Homework 2: Pandas

Today we'll practice data exploration in pandas! Each of these cells should consist of a single line of pandas, answering the question.

First, you'll need to download the dataset "Top American Colleges 2022" from Kaggle.com and get it into this directory. You'll need to make an account first.

Below is a list of useful functions. Part of this homework is practicing reading the documentation, so you'll want to look them up as you go. I'd recommend starting with this: https://pandas.pydata.org/docs/user_guide/10min.html. Once you've read that, in general you can find the API for any of these functions by searching their name plus pandas.

List of helpful functions:

- read_csv
- head
- unique
- groupby
- apply (An important note about this one--pay careful attention to the weird axis argument. When you apply over a series, you often don't need it, but when you apply over a dataframe axis=1 and axis=0 will do very different things.)
- value_counts
- df.columns ('columns' is a dataframe variable that tracks the columns)
- isin
- fillna
- astype
- hist

```
In [1]: import numpy as np

In [2]: import pandas as pd
```

The Basics

First, read the dataframe in. Store it in a variable called "df".

```
df = pd.read_csv ('top_colleges_2022.csv')
```

Let's get a feel for our dataframe. Print out a list of columns

```
dfColumns = df.columns
print(dfColumns)
```

```
Index(['description', 'rank', 'organizationName', 'state', 'studentPopulation',
       'campusSetting', 'medianBaseSalary', 'longitude', 'latitude', 'website',
       'phoneNumber', 'city', 'country', 'state.1', 'region', 'yearFounded',
       'stateCode', 'collegeType', 'carnegieClassification',
       'studentFacultyRatio', 'totalStudentPop', 'undergradPop',
       'totalGrantAid', 'percentOfStudentsFinAid', 'percentOfStudentsGrant'],
      dtype='object')
```

Now print out the first ten elements. There's a single function that does it by default.

```
result = df.head(10)
print(result)
```

```
9  Private not-for-profit  Doctoral Universities: Very High Research Acti...

  studentFacultyRatio  totalStudentPop undergradPop totalGrantAid  \
0                   3            12195         4582    35299332.0
1                   4            20961         8464    51328461.0
2                  19            45878        33208    64495611.0
3                   4             8532         5516    44871096.0
4                   6            33882         8689    44615007.0
5                  18            46947        33641    61100980.0
6                   6             2307         2251    15204855.0
7                   4            14910         7199    48430359.0
8                   6            17855         7278    41087604.0
9                   6            30688        14202    59744979.0

  percentOfStudentsFinAid  percentOfStudentsGrant
0                    75.0                    60.0
1                    70.0                    55.0
2                    63.0                    53.0
3                    62.0                    61.0
4                    58.0                    54.0
5                    73.0                    67.0
6                    62.0                    52.0
7                    61.0                    53.0
8                    63.0                    47.0
9                    57.0                    47.0

[10 rows x 25 columns]
```

Exploration

Now let's learn to do some exploration. Try printing out the median "medianBaseSalary"

```
df['medianBaseSalary'].median()
```

```
112800.0
```

Making it a little more complicated--print out the median "medianBaseSalary" only for urban colleges.

```
df.loc[df['campusSetting'] == 'Urban', 'medianBaseSalary'].median()
```

```
113100.0
```

Now, still using one statement, let's print out median "medianBaseSalary" for all different possible values of "campusSetting". You'll need a statement we haven't used yet.

```
df.groupby("campusSetting")[["medianBaseSalary"]].median()
```

	medianBaseSala...
Rural	111450.0
Suburban	113500.0
Urban	113100.0

Print out the number of colleges by state. Your results should look something like:

NY 63
CA 55

etc.

```
df["state"].value_counts()
```

```
NY    63
CA    55
PA    33
MA    27
TX    26
IL    16
NJ    16
OH    15
MI    15
FL    14
VA    14
WA    13
MN    12
MD    12
IN    12
NC    11
TN     9
OR     9
GA     9
WI     8
MO     8
CT     8
CO     7
SC     6
AL     5
IA     5
DC     5
RI     5
AZ     4
NH     4
ME     4
VT     4
```

Display just the line for University of Maryland. (There are a couple of ways of doing this.)

```
df.loc[df['organizationName'] == 'University of Maryland, College Park']
```

	description object	rank int64	organizationName	state object	studentPopulation	campusSetting o...	medianBaseSala...	longitude float64
39	The University of Maryland, Colleg...	40	University of Maryland, Colleg...	MD	44404	Suburban	124500.0	-76.937269

Modifications

Let's start modifying our dataframe! Remember, dataframe operations return a copy by default, so you'll either need to use the inplace=True, or just assign the dataframe back into itself (as in, df = df.someFunction()).

Start by filling in all blank phone numbers with "no number"

```
dfPhone = df.phoneNumber

dfPhone.fillna('no number')
```

```
0        617-253-1000
1        650-723-2091
2      (510) 642-6000
3        609-258-3000
4        212-854-1754
            ...
493    (631) 687-5100
494      610-861-1320
495         no number
496         no number
497    (901) 678-2000
Name: phoneNumber, Length: 498, dtype: object
```

Take the website column and change it so that no string includes "http://" or "www"

```
dfWeb = df.website
dfweb2 = dfWeb.replace('www','', regex=True)
dfweb2.replace('http://','', regex=True)
```

```
0           web.mit.edu
1         .stanford.edu
2         .berkeley.edu
3        .princeton.edu
4         .columbia.edu
             ...
493          .sjcny.edu
494       .moravian.edu
495    https://.ltu.edu
496                 NaN
497         .mephis.edu
Name: website, Length: 498, dtype: object
```

Create a new column called "faculty" that computes the number of faculty at each university

```
df["faculty"] = df["studentPopulation"] / df["studentFacultyRatio"]
df["faculty"] = df["faculty"].astype(int)
print(df.faculty)
```

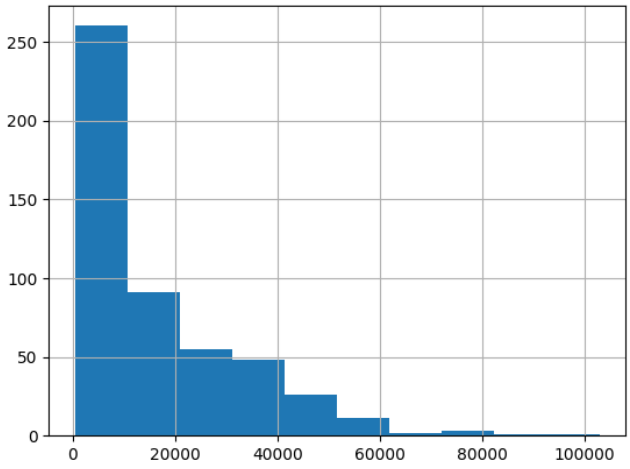
```
0      4065
1      5240
2      2414
3      2133
4      5647
       ...
493     491
494     269
495     287
496     165
497    1570
Name: faculty, Length: 498, dtype: int64
```

Graphs

Let's do some very basic graphing here! Create a histogram for the student population.

```
df["totalStudentPop"].hist()
```

```
<AxesSubplot: >
```


250
200
150
100
50
0
0
20000
40000
60000
80000
100000