Project 1 - California Water Usage

Welcome to the first project in Data 8! We will be exploring possible connections between water usage, geography, and income in California. The water data for this project was procured from the California State Water Resources Control Board (http://www2.pacinst.org/gpcd/table.html) and curated by the Pacific Institute (http://pacinst.org/). The map data includes US topography (https://github.com/jgoodall/us-maps), California counties (https://github.com/johan/world.geo.json/tree/master/countries/USA/CA), and ZIP codes (http://bl.ocks.org/jefffriesen/6892860).

The dataset on income comes from the IRS (documented here (http://www.irs.gov/pub/irs-soi/13zpdoc.doc)). We have identified some interesting columns in the dataset, but a full description of all the columns (and a definition of the population in the dataset and some interesting anonymization procedures they used) is available here (irs_info.pdf).

Administrivia

Piazza

While collaboration is encouraged on this and other assignments, sharing answers is never okay. In particular, posting code or other assignment answers publicly on Piazza (or elsewhere) is academic dishonesty. It will result in a reduced project grade at a minimum. If you wish to ask a question and include code, you *must* make it a private post.

Partners

You may complete the project with up to one partner. Partnerships are an exception to the rule against sharing answers. If you have a partner, one person in the partnership should submit your project on Gradescope and include the other partner in the submission. (Gradescope will prompt you to fill this in.)

Your partner *must be in your lab section*. You can ask your TA to pair you with someone from your lab if you're unable to find a partner. (That will happen in lab the week the project comes out.)

Due Date and Checkpoint

Part of the project will be due early. Parts 1 and 2 of the project (out of 3) are due **Tuesday, September 27th at 5PM**. Unlike the final submission, this early checkpoint will be graded for completion. It will be worth approximately 10% of the total project grade. Simply submit your partially-completed notebook as a PDF, as you would submit any other notebook. (See the note above on submitting with a partner.)

The entire project (parts 1, 2, and 3) will be due **Tuesday, October 4 at 5PM**. (Again, see the note above on submitting with a partner.)

On to the project!

As usual, **run the cell below** to prepare the automatic tests. **Passing the automatic tests does not guarantee full credit on any question.** The tests are provided to help catch some common errors, but it is *your* responsibility to answer the questions correctly.

```
In [ ]: # Run this cell, but please don't change it.

        import numpy as np
        import math
        from datascience import *

        # These lines set up the plotting functionality and formatting.
        import matplotlib
        matplotlib.use('Agg', warn=False)
        %matplotlib inline
        import matplotlib.pyplot as plots
        plots.style.use('fivethirtyeight')

        # These lines load the tests.
        from client.api.assignment import load_assignment
        tests = load_assignment('project1.ok')
```

First, load the data. Loading may take some time.

```
In [ ]: # Run this cell, but please don't change it.

        districts = Map.read_geojson('water_districts.geojson')
        zips = Map.read_geojson('ca_zips.geojson.gz')
        usage_raw = Table.read_table('water_usage.csv', dtype={'pwsid': str})
        income_raw = Table.read_table('ca_income_by_zip.csv', dtype={'ZIP':
        str}).drop('STATEFIPS', 'STATE', 'agi_stub')
        wd_vs_zip = Table.read_table('wd_vs_zip.csv', dtype={'PWSID': str,
        'ZIP': str}).set_format(make_array(2, 3), PercentFormatter)
```

Part 1: Maps

The `districts` and `zips` data sets are `Map` objects. Documentation on mapping in the `datascience` package can be found at data8.org/datascience/maps.html (http://data8.org/datascience/maps.html). To view a map of California's water districts, run the cell below. Click on a district to see its description.

```
In [ ]: districts.format(width=400, height=200)
```

A `Map` is a collection of regions and other features such as points and markers, each of which has a **string** `id` and various properties. You can view the features of the `districts` map as a table using `Table.from_records`.

```
In [ ]: district_table = Table.from_records(districts.features)
        district_table.show(3)
```

To display a `Map` containing only two features from the `district_table`, call `Map` on an array containing those two features from the `feature` column.

Question 1.1. Draw a map of the Alameda County Water District (row 0) and the East Bay Municipal Utilities District (row 2).

```
In [ ]: # Fill in the next line so the last line draws a map of those two distri
        cts.
        alameda_and_east_bay = ...
        Map(alameda_and_east_bay, height=300, width=300)
```

```
In [ ]: _ = tests.grade('q11')
```

Hint: If scrolling becomes slow on your computer, you can clear maps for the cells above by running `Cell > All Output > Clear` from the `Cell` menu.

Part 2: California Income

Let's look at the `income_raw` table, which comes from the IRS. We're going to link this information about incomes to our information about water. First, we need to investigate the income data and get it into a more usable form.

```
In [ ]: income_raw
```

Some observations:

1. The table contains several numerical columns and a column for the ZIP code.
2. For each ZIP code, there are 6 rows. Each row for a ZIP code has data from tax returns in one *income bracket*. (A tax return is the tax filing from one person or household. An income bracket is a group of people whose annual income is in some range, like $25,000 to $34,999.)
3. According to the IRS documentation, all the numerical columns are *totals* -- either total numbers of returns that fall into various categories, or total amounts of money (in thousands of dollars) from returns in those categories. For example, the column `'N02650'` is the number of returns that included a total income amount, and `'A02650'` is the total amount of total income (in thousands of dollars) from those returns.

For the analysis we're about to do, we won't need to use the information about tax brackets. We will need to know the total income, total number of returns, and other totals from each ZIP code.

Question 2.1. Assign the name `income_by_zipcode` to a table with just one row per ZIP code. When you group according to ZIP code, the remaining columns should be summed. In other words, for any other column such as `'N02650'`, the value of `'N02650'` in a row corresponding to ZIP code 90210 (for example) should be the sum of the values of `'N02650'` in the 6 rows of `income_raw` corresponding to ZIP code 90210.

```
In [ ]: income_by_zipcode = ...
        income_by_zipcode
```

```
In [ ]: _ = tests.grade('q21')
```

Your `income_by_zipcode` table probably has column names like `N1 sum`, which looks a little weird.

Question 2.2. Relabel the columns in `income_by_zipcode` to match the labels in `income_raw`

Hint: Inspect `income_raw.labels` and `income_by_zipcode.labels` to find the differences you need to change.

Hint 2: Since there are many columns, it will be easier to relabel each of them by using a `for` statement. See Chapter 8 of the textbook for details.

Hint 3: You can use the `replace` method of a string to remove excess content. See lab02 (../../lab/lab02/) for examples.

Hint 4: To create a new table from an existing table with one label replaced, use `relabeled`. To **change** a label in an existing table permanently, use `relabel`. Both methods take two arguments: the old label and the new label. You can solve this problem with either one, but `relabel` is simpler.

```
In [ ]: ...
            ...
        income_by_zipcode
```

```
In [ ]: _ = tests.grade('q22')
```

Question 2.3. Create a table called `income` with one row per ZIP code and the following columns.

1. A `ZIP` column with the same contents as `'ZIP'` from `income_by_zipcode`.
2. A `num returns` column containing the total number of tax returns that include a total income amount (column `'N02650'` from `income_by_zipcode`).
3. A `total income ($)` column containing the total income in all tax returns in thousands of dollars (column `'A02650'` from `income_by_zipcode`).
4. A `num farmers` column containing the number of farmer returns (column `'SCHF'` from `income_by_zipcode`).

```
In [ ]: income = Table().with_columns(
                ...
                ...
                ...
                ...
            )
        income.set_format('total income ($)', NumberFormatter(0)).show(5)
```

```
In [ ]: _ = tests.grade('q23')
```

Question 2.4. All ZIP codes with less than 100 returns (or some other special conditions) are grouped together into one ZIP code with a special code. Remove the row for that ZIP code from the `income` table.

Hint 1: This ZIP code value has far more returns than any of the other ZIP codes. Try using `group` and `sort` to find it.

Hint 2: To **remove** a row in the `income` table using `where`, assign `income` to the smaller table using the following expression structure:

```
income = income.where(...)
```

Hint 3: Each ZIP code is represented as a string, not an `int`.

```
In [ ]: income = ...
```

```
In [ ]: _ = tests.grade('q24')
```

Because each ZIP code has a different number of people, computing the average income across several ZIP codes requires some care. This will come up several times in this project. Here is a simple example:

Question 2.5 Among all the tax returns that

1. include a total income amount, and
2. are filed by people living in either ZIP code 94576 (a rural area north of Napa) or in ZIP code 94704 (a moderately-dense area in South Berkeley),

what is the average total income? **Express the answer in *dollars* as an `int` rounded to the nearest dollar.**

```
In [ ]: # Our solution took several lines of code.
        average_income = ...
        average_income
```

```
In [ ]: _ = tests.grade('q25')
```

Question 2.6. Among all California tax returns that include a total income amount, what is the average total income? **Express the answer in *dollars* as an `int` rounded to the nearest dollar.**

```
In [ ]: avg_total = ...
        avg_total
```

Farming

Farms use water, so it's plausible that farming is an important factor in water usage. Here, we will check for a relationship between farming and income.

Among the tax returns in California for ZIP codes represented in the `incomes` table, is there an association between income and living in a ZIP code with a lot of farmers?

We'll try to answer the question in 3 ways.

Question 2.7. Make a scatter plot with one point for each ZIP code. Display the average income *in dollars* on the vertical axis and the proportion of returns that are from farmers on the horizontal axis.

```
In [ ]: # Write code to make a scatter plot here.
        ...
```

Question 2.8. From the graph, can you say whether ZIP codes with more farmers typically have lower or higher average income than ZIP codes with few or no farmers? Can you say how much lower or higher?

Write your answer here, replacing this text.

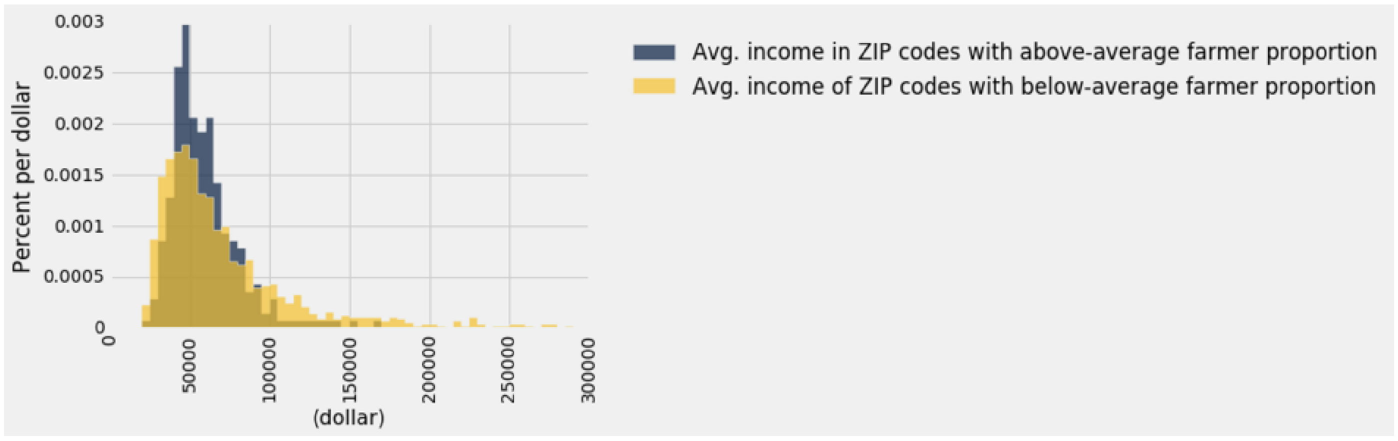
Question 2.9. Compare the average incomes for two groups of *tax returns*: those in ZIP codes with a greater-than-average proportion of farmers and those in ZIP codes with a less-than-average (or average) proportion. Make sure both of these values are displayed (preferably in a table). *Then, describe your findings.*

Hint: Make sure your result correctly accounts for the number of tax returns in each ZIP code, as in questions 2.5 and 2.6.

```
In [ ]: # Build and display a table with two rows:
        #   1) incomes of returns in ZIP codes with a greater-than-average propo
        rtion of farmers
        #   2) incomes of returns in other ZIP codes
```

Write your answer here, replacing this text.

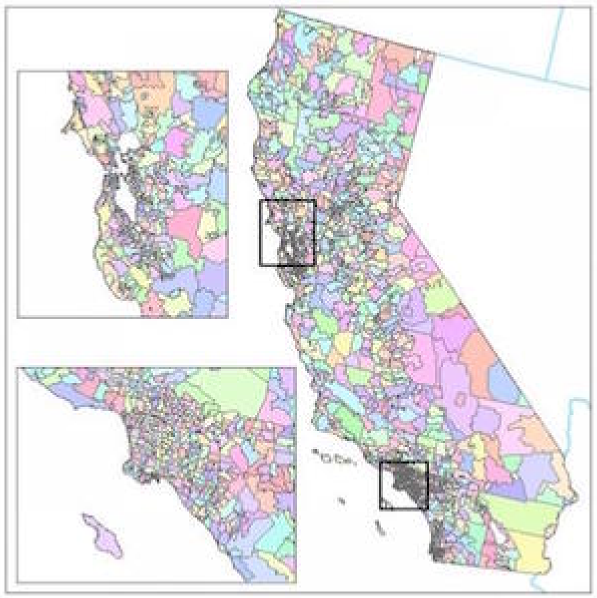
Question 2.10. The graph below displays two histograms: the distribution of average incomes of ZIP codes that have above-average proportions of farmers, and that of ZIP codes with below-average proportions of farmers.



Are ZIP codes with below-average proportions of farmers more or less likely to have very low incomes? Explain how your answer is consistent with your answer to question 2.8.

Write your answer here, replacing this text.

ZIP codes cover all the land in California and do not overlap. Here's a map of all of them.



Question 2.11. Among the ZIP codes represented in the `incomes` table, is there an association between high average income and some aspect of the ZIP code's location? If so, describe one aspect of the location that is clearly associated with high income.

Answer the question by drawing a map of all ZIP codes that have an average income above 100,000 dollars. *Then, describe an association that you observe.*

In order to create a map of certain ZIP codes, you need to:

- Construct a table containing only the ZIP codes of interest, called `high_average_zips`.
- Join `high_average_zips` with the `zip_features` table to find the region for each ZIP code of interest.
- Call `Map(...)` on the column of features (provided).

In [ ]:
```
# Write code to draw a map of only the high-income ZIP codes.
# We have filled in some of it and suggested names for variables
# you might want to define.
zip_features = Table.from_records(zips.features)
high_average_zips = ...
high_zips_with_region = ...
Map(high_zips_with_region.column('feature'), width=400, height=300)
```

Write your answer here, replacing this text.

Part 3: Water Usage

We will now investigate water usage in California. The `usage` table contains three columns:

- `PWSID`: The Public Water Supply Identifier of the district
- `Population`: Estimate of average population served in 2015
- `Water`: Average residential water use (gallons per person per day) in 2014-2015

In [ ]:
```
# Run this cell to create the usage table.

usage_raw.set_format(4, NumberFormatter)
max_pop = usage_raw.select(0, 'population').group(0, max).relabeled(1,
'Population')
avg_water = usage_raw.select(0, 'res_gpcd').group(0,
np.mean).relabeled(1, 'Water')
usage = max_pop.join('pwsid', avg_water).relabeled(0, 'PWSID')
usage
```

Question 3.1. Draw a map of the water districts, colored by the per capita water usage in each district.

Use the `districts.color(...)` method to generate the map. It takes as its first argument a two-column table with one row per district that has the district `PWSID` as its first column. The label of the second column is used in the legend of the map, and the values are used to color each region.

In [ ]:
```
# We have filled in the call to districts.color(...).  Set per_capita_us
age
# to an appropriate table so that a map of all the water districts is
# displayed.
per_capita_usage = ...
districts.color(per_capita_usage, key_on='feature.properties.PWSID')
```

In [ ]:
```
_ = tests.grade('q31')
```

Question 3.2. Based on the map above, which part of California appears to use more water per person: the San Francisco area or the Los Angeles area?

Write your answer here, replacing this text.

Next, we will try to match each ZIP code with a water district. ZIP code boundaries do not always line up with water districts, and one water district often covers multiple ZIP codes, so this process is imprecise. It is even the case that some water districts overlap each other. Nonetheless, we can continue our analysis by matching each ZIP code to the water district with the largest geographic overlap.

The table `wd_vs_zip` describes the proportion of land in each ZIP code that is contained in each water district and vice versa. (The proportions are approximate because they do not correctly account for discontiguous districts, but they're mostly accurate.)

```
In [ ]: wd_vs_zip.show(5)
```

Question 3.3. Complete the `district_for_zip` function that takes a ZIP code as its argument. It returns the PWSID with the largest value of `ZIP in District` for that `zip_code`, if that value is at least 50%. Otherwise, it returns the string `'No District'`.

```
In [ ]: def district_for_zip(zip_code):
            zip_code = str(zip_code) # Ensure that the ZIP code is a string, not
         an integer
            districts = ...
            at_least_half = ...
            if at_least_half:
                ...
            else:
                return 'No District'

        district_for_zip(94709)
```

```
In [ ]: _ = tests.grade('q33')
```

This function can be used to associate each ZIP code in the `income` table with a `PWSID` and discard ZIP codes that do not lie (mostly) in a water district.

```
In [ ]: zip_pwsids = income.apply(district_for_zip, 'ZIP')
        income_with_pwsid = income.with_column('PWSID', zip_pwsids).where('PWSI
        D', are.not_equal_to("No District"))
        income_with_pwsid.set_format(2, NumberFormatter(0)).show(5)
```

Question 3.4. Create a table called `district_data` with one row per PWSID and the following columns:

- `PWSID`: The ID of the district
- `Population`: Population estimate
- `Water`: Average residential water use (gallons per person per day) in 2014-2015
- `Income`: Average income in dollars of all tax returns in ZIP codes that are (mostly) contained in the district according to `income_with_pwsid`.

Hint: First create a `district_income` table that sums the incomes and returns for ZIP codes in each water district.

In [ ]:
```
district_income = ...
district_data = ...
district_data.set_format(make_array('Population', 'Water', 'Income'), Nu
mberFormatter(0))
```

In [ ]:
```
_ = tests.grade('q34')
```

Question 3.5. The `bay_districts` table gives the names of all water districts in the San Francisco Bay Area. Is there an association between water usage and income among Bay Area water districts? Use the tables you have created to compare water usage between the 10 Bay Area water districts with the highest average income and the rest of the Bay Area districts, then describe the association. *Do not include any districts in your analysis for which you do not have income information.*

The names below are just suggestions; you may perform the analysis in any way you wish.

Note: Some Bay Area water districts may not appear in your `district_data` table. That's ok. Perform your analysis only on the subset of districts where you have both water usage & income information.

In [ ]:
```
bay_districts = Table.read_table('bay_districts.csv')
bay_water_vs_income = ...
top_10 = ...
...
```

Complete this one-sentence conclusion: In the Bay Area, people in the top 10 highest-income water districts used an average of __________ more gallons of water per person per day than people in the rest of the districts.

Question 3.6. In one paragraph, summarize what you have discovered through the analyses in this project and suggest what analysis should be conducted next to better understand California water usage, income, and geography. What additional data would be helpful in performing this next analysis?

Replace this line with your conclusion paragraph.

Congratulations - you've finished Project 1 of Data 8!

To submit:

1. Select `Run All` from the `Cell` menu to ensure that you have executed all cells, including the test cells. Make sure that the visualizations you create are actually displayed.
2. Select `Download as PDF via LaTeX (.pdf)` from the `File` menu. (Sometimes that seems to fail. If it does, you can download as HTML, open the .html file in your browser, and print it to a PDF.)
3. Read that file! If any of your lines are too long and get cut off, we won't be able to see them, so break them up into multiple lines and download again. If maps do not appear in the output, that's ok.
4. Submit that downloaded file (called `project1.pdf`) to Gradescope.

If you cannot submit online, come to office hours for assistance. The office hours schedule appears on data8.org/weekly (http://data8.org/weekly).

```
In [ ]: # For your convenience, you can run this cell to run all the tests at on
        ce!
        import os
        _ = [tests.grade(q[:-3]) for q in os.listdir("tests") if
        q.startswith('q')]
```

If you want, draw some more maps below.

```
In [ ]: # Your extensions here (completely optional)
```