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

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

***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

```
list(df)
```

```
['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']
```

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	website	...	yearFounded	stateCode
0	A leading global research university, MIT attr...	1	Massachusetts Institute of Technology	MA	12195	Urban	173700.0	-71.093539	42.359006	http://web.mit.edu	...	1861.0	MA
1	Stanford University sits just outside of Palo ...	2	Stanford University	CA	20961	Suburban	173500.0	-122.168924	37.431370	http://www.stanford.edu	...	1891.0	CA
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.berkeley.edu	...	1868.0	CA
3	Princeton is a leading private research univer...	4	Princeton University	NJ	8532	Urban	167600.0	-74.659119	40.349855	http://www.princeton.edu	...	1746.0	NJ
4	Located in upper Manhattan, Columbia Universit...	5	Columbia University	NY	33882	Urban	148800.0	-73.961288	40.806515	http://www.columbia.edu	...	1754.0	NY
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://ucla.edu	...	1919.0	CA
6	Located in rural Williamstown, MA, Williams Co...	7	Williams College	MA	2307	Rural	152600.0	-73.208078	42.712389	http://www.williams.edu	...	1793.0	MA
7	Yale University is the second oldest Ivy Leagu...	8	Yale University	CT	14910	Urban	163700.0	-72.923425	41.314042	http://www.yale.edu	...	1701.0	CT
8	Duke offers 53 undergraduate majors at its Dur...	9	Duke University	NC	17855	Urban	155000.0	-78.940277	36.001389	http://www.duke.edu	...	1924.0	NC
9	Founded by Benjamin Franklin, The University o...	10	University of Pennsylvania	PA	30688	Urban	164000.0	-75.162369	39.952270	http://www.upenn.edu	...	1740.0	PA

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	medianBaseSalary
Rural	111450.0
Suburban	113500.0
Urban	113100.0

dtype: float64

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

NY 63
CA 55

etc.

```
df.groupby('state')['state'].count()
```

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

dtype: int64

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

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

	description	rank	organizationName	state	studentPopulation	campusSetting	medianBaseSalary	longitude	latitude	website	...	yearFounded	stateCode	c
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.edu	...	1858.0	MD	

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 = df.phoneNumber.fillna('no number')
df.phoneNumber
```

	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

498 rows × 1 columns

dtype: object

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

```
df.website = df.website.str.replace("(http://|https://|www.)", "",regex = True)
df.website
```

	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

498 rows × 1 columns

dtype: object

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

```
import math
df["faculty"] = (df.totalStudentPop / df.studentFacultyRatio).apply(math.ceil)
df.faculty
```

	faculty
0	4065
1	5241
2	2415
3	2133
4	5647
...	...
493	492
494	270
495	288
496	165
497	1571

498 rows × 1 columns

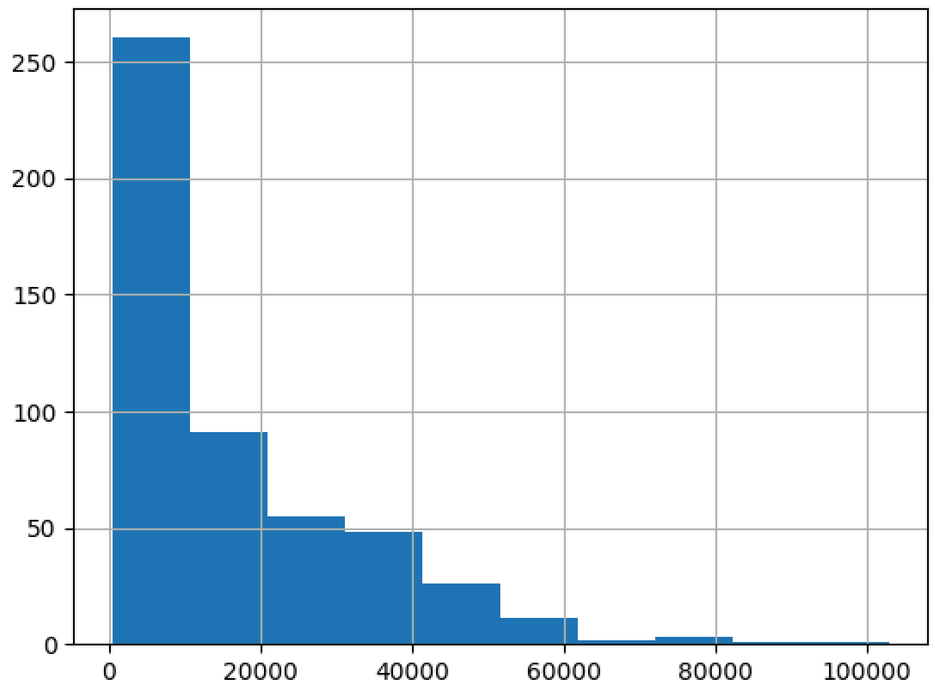
dtype: int64

Graphs

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

```
df.totalStudentPop.hist()
```

```
<Axes: >
```



Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit