Project 1 - California Water Usage

Welcome to your first project! 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 and curated by the Pacific Institute. The map data includes US topography, California counties, and ZIP codes.

The dataset on income comes from the IRS (documented here). 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.

Due Date and Comments

Your project will be due **Friday, March 3 at 11PM**. If you like, you may work with one partner. Outside of your partner, your work should be your own work. **This project has some hard pieces!** It is okay if you do not solve every task, but you should try hard on every one and document your efforts. I and the TAs (Zoe and Lesley) are very happy to help with hints and nudges as well as looking for bugs. If you start early, you should have plenty of time to ask questions.

In [1]:
```
# 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')
%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')
```

First, load the data. Loading may take some time. **You will need to upload the two files from Moodle to the same location on your Jupyter hub that you have Project 1 located.** First download them from Moodle and then use the Upload button on the Jupyter hub to upload them.

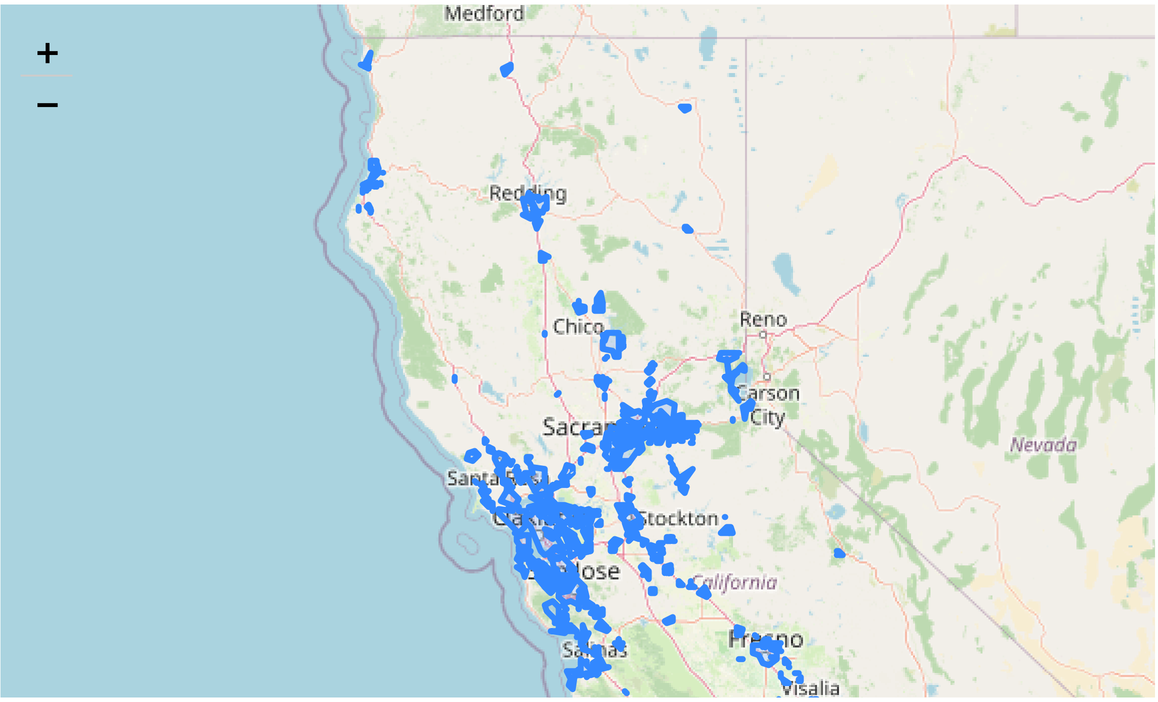
In [2]:
```
# Run this cell, but please don't change it.

districts = Map.read_geojson('water_districts.geojson')
zips = Map.read_geojson('ca_zips.geojson.gz')
usage_raw = Table.read_table('~/DS_113_S23/Projects/Project_1/water_usage.cs
income_raw = Table.read_table('~/DS_113_S23/Projects/Project_1/ca_income_by_
wd_vs_zip = Table.read_table('~/DS_113_S23/Projects/Project_1/wd_vs_zip.csv'
```

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. To view a map of California's water districts, run the cell below. Click on a district to see its description.

