Homework 3: Table Manipulation and Visualization

Reading:

- [Visualization (https://www.inferentialthinking.com/chapters/07/visualization.html)](https://www.inferentialthinking.com/chapters/07/visualization.html)

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.

Homework 3 is due Thursday, 2/6 at 11:59pm. Start early so that you can come to office hours if you're stuck. Check the website for the office hours schedule. Late work will not be accepted as per the policies of this course.

In [ ]:

```
# 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')
```

1. Differences between Universities

Question 1.1. Suppose you're choosing a university to attend, and you'd like to *quantify* how *dissimilar* any two universities are. You rate each university you're considering on several numerical traits. You decide on a very detailed list of 1000 traits, and you measure all of them! Some examples:

- The cost to attend (per year).
- The average Yelp review of nearby Thai restaurants.
- The USA Today ranking of the Medical school.
- The USA Today ranking of the Engineering school.

You decide that the dissimilarity between two universities is the *total* of the differences in their traits. That is, the dissimilarity is:

- the **sum** of
- the absolute values of
- the 1000 differences in their trait values.

In the next cell, we've loaded arrays containing the 1000 trait values for Stanford and Berkeley. Compute the dissimilarity (according to the above technique) between Stanford and Berkeley. Call your answer `dissimilarity` . Use a single line of code to compute the answer.

Note: The data we're using aren't real -- we made them up for this exercise, except for the cost-of-attendance numbers, which were found online.

```
In [ ]: stanford = Table.read_table("stanford.csv").column("Trait value")
        berkeley = Table.read_table("berkeley.csv").column("Trait value")

        dissimilarity = ...
        dissimilarity
```

Question 1.2. Why do we sum up the absolute values of the differences in trait values, rather than just summing up the differences?

Write your answer here, replacing this text.

Weighing the traits

After computing dissimilarities between several schools, you notice a problem with your method: the scale of the traits matters a lot.

Since schools cost tens of thousands of dollars to attend, the cost-to-attend trait is always a much bigger *number* than most other traits. That makes it affect the dissimilarity a lot more than other traits. Two schools that differ in cost-to-attend by $\$900$, but are otherwise identical, get a dissimilarity of $900$. But two schools that differ in graduation rate by 0.9 (a huge difference!), but are otherwise identical, get a dissimilarity of only 0.9.

One way to fix this problem is to assign different "weights" to different traits. For example, we could fix the problem above by multiplying the difference in the cost-to-attend traits by $.001$, so that a difference of $\$900$ in the attendance cost results in a dissimilarity of $\$900 \times .001$, or 0.9.

Here's a revised method that does that for every trait:

1. For each trait, subtract the two schools' trait values.
2. Then take the absolute value of that difference.
3. Now multiply that absolute value by a trait-specific number, like $.001$ or 2.
4. Now, sum the 1000 resulting numbers.

Question 1.3. Suppose you've already decided on a weight for each trait. These are loaded into an array called `weights` in the cell below. `weights.item(0)` is the weight for the first trait, `weights.item(1)` is the weight for the second trait, and so on. Use the revised method to compute a revised dissimilarity between Berkeley and Stanford.

Hint: Using array arithmetic, your answer should be almost as short as in question 1.

In [ ]:

```
weights = Table.read_table("weights.csv").column("Weight")

revised_dissimilarity = ...
revised_dissimilarity
```

2. 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 (https://fred.stlouisfed.org/categories/33509)](https://fred.stlouisfed.org/categories/33509).

Question 2.1. The data are in a CSV file called `unemployment.csv` . Load that file into a table called `unemployment` .

```
In [ ]: unemployment = ...
        unemployment
```

Question 2.2. 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.

```
In [ ]: by_nei = ...
        by_nei_pter = ...
```

Question 2.3. Use `take` to make a table containing the data for the 10 quarters when NEI was greatest. Call that table `greatest_nei` .

```
In [ ]: greatest_nei = ...
        greatest_nei
```

Question 2.4. 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.

In [ ]:

```
pter = ...
pter
```

Question 2.5. 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.

In [ ]:

```
by_pter = ...
by_pter
```

Question 2.6. Create a line plot of the PTER over time. To do this, first add the `year` array and the `pter` array to the `unemployment` table; label these columns "Year" and "PTER", respectively. Then, generate a line plot using one of the table methods you've learned in class. Assign this new table to `pter_over_time`.

In [ ]:

```
year = 1994 + np.arange(by_pter.num_rows)/4
pter_over_time = ...
...
```

Question 2.7. Were PTER rates high during or directly after 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.

In [ ]:

```
highPTER = ...
```

3. 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 (http://www2.census.gov/programs-surveys/popest/datasets/2010-2016/national/totals/nst-est2016-alldata.csv)](http://www2.census.gov/programs-surveys/popest/datasets/2010-2016/national/totals/nst-est2016-alldata.csv).

If you want to read more about the different column descriptions, go [here (http://www2.census.gov/programs-surveys/popest/datasets/2010-2015/national/totals/nst-est2015-alldata.pdf)](http://www2.census.gov/programs-surveys/popest/datasets/2010-2015/national/totals/nst-est2015-alldata.pdf)! As of February 2017, no descriptions were posted for 2010 - 2016.

In [ ]:

```
# Don't change this cell; just run it.
pop = Table.read_table('nst-est2016-alldata.csv').where('SUMLEV', 40).
select([1, 4, 12, 13, 27, 34, 62, 69])
pop = pop.relabeled('POPESTIMATE2015', '2015').relabeled('POPESTIMATE2
016', '2016')
pop = pop.relabeled('BIRTHS2016', 'BIRTHS').relabeled('DEATHS2016', 'D
EATHS')
pop = pop.relabeled('NETMIG2016', 'MIGRATION').relabeled('RESIDUAL2016
', 'OTHER')
pop.set_format([2, 3, 4, 5, 6, 7], NumberFormatter(decimals=0)).show(5
)
```

Question 3.1. 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: What year corresponds to the start of the time period?

In [ ]:

```
us_birth_rate = ...
us_birth_rate
```

Question 3.2. Assign `fastest_growth` to an array of the names of the five states with the fastest population growth rates in *descending order of growth rate*. We have first created a new version of the `pop` table, called `growth_rates` , which includes a column with the growth rate of each state. Making intermediate tables can improve the readability of the code and make it easier to follow when revisting at a later time.

In [ ]:

```
growth_rates = pop.with_column('Growth Rate', (pop.column(3) / pop.col
umn(2)) - 1)
fastest_growth = ...
fastest_growth
```

Question 3.3. 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.

In [ ]:

```
migration_rates = ...
movers = ...
movers
```

Question 3.4. 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.

In [ ]:

```
west_births =
west_births
```

Question 3.5. 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.

In [ ]:

```
less_than_west_births = ...
less_than_west_births
```

Question 3.6.

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.

In [ ]:

```
# Generate your chart in this cell
...
```

Question 3.7. `True` or `False` : There appears to be a negative association between birth rate and death rate during this time interval.

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

```
In [ ]: assoc = ...
```

4. Marginal Histograms

Consider the following scatter plot:

The axes of the plot represent values of two variables: x and y.

Suppose we have a table called `t` that has two columns in it:

- `x` : a column containing the x-values of the points in the scatter plot
- `y` : a column containing the y-values of the points in the scatter plot

Question 4.1: Match each of the following histograms to the code that produced them. Explain your reasoning.

Histogram A:

Histogram B:

Line 1: `t.hist('x')`

Histogram for Line 1: ...

Explanation:...

Line 2: `t.hist('y')`

Histogram for Line 2:...

Explanation:...

5. Submission

Once you're finished, submit your assignment as a .ipynb (Jupyter Notebook) and .pdf (download as .html, then print to save as a .pdf) on the class Canvas site.

In [ ]: