project2

November 9, 2016

1 Project 2: Inference and Capital Punishment

Welcome to Project 2! You will investigate the relationship between murder and capital punishment (the death penalty) in the United States. By the end of the project, you should know how to:

1. Test whether observed data appears to be a random sample from a distribution
2. Analyze a natural experiment
3. Implement and interpret a sign test
4. Create a function to run a general hypothesis test
5. Analyze visualizations and draw conclusions from them

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 5) are due **Tuesday, November 1st at 7PM**. 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, 3, 4, and 5) will be due **Tuesday, November 9th at 7PM**. (Again, see the note above on submitting with a partner.)

On to the project! **Run the cell below** to prepare the automatic tests. The automated tests for this project **definitely don't** catch all possible errors; they're designed to help you avoid some common mistakes. Merely passing the tests does not guarantee full credit on any question.

```
In [204]: # Run this cell to set up the notebook, but please don't change it.
          import numpy as np
          from datascience import *

          # These lines do some fancy plotting magic.
          import matplotlib
          %matplotlib inline
          import matplotlib.pyplot as plt
          plt.style.use('fivethirtyeight')
          import warnings
          warnings.simplefilter('ignore', FutureWarning)

          from client.api.assignment import load_assignment
          tests = load_assignment('project2.ok')
```

```
=====================================================================
Assignment: Project 2: Inference and Capital Punishment
OK, version v1.6.4
=====================================================================
```

1.1 1. Murder rates

Punishment for crime has many philosophical justifications. An important one is that fear of punishment may *deter* people from committing crimes.

In the United States, some jurisdictions execute some people who are convicted of particularly serious crimes, like murder. This punishment is called the *death penalty* or *capital punishment*. The death penalty is controversial, and deterrence has been one focal point of the debate. There are other reasons to support or oppose the death penalty, but in this project we'll focus on deterrence.

The key question about deterrence is:

> Does instituting a death penalty for murder actually reduce the number of murders?

You might have a strong intuition in one direction, but the evidence turns out to be surprisingly complex. Different sides have variously argued that the death penalty has no deterrent effect and that each execution prevents 8 murders, all using statistical arguments! We'll try to come to our own conclusion.

Here is a road map for this project:

1. In the rest of this section, we'll investigate the main dataset we'll be using.
2. In section 2, we'll see how to test null hypotheses like this: "For this set of U.S. states, the murder rate was equally likely to go up or down each year."
3. In section 3, we'll apply a similar test to see whether U.S. states that suddenly ended or reinstituted the death penalty were more likely to see murder rates increase than decrease.

4. In section 4, we will run some more tests to further claims we had been developing in previous sections.
5. In section 5, we'll try to answer our question about deterrence using a visualization rather than a formal hypothesis test.

The data The main data source for this project comes from a paper by three researchers, Dezhbakhsh, Rubin, and Shepherd. The dataset contains rates of various violent crimes for every year 1960-2003 (44 years) in every US state. The researchers compiled their data from the FBI's Uniform Crime Reports.

Since crimes are committed by people, not states, we need to account for the number of people in each state when we're looking at state-level data. Murder rates are calculated as follows:

$$\text{murder rate for state X in year Y} = \frac{\text{number of murders in state X in year Y}}{\text{population in state X in year Y}} * 100000$$

(Murder is rare, so we multiply by 100,000 just to avoid dealing with tiny numbers.)

```
In [205]: murder_rates = Table.read_table('crime_rates.csv').select('State', 'Year'
          murder_rates.set_format("Population", NumberFormatter)
```

```
Out[205]: State   | Year | Population | Murder Rate
          Alaska  | 1960 | 226,167    | 10.2
          Alaska  | 1961 | 234,000    | 11.5
          Alaska  | 1962 | 246,000    | 4.5
          Alaska  | 1963 | 248,000    | 6.5
          Alaska  | 1964 | 250,000    | 10.4
          Alaska  | 1965 | 253,000    | 6.3
          Alaska  | 1966 | 272,000    | 12.9
          Alaska  | 1967 | 272,000    | 9.6
          Alaska  | 1968 | 277,000    | 10.5
          Alaska  | 1969 | 282,000    | 10.6
          ... (2190 rows omitted)
```

So far, this looks like a dataset that lends itself to an observational study. In fact, these data aren't even enough to demonstrate an *association* between the existence of the death penalty in a state in a year and the murder rate in that state and year!

Question 1.1. What additional information will we need before we can check for that association?

Answer: We need to know whether or not the respective state implemented the death penalty or not.

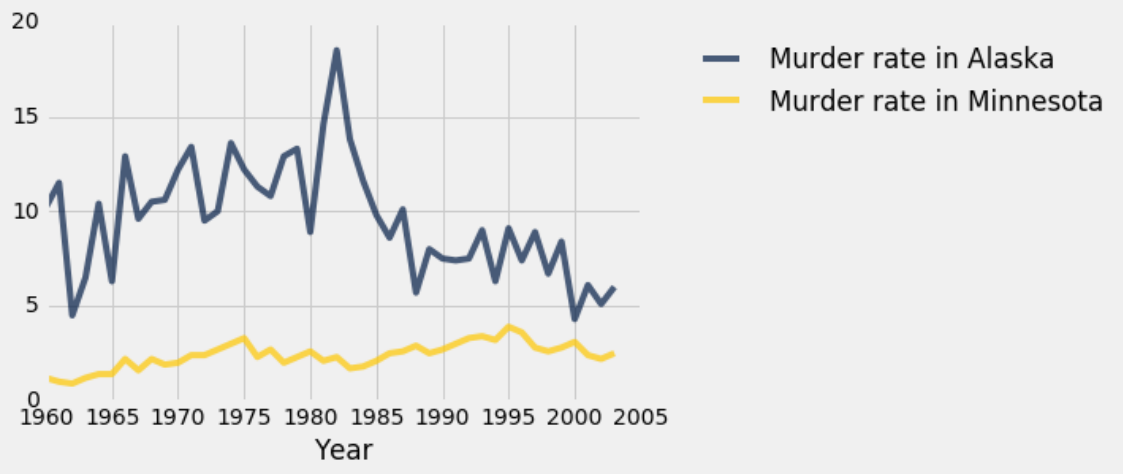
Murder rates vary over time, and different states exhibit different trends. The rates in some states change dramatically from year to year, while others are quite stable. Let's plot a couple, just to see the variety.

Question 1.2. Draw a line plot with years on the horizontal axis and murder rates on the vertical axis. Include two lines: one for Alaska murder rates and one for Minnesota murder rates. Create this plot using a single call, `ak_mn.plot('Year')`.

Hint: To create two lines, you will need create the table `ak_mn` with two columns of murder rates, in addition to a column of years. You can use `join` to create this table, which will have the following structure:

Year	Murder rate in Alaska	Murder rate in Minnesota
1960	10.2	1.2
1961	11.5	1
1962	4.5	0.9

```
In [206]: # The next lines are provided for you.  They create a table
          # containing only the Alaska information and one containing
          # only the Minnesota information.
          ak = murder_rates.where('State', 'Alaska').drop('State', 'Population').re
          mn = murder_rates.where('State', 'Minnesota').drop('State', 'Population')

          # Fill in this line to make a table like the one pictured above.
          ak_mn = ak.join("Year", mn, "Year")
          ak_mn.plot('Year')
```




```
In [207]: _ = tests.grade('q1_1_2')

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed
```

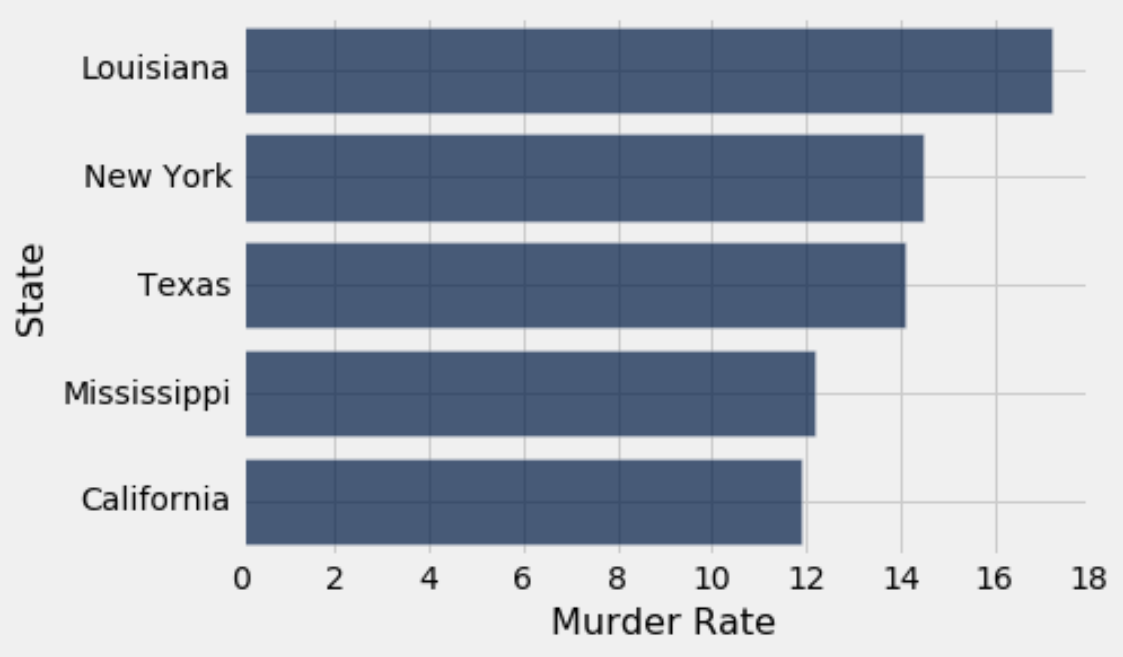
A reminder about tests The automated tests check for basic errors (like the number of rows in your `ak_mn` table, or whether you defined a function named `most_murderous` for the next question), but they **aren't comprehensive**.

If you're not sure that your answer is correct, think about how you can check it. For example, if a table has the right number of rows and columns, and a few randomly-selected values from each column are correct, then you can be somewhat confident you've computed it correctly. For the previous question, try checking some of the values in `ak_mn` manually, by searching through the `murder_rates` table.