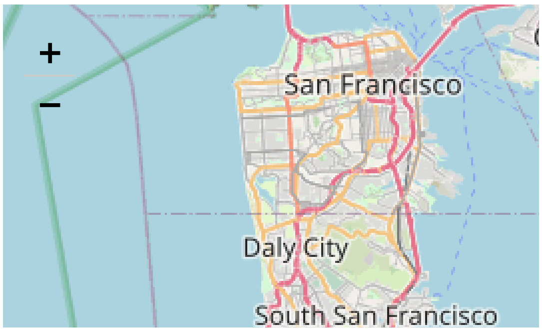
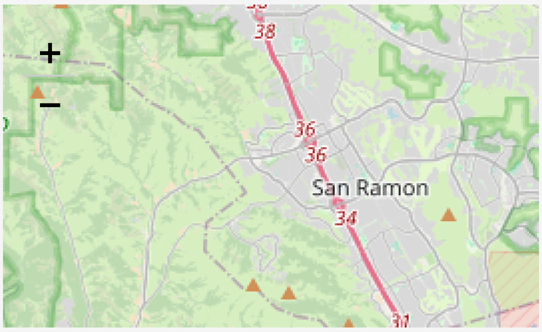
In [3]:
```
districts.format(width=800, height=600)
```

Out[3]:



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 [4]:
```
district_table = Table.from_records(districts.features)
district_table.show(3)
```

PWSID	feature	id	popupContent
0110001		0	Alameda County Water District
0110003		1	California Water Service Company Livermore
0110005		2	East Bay Municipal Utilities District

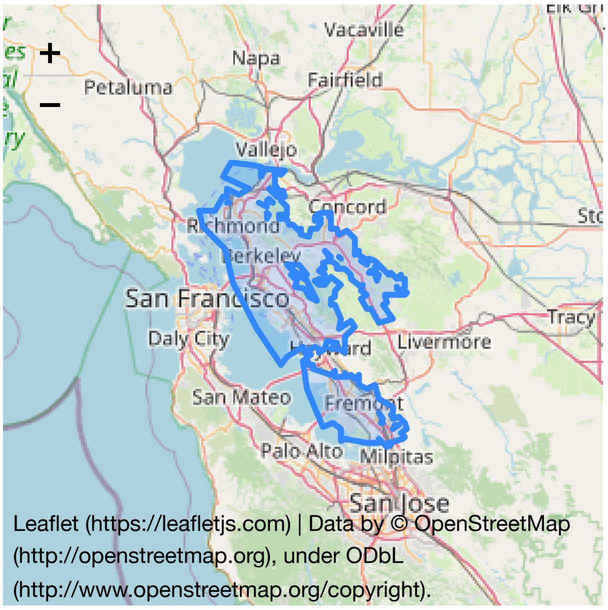
... (407 rows omitted)

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 [5]:
```
# Fill in the next line to make an array containing the two elements from di
# that you want out of the feature column.
alameda_and_east_bay = make_array(district_table["feature"][0],district_tabl
Map(alameda_and_east_bay, height=300, width=300)
```

Out[5]:



-> Trust Notebook

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 [6]:
```
income_raw
```

Out[6]:

ZIP	N1	MARS1	MARS2	MARS4	PREP	N2	NUMDEP	A00100	N02650	A02650
90001	13100	6900	1890	4270	10740	29670	15200	181693	13100	184344
90001	5900	1700	1970	2210	4960	17550	9690	203628	5900	204512
90001	1480	330	760	390	1240	4710	2470	89065	1480	89344
90001	330	50	210	70	290	1100	560	28395	330	28555
90001	160	30	100	40	130	510	250	24676	160	25017
90001	0	0	0	0	0	0	0	0	0	0
90002	12150	6330	1460	4330	9580	27240	14070	167261	12150	170095
90002	5030	1510	1490	1980	4120	14410	7890	173280	5030	174335
90002	1320	300	600	400	1060	4090	2180	78559	1320	78871
90002	340	90	190	90	270	1060	530	28502	340	28558

