Homework 3: Table Manipulation and Visualization

Reading:

- Visualization

Please complete this notebook by filling in the cells provided. Before you begin, execute the following cell to load the provided tests. Each time you start your server, you will need to execute this cell again to load the tests.

Throughout this homework and all future ones, please be sure to not re-assign variables throughout the notebook! For example, if you use `max_temperature` in your answer to one question, do not reassign it later on. Moreover, please be sure to only put your written answers in the provided cells.

```
# Don't change this cell; just run it.

import numpy as np
from datascience import *


# These lines do some fancy plotting magic.\n",
import matplotlib
%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')

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

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

1. Unemployment

The Federal Reserve Bank of St. Louis publishes data about jobs in the US. Below, we've loaded data on unemployment in the United States. There are many ways of defining unemployment, and our dataset includes two notions of the unemployment rate:

1. Among people who are able to work and are looking for a full-time job, the percentage who can't find a job. This is called the Non-Employment Index, or NEI.
2. Among people who are able to work and are looking for a full-time job, the percentage who can't find any job *or* are only working at a part-time job. The latter group is called "Part-Time for Economic Reasons", so the acronym for this index is NEI-PTER. (Economists are great at marketing.)

The source of the data is here.

Question 1. The data are in a CSV file called `unemployment.csv`. Load that file into a table called `unemployment`.

Note: You will need to import your google drive, and then read from your google drive. You can look at previous labs/hws to copy and adjust the code. The file unemployment.csv should already be shared with you on drive.

```
unemployment = Table.read_table('/content/drive/MyDrive/unemployment.csv')
unemployment
```

Date	NEI	NEI-PTER
1994-01-01	10.0974	11.172
1994-04-01	9.6239	10.7883
1994-07-01	9.3276	10.4831
1994-10-01	9.1071	10.2361
1995-01-01	8.9693	10.1832
1995-04-01	9.0314	10.1071
1995-07-01	8.9802	10.1084
1995-10-01	8.9932	10.1046
1996-01-01	9.0002	10.0531
1996-04-01	8.9038	9.9782

... (80 rows omitted)

Question 2. Sort the data in descending order by NEI, naming the sorted table `by_nei`. Create another table called `by_nei_pter` that's sorted in descending order by NEI-PTER instead.

```
by_nei = unemployment.sort(1)
by_nei_pter = unemployment.sort(2, descending=True)
```

Question 3. Use `take` to make a table containing the data for the 10 quarters when NEI was greatest. Call that table `greatest_nei`.

`greatest_nei` should be sorted in descending order of `NEI`. Note that each row of `unemployment` represents a quarter.

```
greatest_nei = by_nei.take(np.arange(79,89)).sort(1, descending=True)
greatest_nei
```

Date	NEI	NEI-PTER
2010-01-01	10.9054	12.7311
2009-07-01	10.8089	12.7404
2009-04-01	10.7082	12.5497
2010-04-01	10.6597	12.5664
2010-10-01	10.5856	12.4329
2010-07-01	10.5521	12.3897
2011-01-01	10.5024	12.3017
2011-07-01	10.4856	12.2507
2011-04-01	10.4409	12.247
2011-10-01	10.3287	12.1214

Question 4. It's believed that many people became PTER (recall: "Part-Time for Economic Reasons") in the "Great Recession" of 2008-2009. NEI-PTER is the percentage of people who are unemployed (and counted in the NEI) plus the percentage of people who are PTER. Compute an array containing the percentage of people who were PTER in each quarter. (The first element of the array should correspond to the first row of `unemployment`, and so on.)

Note: Use the original `unemployment` table for this.