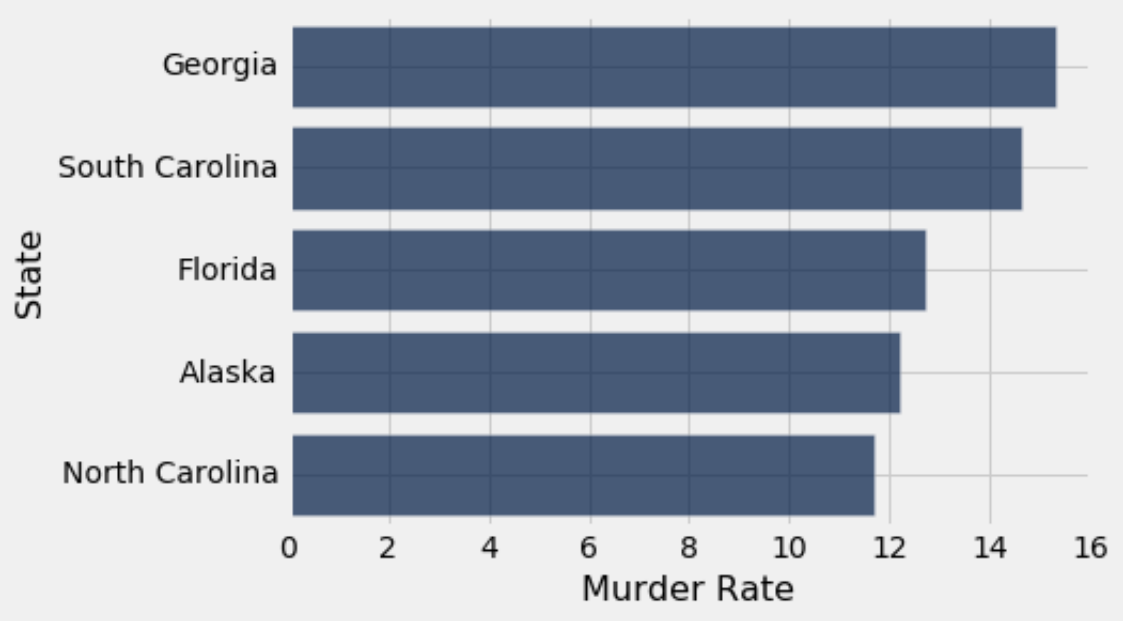
Question 1.3. Implement the function `most_murderous`, which takes a year (an integer) as its argument. It does two things: 1. It draws a horizontal bar chart of the 5 states that had the highest murder rate in that year. 2. It returns an array of the names of these states in order of *increasing* murder rate.

If the argument isn't a year in `murder_rates`, your function can do anything.

```
In [208]: def most_murderous(year):
              # Fill in this line so that the next 2 lines do what the function
              # is supposed to do.  most should be a table.
              most = murder_rates.where("Year", are.equal_to(year)).sort("Murder Ra
                      descending = True).take(np.arange(5))
              most.barh('State', 'Murder Rate')
              return most.column('State')

          most_murderous(1990)

Out[208]: array(['Louisiana', 'New York', 'Texas', 'Mississippi', 'California'],
                dtype='<U14')
```



```
In [209]: _ = tests.grade('q1_1_3')

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed
```



Question 1.4. How many more people were murdered in California in 1988 than in 1975? Assign `ca_change` to the answer.

Hint: Consider using the formula in the beginning of the section to answer this question.

```
In [210]: ca = murder_rates.where('State', are.equal_to('California'))
          ca1988 = ca.where("Year", are.equal_to(1988))
          ca1975 = ca.where("Year", are.equal_to(1975))
          ca1988_murders = ca1988.column("Population")[0]*ca1988.column("Murder Rat
          ca1975_murders = ca1975.column("Population")[0]*ca1975.column("Murder Rat
          ca_change = ca1988_murders - ca1975_murders
          ca_change
          np.round(ca_change)

Out[210]: 726.0

In [211]: _ = tests.grade('q1_1_4')
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed
```

Certain mistakes would make your answer to the previous question way too small or way too big, and the automatic tests don't check that. Make sure your answer looks reasonable after carefully reading the question.

2 2. Changes in Murder Rates

Murder rates vary widely across states and years, presumably due to the vast array of differences among states and across US history. Rather than attempting to analyze rates themselves, here we will restrict our analysis to whether or not murder rates increased or decreased over certain time spans. We will not concern ourselves with how much rates increased or decreased; only the direction of the change - *whether* they increased or decreased.

The `np.diff` function takes an array of values and computes the differences between adjacent items of a list or array. Instead, we may wish to compute the difference between items that are two positions apart. For example, given a 5-element array, we may want:

```
[item 2 - item 0 , item 3 - item 1 , item 4 - item 2]
```

The `diff_n` function below computes this result. Don't worry if the implementation doesn't make sense to you, as long as you understand its behavior.

```
In [212]: def diff_n(values, n):
              return np.array(values)[n:] - np.array(values)[:-n]

          diff_n(make_array(1, 10, 100, 1000, 10000), 2)

Out[212]: array([  99,  990, 9900])
```

Question 2.1. Implement the function `two_year_changes` that takes an array of murder rates for a state, ordered by increasing year. For all two-year periods (e.g., from 1960 to 1962), it computes and returns the number of increases minus the number of decreases.

For example, the rates `r = make_array(10, 7, 12, 9, 13, 9, 11)` contain three increases (10 to 12, 7 to 9, and 12 to 13), one decrease (13 to 11), and one change that is neither an increase or decrease (9 to 9). Therefore, `two_year_changes(r)` would return 2, the difference between three increases and 1 decrease.

```
In [213]: def two_year_changes(rates):
              "Return the number of increases minus the number of decreases after t
```

```
        differences = diff_n(rates, 2)
        return np.count_nonzero(differences > 0) - np.count_nonzero(differenc
    print('Alaska:',    two_year_changes(ak.column('Murder rate in Alaska')))
    print('Minnesota:', two_year_changes(mn.column('Murder rate in Minnesota'
```

```
Alaska: -5
Minnesota: 6
```

```
In [214]: _ = tests.grade('q1_2_1')
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed
```

We can use `two_year_changes` to summarize whether rates are mostly increasing or decreasing over time for some state or group of states. Let's see how it varies across the 50 US states.

Question 2.2. Assign `changes_by_state` to a table with one row per state that has two columns: the `State` name and the `Murder Rate two_year_changes` statistic computed across all years in our data set for that state. Its first 2 rows should look like this:

State	Murder Rate two_year_changes
Alabama	-6
Alaska	-5

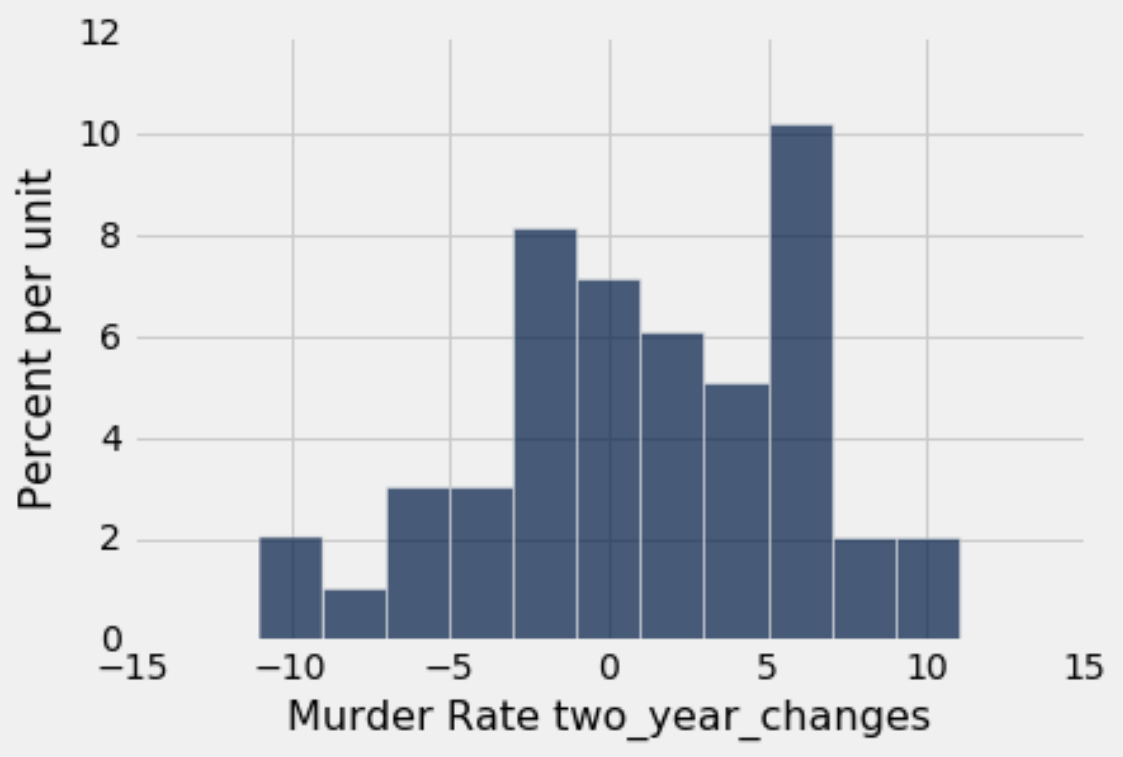
```
In [215]: states = murder_rates.group("State").column(0)
          empty = make_array()

          for i in states:
              state_rates = murder_rates.where("State", are.equal_to(i))
              array_rates = two_year_changes(state_rates.column("Murder Rate"))
              empty = np.append(empty, array_rates)

          changes_by_state = Table().with_columns('State', states, "Murder Rate two
          changes_by_state

          #year = murder_rates.group("State", murder_rates, )
          #changes_by_state = two_year_changes(murder_rates)
```

```
# Here is a histogram of the two-year changes for the states.
# Since there are 50 states, each state contributes 2% to one
# bar.
changes_by_state.hist("Murder Rate two_year_changes", bins=np.arange(-11,
changes_by_state
```

