Homework 3 Due Thursday, February 11

Please complete this notebook by filling in the cells provided. When you're done

1. Select `Run All` from the `Cell` menu to ensure that you have executed all cells.
2. Select `Download as PDF via LaTeX (.pdf)` from the `File` menu
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.
4. Submit that downloaded file called `hw03.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).

This assignment is due 5pm Thursday, February 11. You will receive an early submission bonus point if you turn it in by 5pm Wednesday, February 10. Directly sharing answers is not okay, but discussing problems with course staff or students is encouraged.

Reading:

- Textbook section 1.6 (http://www.inferentialthinking.com/chapter1/tables.html)
- Datascience module documentation at data8.org/datascience (http://data8.org/datascience)

0. Preliminaries

Run the cell below to import `Table` and `numpy` and 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 [1]:
```
# Please do not change this cell!

from datascience import *
import numpy as np

from client.api.assignment import load_assignment
hw03 = load_assignment('hw03.ok')
```

The questions in this homework involve a table loaded from presidentialElections.csv (). This table contains historical voting information, which will get us ready for November. It contains five columns:

- A row number
- A state name
- The percentage of votes that supported the *Democratic Party* candidate for president in a particular state and year
- An election year
- Whether the state is in the South

You can view the contents of this file in your browser by executing the cell below. A separate pane will pop up; you can close it by clicking the `x` in its upper-right corner.

In [2]:
```
%less presidentialElections.csv
```

1. Cleaning the Table

Question 1.0. Assign the name `full` to the full table read from the `presidentialElections.csv` file.

In [3]:
```
full = ...
full
```

After each programming question, there will be a test cell that will allow you to check your work. Don't change it; just execute it.

In [4]:
```
_ = hw03.grade('q10')
```

Question 1.1. Set `elections` to a table based on `full` that has five columns, labeled `year`, `state`, and `south`, and `democrat`. The first three columns appear in `full` already. The `democrat` column is based on `full`'s `demVote` column. It should contain the *proportion* of voters for the *Democratic Party* candidate in each state and year, a number between 0 and 1.

In [5]:
```
elections = ...
elections
```

In [6]:
```
_ = hw03.grade('q11')
```

Question 1.2. Set the *format* of the `democrat` column in the `elections` table to be percentages (such as `67.03%` instead of `0.6703`).

In [7]:
```
...
```

In [8]:
```
_ = hw03.grade('q12')
```

2. California History

Now that we have a clean table, it's time to answer some questions with it.

Question 2.0. Assign `cali` to a table with all rows from `elections` that are about `California`. This table should have the same column labels as `elections` (but fewer rows).

In [9]:
```
cali = ...
cali
```

In [10]:
```
_ = hw03.grade('q20')
```

Question 2.1 Assign `most_democratic_cali` to the year in which the highest proportion of Californians voted for the democratic presidential candidate. **Don't just type in the year.** Write an expression that computes it from the `cali` table.

In [11]:
```
most_democratic_cali = ...
most_democratic_cali
```

In [12]:
```
_ = hw03.grade('q21')
```

Question 2.2. Assign `least_democratic_before_1980_ca` to the **year before 1980** in which the lowest proportion of Californians voted for the democratic presidential candidate. **Don't just type in the year.** Write an expression that computes it from the `cali` table. *Hint*: First make a table containing only the years before 1980.

In [13]:
```
cali_before_1980 = ...
least_democratic_before_1980_ca = ...
least_democratic_before_1980_ca
```

In [14]:
```
_ = hw03.grade('q22')
```

3. All States

Here, we will transition back to looking at the `elections` table in order to observe how all of the states voted.

Question 3.0. Assign `dem_years` to a two-column table with one row per year. In addition to the `year` column, the table should have a `dem states` column that has the *number of states for which at least 50% of votes were cast democrat* in each year. The table should be sorted in increasing order of `year`.

```
year | dem states
1932 | 40
1936 | 45
1940 | 38
... (18 rows omitted)
```

Hints:

- First, create a `dems_won` array of True/False values for each state and year that indicates whether democrats won at least 50% of the vote.
- Construct an appropriate table and `group` it by `'year'`.
- Use `np.count_nonzero` to count how many `True` values appear in a sequence of True/False values.

If the table doesn't come out quite right, fix it up using `select` and `relabeled`.

In [15]:
```
dems_won = ...
dem_years = ...
dem_years
```

In [16]:
```
_ = hw03.grade('q30')
```

Question 3.1. Assign `state_spread` to a two-column table with one row per state. In addition to the `state` column, the table should have a `democrat spread` column that contains the difference between the maximal and minimal historic proportion of democratic votes.

For example, the `California` row of the `democrat spread` column should have 0.3104, the difference between .6695 (California in 1936) and .3591 (California in 1980).

The `spread` function from lecture is defined below. *Hint*: You can use `spread` in a call to `group`.

In [17]:
```
def spread(values):
    return max(values) - min(values)

state_spread = ...
state_spread
```

In [18]:
```
_ = hw03.grade('q31')
```

Question 3.2. Do Southern states vote differently from the rest of the states? Is there a time period in which the difference is large? Explore the `elections` table until you discover an *association* and then describe that association.

- `np.average` computes the average of a sequence of numbers
- `np.median` computes the median of a sequence of numbers

In [30]:
```
# Write code that demonstrates an association
after_1970 = elections.where(elections.column('year') > 1970)
print(np.average(after_1970.where('south', True).column('democrat')))
print(np.average(after_1970.where('south', False).column('democrat')))
```

Briefly describe the association you have illustrated:

...

4. Functions

Question 4.0 Complete the `most_democratic_state` function below. It accepts an input `year` (an `int`). It returns the name of the state that had the highest proportion of democratic votes that year. Use the `elections` table to compute the result.

In [20]:
```
def most_democratic_state(year):
    ...
```

In [21]:
```
_ = hw03.grade('q40')
```

Question 4.1 Complete the `shape` function, which accepts a table `t` as input and returns a string describing the shape of `t`:

- More rows than columns: `return 'tall'`
- More columns than rows: `return 'wide'`
- Equal number of rows and columns: `return 'square'`

In [22]:
```
def shape(t):
    ...
        ...
    ...
        ...
    ...
        ...
```

In [23]:
```
_ = hw03.grade('q41')
```