Homework 1 Part 1: Pandas

Today we'll practice data exploration in pandas! Each of these cells should consist of *one or two lines of pandas**, answering the question.

First, you'll need to download the dataset "Top American Colleges 2022" (https://www.kaggle.com/datasets/kabhishm/top-american-colleges-2022) from Kaggle.com and get it into this directory. You'll need to make an account on kaggle 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

***Remember, all answers must be in ONE OR TWO LINES OF CODE. ***

The Basics

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

```
import pandas as pd

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

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

```
df.columns
```

```
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.

```
df.head(10)
```

	description	rank	organizationName	state	studentPopulation	campusSetting	medianBaseSalary	longitude	latitude	
0	A leading global research university, MIT attr...	1	Massachusetts Institute of Technology	MA	12195	Urban	173700.0	-71.093539	42.359006	http://web
1	Stanford University sits just outside of Palo ...	2	Stanford University	CA	20961	Suburban	173500.0	-122.168924	37.431370	http://www.stan
2	One of the top public universities in the coun...	2	University of California, Berkeley	CA	45878	Urban	154500.0	-122.258393	37.869236	http://www.berk
3	Princeton is a leading private research univer...	4	Princeton University	NJ	8532	Urban	167600.0	-74.659119	40.349855	http://www.princ
4	Located in upper Manhattan, Columbia Universit...	5	Columbia University	NY	33882	Urban	148800.0	-73.961288	40.806515	http://www.colun
5	The University of California, Los Angeles is t...	6	University of California, Los Angeles	CA	46947	Urban	137200.0	-118.437855	34.073903	http://
6	Located in rural Williamstown, MA, Williams Co...	7	Williams College	MA	2307	Rural	152600.0	-73.208078	42.712389	http://www.willi
7	Yale University is the second oldest Ivy Leagu...	8	Yale University	CT	14910	Urban	163700.0	-72.923425	41.314042	http://www.
8	Duke offers 53 undergraduate majors at its Dur...	9	Duke University	NC	17855	Urban	155000.0	-78.940277	36.001389	http://www.c
9	Founded by Benjamin Franklin, The University o...	10	University of Pennsylvania	PA	30688	Urban	164000.0	-75.162369	39.952270	http://www.up

10 rows × 25 columns

Exploration

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

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

```
112800.0
```

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

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

```
113100.0
```

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

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

```
campusSetting
Rural       111450.0
Suburban    113500.0
Urban       113100.0
Name: medianBaseSalary, dtype: float64
```

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

NY 63

CA 55

etc.

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

```
state
NY    63
CA    55
PA    33
MA    27
TX    26
NJ    16
IL    16
MI    15
OH    15
VA    14
FL    14
WA    13
IN    12
MN    12
MD    12
NC    11
GA     9
OR     9
TN     9
CT     8
MO     8
WI     8
CO     7
SC     6
AL     5
RI     5
DC     5
IA     5
UT     4
LA     4
AZ     4
VT     4
ME     4
NH     4
KY     4
OK     3
ID     3
NE     3
NM     3
MS     2
MT     2
ND     2
SD     2
HI     2
AR     2
NV     2
KS     2
PR     1
WV     1
WY     1
DE     1
Name: count, dtype: int64
```

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

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

	description	rank	organizationName	state	studentPopulation	campusSetting	medianBaseSalary	longitude	latitude	websi
39	The University of Maryland, College Park, is a...	40	University of Maryland, College Park	MD	44404	Suburban	124500.0	-76.937269	38.980725	http://www.umd.e

1 rows × 25 columns

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"

```
df['phoneNumber'].fillna(value='no number', inplace=True)

df['phoneNumber']
```

```
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://", "https://" or "www."

```
import re

df['website'] = df['website'].apply(func=lambda s: re.sub(r'(?:https?:\/\/)?(?:www\.)?', '', str(s)))

df['website']
```

```
0        web.mit.edu
1       stanford.edu
2       berkeley.edu
3      princeton.edu
4       columbia.edu
           ...
493        sjcny.edu
494     moravian.edu
495          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['totalStudentPop'] // df['studentFacultyRatio']

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.

```
import matplotlib.pyplot as plt

df['studentPopulation'].plot(kind='hist')

plt.xlabel('Student Population')
plt.ylabel('Frequency')
plt.title('Student Population Across Top 500 Colleges')

plt.show()
```