```
pter_column = unemployment.column(2) - unemployment.column(1)
pter = unemployment.with_column("PTER", pter_column)
pter
```

Date	NEI	NEI-PTER	PTER
1994-01-01	10.0974	11.172	1.0746
1994-04-01	9.6239	10.7883	1.1644
1994-07-01	9.3276	10.4831	1.1555
1994-10-01	9.1071	10.2361	1.129
1995-01-01	8.9693	10.1832	1.2139
1995-04-01	9.0314	10.1071	1.0757
1995-07-01	8.9802	10.1084	1.1282
1995-10-01	8.9932	10.1046	1.1114
1996-01-01	9.0002	10.0531	1.0529
1996-04-01	8.9038	9.9782	1.0744

... (80 rows omitted)

Question 5. Add `pter` as a column to `unemployment` (named "PTER") and sort the resulting table by that column in descending order. Call the table `by_pter`.

Try to do this with a single line of code, if you can.

```
by_pter = unemployment.with_column("PTER", pter_column).sort(3, descending=True)
by_pter
```

Date	NEI	NEI-PTER	PTER
2009-07-01	10.8089	12.7404	1.9315
2010-04-01	10.6597	12.5664	1.9067
2009-10-01	10.9698	12.8557	1.8859
2010-10-01	10.5856	12.4329	1.8473
2009-04-01	10.7082	12.5497	1.8415
2010-07-01	10.5521	12.3897	1.8376
2010-01-01	10.9054	12.7311	1.8257
2011-04-01	10.4409	12.247	1.8061
2011-01-01	10.5024	12.3017	1.7993
2011-10-01	10.3287	12.1214	1.7927

... (80 rows omitted)

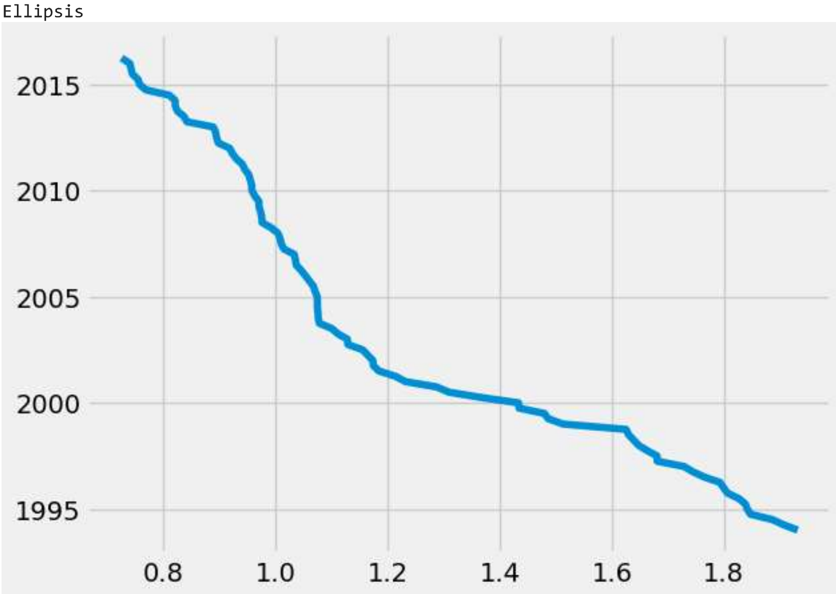
Question 6.

Create a line plot of the PTER over time.

To do this, create a new table called `pter_over_time` that adds the `year` array and the `pter` array to the `unemployment` table. Label these columns `Year` and `PTER`. Then, generate a line plot using one of the table methods you've learned in class.

```
year = 1994 + np.arange(by_pter.num_rows)/4
pter_over_time = by_pter.with_column("year", year)
plots.plot(pter_over_time.column(3),pter_over_time.column(4))
...
```

```
Ellipsis
```



Question 7. Were PTER rates high during the Great Recession (that is to say, were PTER rates particularly high in the years 2008 through 2011)? Assign highPTER to `True` if you think PTER rates were high in this period, and `False` if you think they weren't.

```
highPTER = True
```

2. Birth Rates

The following table gives census-based population estimates for each state on both July 1, 2015 and July 1, 2016. The last four columns describe the components of the estimated change in population during this time interval. **For all questions below, assume that the word "states" refers to all 52 rows including Puerto Rico & the District of Columbia.**

The data was taken from here.

If you want to read more about the different column descriptions, click here!

The raw data is a bit messy - run the cell below to clean the table and make it easier to work with.

```
#You may need to change the file path below.
pop = Table.read_table('/content/drive/MyDrive/nst-est2016-alldata.csv').where('SUMLEV', 40).select([1, 4, 12, 13, 27, 34, 62, 69])
pop = pop.relabeled('POPESTIMATE2015', '2015').relabeled('POPESTIMATE2016', '2016')
pop = pop.relabeled('BIRTHS2016', 'BIRTHS').relabeled('DEATHS2016', 'DEATHS')
pop = pop.relabeled('NETMIG2016', 'MIGRATION').relabeled('RESIDUAL2016', 'OTHER')
pop = pop.with_columns("REGION", np.array([int(region) if region != "X" else 0 for region in pop.column("REGION")]))
pop.set_format([2, 3, 4, 5, 6, 7], NumberFormatter(decimals=0)).show(5)
```

REGION	NAME	2015	2016	BIRTHS	DEATHS	MIGRATION	OTHER
3	Alabama	4,853,875	4,863,300	58,556	52,405	3,874	-600
4	Alaska	737,709	741,894	11,255	4,511	-2,557	-2
4	Arizona	6,817,565	6,931,071	87,204	56,564	76,405	6,461
3	Arkansas	2,977,853	2,988,248	37,936	30,581	3,530	-490
4	California	38,993,940	39,250,017	502,848	273,850	33,530	-6,451

... (47 rows omitted)

Question 1. Assign `us_birth_rate` to the total US annual birth rate during this time interval. The annual birth rate for a year-long period is the total number of births in that period as a proportion of the population size at the start of the time period.

Hint: Which year corresponds to the start of the time period?

```
us_birth_rate = sum(pop.column(4)/pop.column(2))/52*100
us_birth_rate
```

```
1.232821678238599
```

Question 2. Assign `movers` to the number of states for which the **absolute value** of the **annual rate of migration** was higher than 1%. The annual rate of migration for a year-long period is the net number of migrations (in and out) as a proportion of the population size at the start of the period. The `MIGRATION` column contains estimated annual net migration counts by state.

```
migration_rates = pop.column(6)/pop.column(2)*100
movers = [False] * 51
for x in range(51):
  if(migration_rates[x]>1):
    movers[x]=True
movers
```

```
[False,
 False,
 True,
 False,
 False,
 True,
 False,
 False,
 False,
 True,
 False,
 False,
 True,
 False,
 False,
 False,
 False,
 False,
 False,
 False,
 False,
 False,
 False,
 False,
 False,
 False,
 False,
 False,
 True,
 False,
 False,
 False,
 False,
 False,
 False,
 False,
 False,
 True,
 False,
 False,
 True,
 False,
 False,
 False,
 False,
 False,
 False,
 True,
 False,
 False,
 False]
```

Question 3. Assign `west_births` to the total number of births that occurred in region 4 (the Western US).

Hint: Make sure you double check the type of the values in the region column, and appropriately filter (i.e. the types must match!).

```
west_births = sum(pop.where("REGION"==4).column(4))
west_births
```

```
3977745
```

Question 4. Assign `less_than_west_births` to the number of states that had a total population in 2016 that was smaller than the *total number of births in region 4 (the Western US)* during this time interval.