... (8888 rows omitted)

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 USD to 34,999 USD.)
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 [7]:
```
income_by_zipcode = income_raw.group("ZIP",np.sum)
income_by_zipcode
```

Out[7]:

ZIP	N1 sum	MARS1 sum	MARS2 sum	MARS4 sum	PREP sum	N2 sum	NUMDEP sum	A00100 sum	N02650 sum	
90001	20970	9010	4930	6980	17360	53540	28170	527457	20970	
90002	18960	8230	3830	6800	15120	47200	24850	462823	18960	
90003	26180	11310	5130	9640	20570	64470	33760	612733	26180	
90004	27360	15330	7000	4670	20260	51180	17800	1.61777e+06	27360	1.6
90005	15430	8550	3870	2830	11210	29910	11130	707020	15430	
90006	22630	11470	5400	5630	17840	47590	20210	563530	22630	
90007	11710	6350	2270	3020	8310	23380	9950	311779	11710	
90008	14710	8060	2310	4110	9990	27000	10310	662036	14710	
90010	2210	1270	690	210	1760	3790	960	314333	2210	
90011	36670	15540	8600	12390	30240	95640	51260	857731	36670	

... (1473 rows omitted)

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 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 [8]:
```
for x in income_by_zipcode.labels:
    income_by_zipcode.relabel(x,x.replace(" sum",""))
income_by_zipcode
```

Out[8]:

ZIP	N1	MARS1	MARS2	MARS4	PREP	N2	NUMDEP	A00100	N02650
90001	20970	9010	4930	6980	17360	53540	28170	527457	20970
90002	18960	8230	3830	6800	15120	47200	24850	462823	18960
90003	26180	11310	5130	9640	20570	64470	33760	612733	26180
90004	27360	15330	7000	4670	20260	51180	17800	1.61777e+06	27360 1.6
90005	15430	8550	3870	2830	11210	29910	11130	707020	15430
90006	22630	11470	5400	5630	17840	47590	20210	563530	22630
90007	11710	6350	2270	3020	8310	23380	9950	311779	11710
90008	14710	8060	2310	4110	9990	27000	10310	662036	14710
90010	2210	1270	690	210	1760	3790	960	314333	2210
90011	36670	15540	8600	12390	30240	95640	51260	857731	36670

... (1473 rows omitted)

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 [9]:
```
income = Table().with_columns(
        "ZIP", income_by_zipcode["ZIP"],
        "num returns", income_by_zipcode["N02650"],
        "total income ($)", income_by_zipcode["A02650"],
        "num farmers", income_by_zipcode["SCHF"]
    )
income.set_format('total income ($)', NumberFormatter(0)).show(5)
```

ZIP	num returns	total income ($)	num farmers
90001	20970	531,772	0
90002	18960	467,128	0
90003	26180	618,848	0
90004	27360	1,649,431	0
90005	15430	717,290	0

... (1478 rows omitted)

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 [10]:
```
removeZIP = income.sort("num returns", descending=True)["ZIP"].item(0)
income = income.where("ZIP", are.not_containing(removeZIP))
income
```

Out[10]:

ZIP	num returns	total income ($)	num farmers
90001	20970	531,772	0
90002	18960	467,128	0
90003	26180	618,848	0
90004	27360	1,649,431	0
90005	15430	717,290	0
90006	22630	571,157	0
90007	11710	315,581	0
90008	14710	668,523	0
90010	2210	320,471	0
90011	36670	864,961	0

... (1472 rows omitted)

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 [11]:
```
# total income of the poeple living in 94576
total_income94576 = income.where("ZIP", are.equal_to("94576"))["total income
# tax return of the poeple living in 94576
tax_return94576 = income.where("ZIP", are.equal_to("94576"))["num returns"]
# total income of the poeple living in 94704
total_income94704 = income.where("ZIP", are.equal_to("94704"))["total income
# tax return of the poeple living in 94704
tax_return94704 = income.where("ZIP", are.equal_to("94704"))["num returns"]


average_income = round(sum(total_income94576+total_income94704)*1000/sum(tax
average_income
```

Out[11]: 52773

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 [12]:
```
avg_total = round(sum(income["total income ($)"])*1000/sum(income["num retur
avg_total
```

Out[12]: 72791

Farming

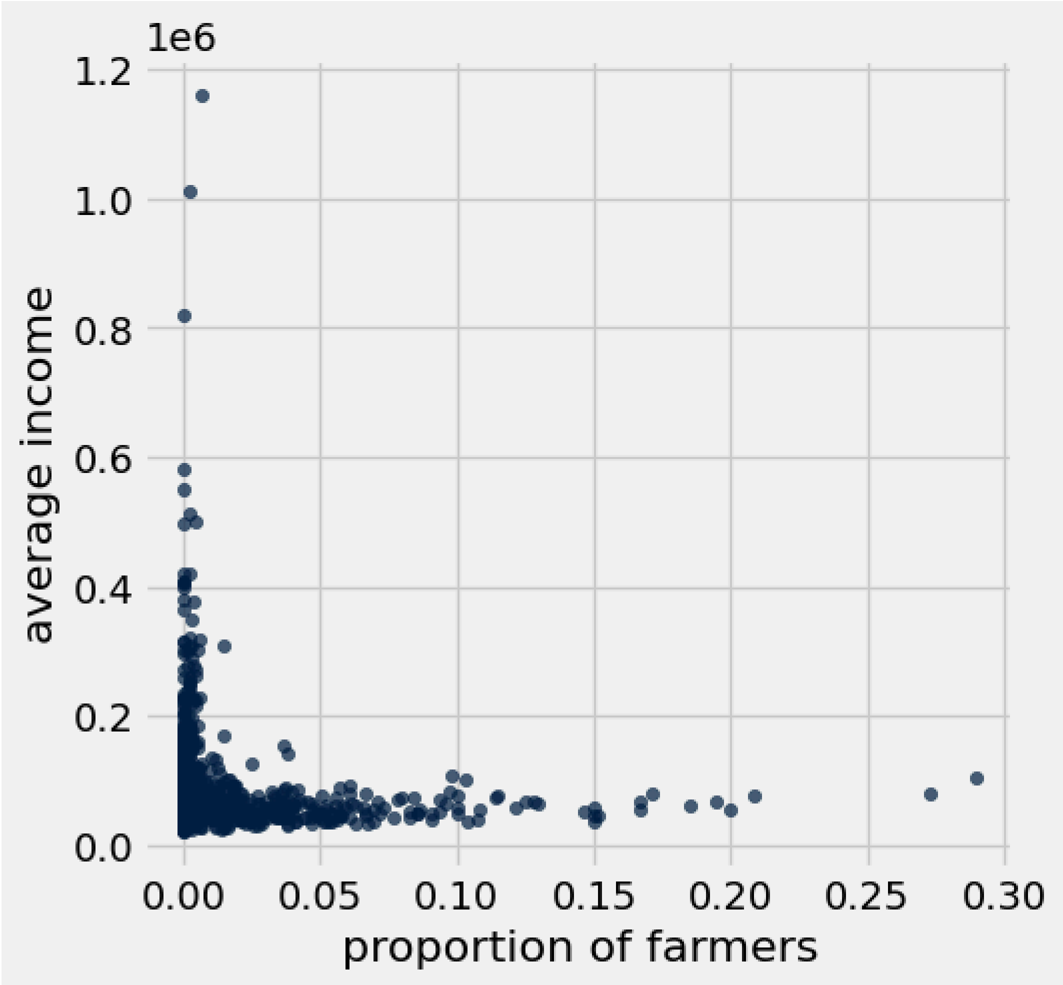
Farms use water, so it's plausible that farming is an important factor in water usage. Here, we will check for a relationship between farming and income.

Among the tax returns in California for ZIP codes represented in the `income` 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 [13]:
```
# make a table containing the average income in dollars for each Zip code
avg_income = income.with_columns("average income", income["total income ($)'
                              "proportion of farmers", income["num farmers'
avg_income.scatter("proportion of farmers", "average income")
```



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?

The range in the average income is greater in the areas with less proportion of farmers than the areas with greater proportion of farmers. The areas with the greater proportion of farmers have a steady average incomes. We cannot say that the ZIP codes with more farmers typically have lower or higher average income because the range in the average income is great in the areas with low number of farmers.

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 [14]:
```
# find the average proportion of farmers
farmer_avg_total = sum(income["num farmers"])/sum(income["num returns"])

# add a column of the average proportion of farmers in the table
income_farmers = income.with_column("average farmers", income["num farmers"]

# table with a greater-than-average proportion of farmers
farmers_greater = income_farmers.where("average farmers", are.above(farmer_a
# get the average incomes of tax returns in dollars rounded to the nearest d
```

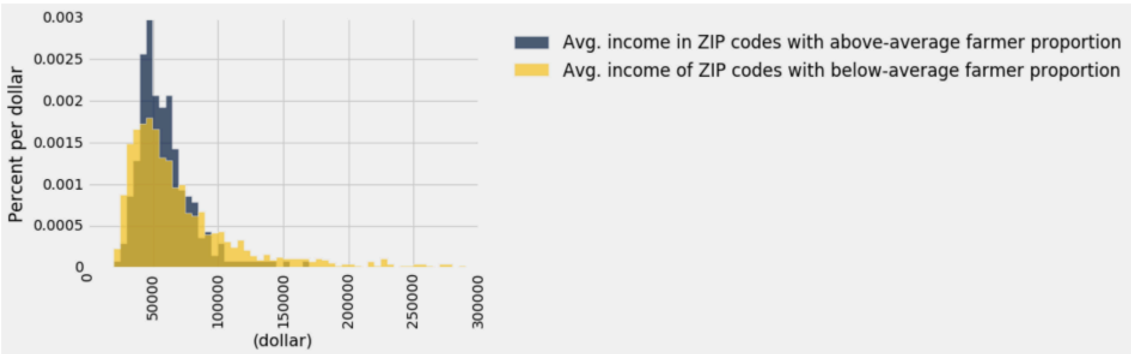
```
income_avg_greater = round(sum(farmers_greater["total income ($)"])*1000/sum

# table with a less-than-average proportion of farmers
farmers_less = income_farmers.where("average farmers", are.below_or_equal_tc
# get the average incomes of tax returns in dollars rounded to the nearest c
income_avg_less = round(sum(farmers_less["total income ($)"])*1000/sum(farme

# Build and display a table with two rows:
#   1) incomes of returns in ZIP codes with a greater-than-average proportic
#   2) incomes of returns in other ZIP codes
avg_income_farmers = Table().with_columns("avg farmer proportion", ["above-a
                                          "average income", [income_avg_great
avg_income_farmers
```

Out[14]:

avg farmer proportion	average income
above-average	78231
below-average	71464

The areas with the average number of farmers greater than the average proportion have the higher average income than the areas with the average number of farmers below the average proportion.

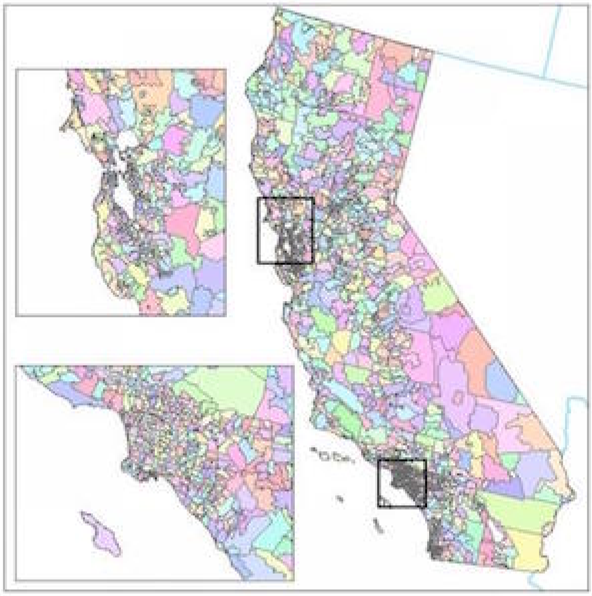
Question 2.10. The graph below displays two histograms: the distribution of average incomes of ZIP codes that have above-average proportions of farmers, and that of ZIP codes with below-average proportions of farmers.



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.

The ZIP codes with below-average proportions of farmers more likely to have very low incomes than the ZIP codes with above-average proportions of farmers. However, the ZIP codes with below-average proportions of farmers have more likely to have high average incomes too. We can also see this from the graph in question 2.7. In the graph, the range of the average incomes are very large when the proportion of farmers are small, which is consistent with the graph in question 2.10.

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 `income` 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.

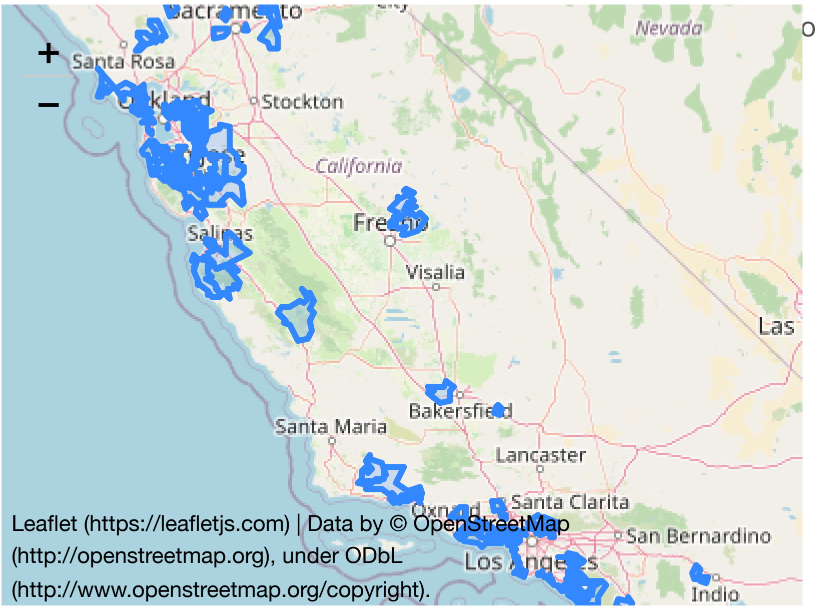
Answer the question by drawing a map of all ZIP codes that have an average income above 100,000 dollars. *Then, describe an association that you observe.*

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 [15]:
```
# 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 = avg_income.where("average income", are.above(100000))
high_zips_with_region = zip_features.join('ZIP', high_average_zips)
Map(high_zips_with_region.column('feature'), width=400, height=300)
```

Out[15]:



ok

Most of the ZIP codes with high income are gathered together near Los Angeles, San Hose, and San Francisco. These cities are located near coast and known to others.

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 [16]:
```
# Run this cell to create the usage table.

usage_raw.set_format(4, NumberFormatter)
max_pop = usage_raw.select(0, 1, 'population').group(0, max).relabeled(2, 'F
avg_water = usage_raw.select(0, 'res_gpcd').group(0, np.mean).relabeled(1, '
usage = max_pop.join('pwsid', avg_water).relabeled(0, 'PWSID').relabeled(1,
usage
```

Out[16]:

PWSID	District	Population	Water
0110001	Alameda County Water District	340000	70.7
0110003	California Water Service Company Livermore	57450	90.2727
0110005	East Bay Municipal Utilities District	1390000	76
0110006	Hayward, City of	151037	57.1818
0110008	Pleasanton, City of	73067	96.6364
0110009	Dublin San Ramon Services District	79547	68.6364
0110011	Livermore, City of	31994	85.8182
0310003	Amador Water Agency	23347	82.8182
0410002	California Water Service Company Chico District	101447	142
0410005	California Water Service Company Oroville	11208	88.8182

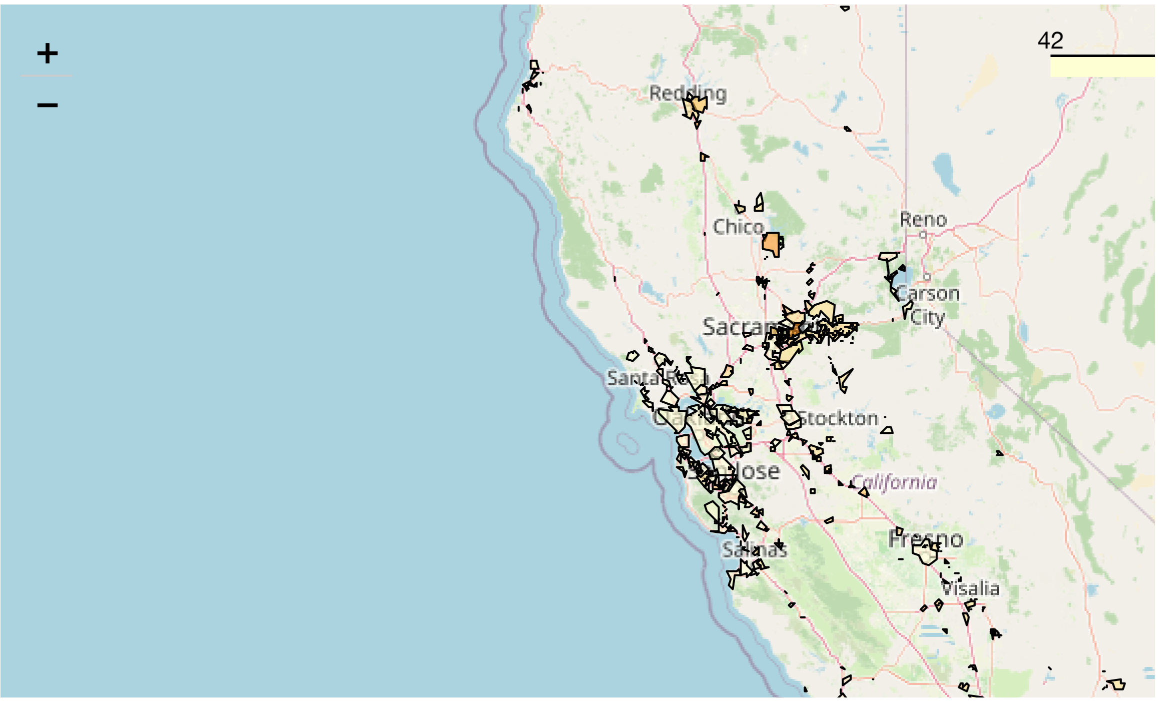
... (401 rows omitted)

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 [17]:
```
# We have filled in the call to districts.color(...).  Set per_capita_usage
# to an appropriate table so that a map of all the water districts is
# displayed.
per_capita_usage = usage.select("PWSID","Water")
districts.color(per_capita_usage, key_on='feature.properties.PWSID')
```

Out[17]:



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?

The Los Angeles area use more water per person. There are more areas with the darker color in the Los Angeles area compared to the San Francisco area.

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 [18]:
```
wd_vs_zip.show(5)
```

PWSID	ZIP	District in ZIP	ZIP in District
0110001	94536	9.41%	68.51%
0110001	94538	18.87%	67.31%
0110001	94539	13.13%	44.36%
0110005	94541	1.61%	68.11%
0110006	94541	18.68%	98.46%

... (3201 rows omitted)

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 [19]:
```
def district_for_zip(zip_code):
    zip_code = str(zip_code) # Ensure that the ZIP code is a string, not an
    districts = wd_vs_zip.where("ZIP", are.equal_to(zip_code)).sort("ZIP in
    at_least_half = districts.column("ZIP in District").item(0)
    if at_least_half >= 0.5:
        return districts["PWSID"].item(0)
    else:
        return 'No District'

district_for_zip(94709)
```

Out[19]: '0110005'

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 [20]:
```
zip_pwsids = income.apply(district_for_zip, 'ZIP')
income_with_pwsid = income.with_column('PWSID', zip_pwsids).where('PWSID', a
income_with_pwsid.set_format(2, NumberFormatter(0)).show(5)
```

ZIP	num returns	total income ($)	num farmers	PWSID
90001	20970	531,772	0	1910067
90022	26680	767,484	0	1910036
90024	14690	4,395,487	20	1910067
90025	25110	4,019,082	20	1910067
90034	29950	1,828,572	0	1910067

... (662 rows omitted)

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 [21]:
```
district_income = income_with_pwsid.group("PWSID", np.sum).drop(1,4)

## add a column with the average income in dollars of all tax returns
district_income = district_income.with_column(
    "average income", district_income["total income ($) sum"]*1000/district_
district_income = district_income.drop(1,2)

district_data = usage.join("PWSID", district_income).relabeled(4, "Income")
district_data
```

Out[21]:

PWSID	District	Population	Water	Income
0110001	Alameda County Water District	340000	70.7	79032
0110005	East Bay Municipal Utilities District	1390000	76	82497.2
0110006	Hayward, City of	151037	57.1818	52923.5
0110008	Pleasanton, City of	73067	96.6364	163257
0110009	Dublin San Ramon Services District	79547	68.6364	133902
0410002	California Water Service Company Chico District	101447	142	50400.8
0410006	South Feather Water and Power Agency	18300	286.2	38720.5
0410011	Del Oro Water Company	9615	92.1818	44706.7
0710001	Antioch, City of	106455	110.273	53550.8
0710003	Contra Costa Water District	197536	101.636	73913.7

... (200 rows omitted)

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 [22]:
```
bay_districts = Table.read_table('~/DS_113_S23/Projects/Project_1/bay_distri

## find the Bay Area districts
bay_district_data = district_data.where("District", are.contained_in(bay_dis
## sort the district_data table for the top average income
bay_district_data = bay_district_data.sort("Income", descending=True)
bay_district_data.show(10)
```

PWSID	District	Population	Water	Income
4110006	California Water Service Company Bear Gulch	58895	170.455	1.16049e+06
4310001	California Water Service Company Los Altos/Suburban	68163	124.545	349183
4310009	Palo Alto, City of	64403	89.1818	334057
4110017	Menlo Park, City of	16066	75.1818	278733
2110002	Marin Municipal Water District	188200	85.6364	176643
0110008	Pleasanton, City of	73067	96.6364	163257
4310007	Mountain View, City of	76781	65.5455	138570
4110021	Estero Municipal Improvement District	37165	66.3636	137893
4110001	Mid-Peninsula Water District	26730	83.7273	137537
4110008	California Water Service Company Mid Peninsula	135918	70.3636	135967

... (19 rows omitted)

Complete this one-sentence conclusion: In the Bay Area, people in the top 10 highest-income water districts used an average of `24.645` more gallons of water per person per day than people in the rest of the districts.

In [30]:
```
# the average of water usage in the top 10 highest-income water districts
top10_avg_water = (sum(bay_district_data["Water"][0:10]))/10
# the average of water usage in the rest of the water districts
rest_avg_water = sum(bay_district_data["Water"][10:len(bay_district_data["Wa

difference = top10_avg_water-rest_avg_water
difference
```

Out[30]: 24.645454545454541

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?

When we see the maps, both the Los Angeles areas and San Francisco areas had the higher average income compared to other areas, but the Los Angeles areas have a higher water usage than the San Francisco areas. However, it is significant that the average income has a positive correlation with the water usage. For the better analysis, it would be good to note the number of farmers for each area, since the farmers use more water than the others and the areas with the higher average income usually did not had a lot of farmers.

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 HTML (.html)` 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. Upload your HTML to Moodle,

If you want, draw some more maps below.

In [24]:
```
# Your extensions here (completely optional)
```