777 non-null float64
perc.alumni    777 non-null int64
Expend         777 non-null int64
Grad.Rate      777 non-null int64
dtypes: float64(1), int64(16), object(1)
memory usage: 115.3+ KB
None
```

1. Remove instances where there is no value for the 'Apps' column, if any, and replace them with 0. This dataframe will henceforth be our original dataframe. Hint: Na means no value, NaN means Not a Number. It is perfectly fine if there are no Na's in our dataset, but we should always check just in case.

In [39]:
```
print('detect missing data in "Apps" column:')
print(collegeDF.isnull(), '\n')
# replacing any missing data in Apps column with 0:
collegeDF['Apps'] = collegeDF['Apps'].fillna(0)
```

```
detect missing data in "Apps" column:
                                Private   Apps  Accept  Enroll  Top10perc  \
Names
Abilene Christian University      False  False   False   False      False
Adelphi University                False  False   False   False      False
Adrian College                    False  False   False   False      False
Agnes Scott College               False  False   False   False      False
Alaska Pacific University         False  False   False   False      False
...                                 ...    ...     ...     ...        ...
Worcester State College           False  False   False   False      False
Xavier University                 False  False   False   False      False
Xavier University of Louisiana    False  False   False   False      False
Yale University                   False  False   False   False      False
York College of Pennsylvania      False  False   False   False      False

                                Top25perc  F.Undergrad  P.Undergrad  Outstate  \
Names
Abilene Christian University        False        False        False     False
Adelphi University                  False        False        False     False
Adrian College                      False        False        False     False
Agnes Scott College                 False        False        False     False
Alaska Pacific University           False        False        False     False
...                                   ...          ...          ...       ...
Worcester State College             False        False        False     False
Xavier University                   False        False        False     False
Xavier University of Louisiana      False        False        False     False
Yale University                     False        False        False     False
York College of Pennsylvania        False        False        False     False

                                Room.Board  Books  Personal    PhD  Terminal  \
Names
Abilene Christian University         False  False     False  False     False
Adelphi University                   False  False     False  False     False
Adrian College                       False  False     False  False     False
Agnes Scott College                  False  False     False  False     False
Alaska Pacific University            False  False     False  False     False
...                                    ...    ...       ...    ...       ...
Worcester State College              False  False     False  False     False
Xavier University                    False  False     False  False     False
Xavier University of Louisiana       False  False     False  False     False
Yale University                      False  False     False  False     False
York College of Pennsylvania         False  False     False  False     False

                                S.F.Ratio  perc.alumni  Expend  Grad.Rate
Names
Abilene Christian University        False        False   False      False
Adelphi University                  False        False   False      False
Adrian College                      False        False   False      False
Agnes Scott College                 False        False   False      False
Alaska Pacific University           False        False   False      False
...                                   ...          ...     ...        ...
Worcester State College             False        False   False      False
Xavier University                   False        False   False      False
Xavier University of Louisiana      False        False   False      False
Yale University                     False        False   False      False
York College of Pennsylvania        False        False   False      False

[777 rows x 18 columns]
```

1. Find the college with the least out-of-state tuition and name this variable `college_least_tuition`. The variable should return the name of a college, not its tuition. **Answer on Coursera: Which college in this dataset has the least amount of out-of-state tuition?**

In [40]:
```
college_least_tuitionVal = collegeDF['Outstate'].min()
print('college_least_tuition, minimum value of column "Outstate": ',college_least_tuitionVal
, '\n')

x  = collegeDF.loc[collegeDF['Outstate'] == college_least_tuitionVal]
print(x)
print('type: ', type(x))
indx = x.index
print('indx: ', indx[0])
college_least_tuition = indx[0]
print(college_least_tuition)
```

```
college_least_tuition, minimum value of column "Outstate":  2340

                                  Private  Apps  Accept  Enroll  Top10perc  \
Names
Brigham Young University at Provo     Yes  7365    5402    4615         48

                                   Top25perc  F.Undergrad  P.Undergrad  \
Names
Brigham Young University at Provo         82        27378         1253

                                   Outstate  Room.Board  Books  Personal  PhD  \
Names
Brigham Young University at Provo      2340        3580    860      1220   76

                                   Terminal  S.F.Ratio  perc.alumni  Expend  \
Names
Brigham Young University at Provo        76       20.5           40    7916

                                   Grad.Rate
Names
Brigham Young University at Provo         33
type:  <class 'pandas.core.frame.DataFrame'>
indx:  Brigham Young University at Provo
Brigham Young University at Provo
```

In [41]:
```
grader.grade(test_case_id = 'college_least_tuition_test', answer = college_least_tuition)
```

```
Correct! You earned 3/3 points. You are a star!

Your submission has been successfully recorded in the gradebook.
```

1. From the original dataframe, select the 'PhD' column and name this dataframe `phd_column`. Find the length of this column and name this variable `phd_column_length`. **Enter the value of phd_column_length on Coursera.** Hint: Make sure to use double brackets to select a Pandas dataframe and single brackets to select a Pandas series. You will notice the difference in formatting: Pandas dataframes are neatly formatted, while Pandas series are not formatted. A Pandas dataframe gives extra functionality compared to a Pandas series, such as appending other dataframes and selecting multiple columns. A Pandas series is essentially a Numpy column.

In [42]:
```
# Double bracket for dataframe
phd_column = collegeDF[["PhD"]].copy()
phd_column
```

Out[42]:

Names	PhD
Abilene Christian University	70
Adelphi University	29
Adrian College	53
Agnes Scott College	92
Alaska Pacific University	76
...	...
Worcester State College	60
Xavier University	73
Xavier University of Louisiana	67
Yale University	96
York College of Pennsylvania	75

777 rows × 1 columns

In [44]:
```
grader.grade(test_case_id = 'phd_column_test', answer = phd_column)
```

```
Correct! You earned 3/3 points. You are a star!

Your submission has been successfully recorded in the gradebook.
```

In [43]:
```
phd_column_length = len(phd_column.index)
print(phd_column_length)
```

```
777
```

In [46]:
```
grader.grade(test_case_id = 'phd_column_length_test', answer = phd_column_length)
```

```
Correct! You earned 1/1 points. You are a star!

Your submission has been successfully recorded in the gradebook.
```

1. From the original dataframe, select both the 'Private' and 'Top10perc' columns, and slice them such that only the rows with index 15 and 16 remain. Name this dataframe `private_top10`. From this dataframe, find the length of the filtered 'Private' column and name this variable `private_column_length`. **Enter the value of private_column_length on Coursera.**

In [47]:
```
# Double bracket to obtain multiple columns
private_top10 =  collegeDF[['Private', 'Top10perc']].copy()
private_top10 =  private_top10[15:17 ]
```

In [48]:
```
grader.grade(test_case_id = 'private_top_10_test', answer = private_top10)
```

```
Correct! You earned 3/3 points. You are a star!

Your submission has been successfully recorded in the gradebook.
```

In [49]:
```
private_column_length = len(private_top10.index)
print(private_column_length)
```

```
2
```

In [50]:
```
grader.grade(test_case_id = 'private_column_length_test', answer = private_column_length)
```

```
Correct! You earned 2/2 points. You are a star!

Your submission has been successfully recorded in the gradebook.
```

1. From the original dataframe, select the row that only contains data about the "University of Pennsylvania". Note that many other colleges share the same name, but there is only one unique University of Pennsylvania. Name this dataframe `penn`.

In [51]:
```
ndxs = collegeDF.index
ndx = list(ndxs).index("University of Pennsylvania")
penn = collegeDF.iloc[[ndx]]
```

In [52]:
```
grader.grade(test_case_id = 'penn_test', answer = penn)
```

```
Correct! You earned 4/4 points. You are a star!

Your submission has been successfully recorded in the gradebook.
```

1. From the original dataframe, select the rows that contain all colleges with the name "Penn" included. The "P" should be capitalized. Name this dataframe `many_penns`. Comment on your observations. **Answer on Coursera: How many universities are there in this dataset that include "Penn"?**

In [53]:
```
subString = "Penn"
subList = [stng for stng in list(ndxs) if subString in stng]
many_penns = collegeDF.loc[subList]
```

In [54]:
```
grader.grade(test_case_id = 'many_penns_test', answer = many_penns)
```

```
Correct! You earned 4/4 points. You are a star!

Your submission has been successfully recorded in the gradebook.
```

Continue to explore the dataset by using any of the skills you learned in Recitation 1. Keep this notebook in a location that is easily accessible, as we will continue with it for the next assignment.