```
less_than_west_births = len(pop.where("2016", are.above(3977745)))
less_than_west_births
```

    8

Question 5.

In the next question, you will be creating a visualization to understand the relationship between birth and death rates by looking at birth rates vs death rates for the different states. The annual death rate for a year-long period is the total number of deaths in that period as a proportion of the population size at the start of the time period.

What visualization is most appropriate to see if there is an association between birth and death rates during a given time interval?

1. Line Graph
2. Scatter Plot
3. Bar Chart

Assign `visualization` below to the number corresponding to the correct visualization.

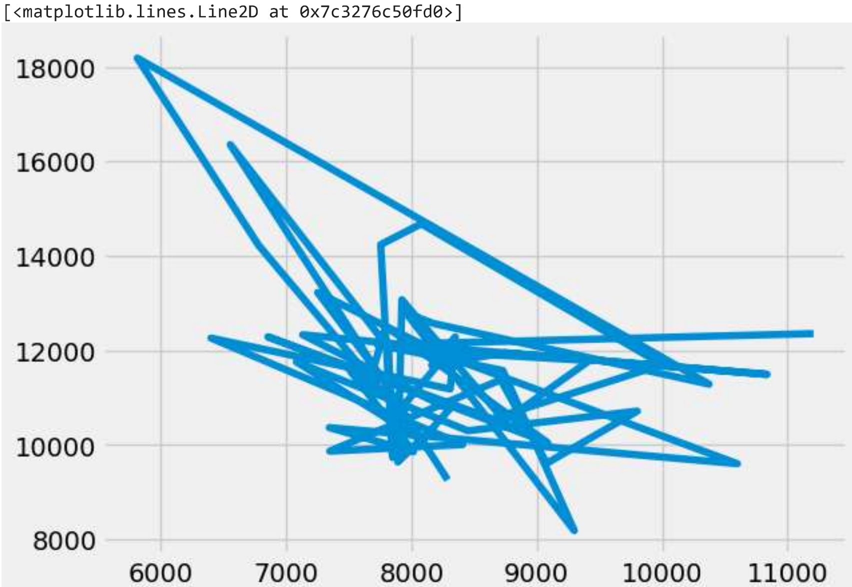
```
visualization = 1
```

Question 6.

In the code cell below, create a visualization that will help us determine if there is an association between birth rate and death rate during this time interval. It may be helpful to create an intermediate table here.

```
# Generate your chart in this cell
birth_rates = pop.column(2) / pop.column(4)*100
death_rates = pop.column(2) / pop.column(5)*100
plots.plot(birth_rates, death_rates)
```

    [<matplotlib.lines.Line2D at 0x7c3276c50fd0>]



Question 7. `True` or `False`: There is an association between birth rate and death rate during this time interval.

Assign `assoc` to `True` or `False` in the cell below.

```
assoc = False
```

3. Uber

Below we load tables containing 200,000 weekday Uber rides in the Manila, Philippines, and Boston, Massachusetts metropolitan areas from the [Uber Movement](https://movement.uber.com) project. The `sourceid` and `dstid` columns contain codes corresponding to start and end locations of each ride. The `hod` column contains codes corresponding to the hour of the day the ride took place. The `ride time` column contains the length of the ride, in minutes.

```
boston = Table.read_table("/content/drive/MyDrive/boston.csv")
manila = Table.read_table("/content/drive/MyDrive/manila.csv")
print("Boston Table")
boston.show(4)
print("Manila Table")
manila.show(4)
```

Boston Table

sourceid	dstid	hod	ride time
584	33	7	11.866
1013	1116	13	17.7993
884	1190	22	19.3488
211	364	1	1.7235

... (199996 rows omitted)

Manila Table

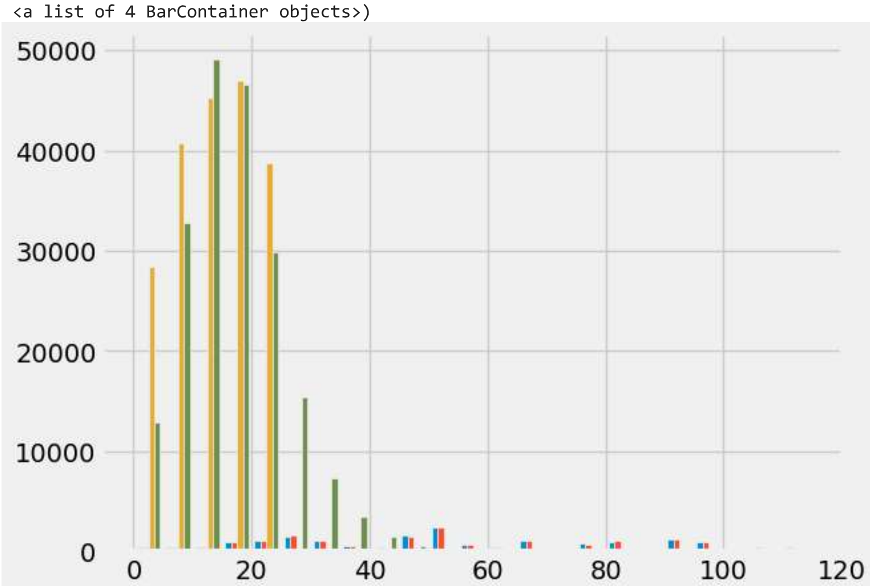
sourceid	dstid	hod	ride time
544	5	22	22.8115
302	240	21	7.02267
278	99	16	21.6437
720	775	18	13.0597

... (199996 rows omitted)

Question 1. Produce histograms of all ride times in Boston using the given bins.

```
equal_bins = np.arange(0, 120, 5)
plots.hist(boston, equal_bins)
```

```
          4.14000000e+02,   1.06500000e+03,   1.95000000e+02,
          7.78000000e+02,   9.96000000e+02,   1.34000000e+02,
          1.18300000e+03,   9.70000000e+02,   4.10000000e+01,
          4.04000000e+02,   3.56000000e+02],
       [  3.92000000e+02,   3.54000000e+02,   3.97000000e+02,
          9.16000000e+02,   1.06500000e+03,   1.59800000e+03,
          1.07700000e+03,   5.50000000e+02,   3.67000000e+02,
          1.54100000e+03,   2.38000000e+03,   6.56000000e+02,
          4.49000000e+02,   1.13300000e+03,   2.12000000e+02,
          7.42000000e+02,   1.01900000e+03,   1.79000000e+02,
          1.21400000e+03,   1.01200000e+03,   4.10000000e+01,
          3.37000000e+02,   3.03000000e+02],
       [  2.83900000e+04,   4.07090000e+04,   4.51940000e+04,
          4.69210000e+04,   3.87860000e+04,   0.00000000e+00,
          0.00000000e+00,   0.00000000e+00,   0.00000000e+00,
          0.00000000e+00,   0.00000000e+00,   0.00000000e+00,
          0.00000000e+00,   0.00000000e+00,   0.00000000e+00,
          0.00000000e+00,   0.00000000e+00,   0.00000000e+00,
          0.00000000e+00,   0.00000000e+00,   0.00000000e+00,
          0.00000000e+00,   0.00000000e+00],
       [  1.29400000e+04,   3.27180000e+04,   4.91110000e+04,
          4.65790000e+04,   2.98980000e+04,   1.54260000e+04,
          7.30200000e+03,   3.42400000e+03,   1.47400000e+03,
          6.13000000e+02,   2.68000000e+02,   1.14000000e+02,
          5.40000000e+01,   2.20000000e+01,   1.10000000e+01,
          5.00000000e+00,   4.00000000e+00,   0.00000000e+00,
          1.00000000e+00,   0.00000000e+00,   0.00000000e+00,
          1.00000000e+00,   1.00000000e+00]]),
array([   0.,    5.,   10.,   15.,   20.,   25.,   30.,   35.,   40.,
         45.,   50.,   55.,   60.,   65.,   70.,   75.,   80.,   85.,
         90.,   95.,  100.,  105.,  110.,  115.]),
<a list of 4 BarContainer objects>)
```



Question 2. Now, produce histograms of all ride times in Manila using the given bins.

```
plots.hist(manila, equal_bins)

# Don't delete the following line!
plots.ylim(0, 0.05)
```

```
(0.0, 0.05)
```