Out[215]:

State	Murder Rate two_year_changes
Alabama	-6
Alaska	-5
Arizona	1
Arkansas	-1
California	17
Colorado	-4
Connecticut	4
Delaware	-3
Florida	-6
Georgia	-3

... (40 rows omitted)



Some states have more increases than decreases (a positive number), while some have more decreases than increases (a negative number).

Question 2.3. Assign `total_changes` to the total increases minus the total decreases for all two-year periods and all states in our data set.

```
In [216]: total_changes = sum(changes_by_state.column(1))
          print('Total increases minus total decreases, across all states and years
```

```
Total increases minus total decreases, across all states and years: 45.0
```

"More increases than decreases," one student exclaims, "Murder rates tend to go up across two-year periods. What dire times we live in."

"Not so fast," another student replies, "Even if murder rates just moved up and down uniformly at random, there would be some difference between the increases and decreases. There were a lot of states and a lot of years, so there were many chances for changes to happen. Perhaps this difference we observed is a typical value when so many changes are observed if the state murder rates increase and decrease at random!"

Question 2.4. Set `num_changes` to the number of different two-year periods in the entire data set that could result in a change of a state's murder rate. Include both those periods where a change occurred and the periods where a state's rate happened to stay the same.

For example, 1968 to 1970 of Alaska would count as one distinct two-year period.

```
In [217]: num_changes = murder_rates.num_rows - 100
          num_changes

Out[217]: 2100

In [218]: _ = tests.grade('q1_2_4')

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed
```

We now have enough information to perform a hypothesis test.

> **Null Hypothesis**: State murder rates increase and decrease over two-year periods as if "increase" or "decrease" were sampled at random from a uniform distribution, like a fair coin flip.

Since it's possible that murder rates are more likely to go up or more likely to go down, our alternative hypothesis should contemplate either case:

> **Alternative Hypothesis**: State murder rates are *either* more likely or less likely to increase than decrease over two-year periods.

Technical note: These changes in murder rates are not random samples from any population. They describe all murders in all states over all recent years. However, we can imagine that history could have been different, and that the observed changes are the values observed in only one possible world: the one that happened to occur. In this sense, we can evaluate whether the observed

"total increases minus total decreases" is consistent with a hypothesis that increases and decreases are drawn at random from a uniform distribution.

Question 2.5 Given these null and alternative hypotheses, define a good test statistic.

Important requirements for your test statistic: Choose a test statistic for which large positive values are evidence in favor of the alternative hypothesis, and other values are evidence in favor of the null hypothesis. Your test statistic should depend only on whether murder rates increased or decreased, not on the size of any change.

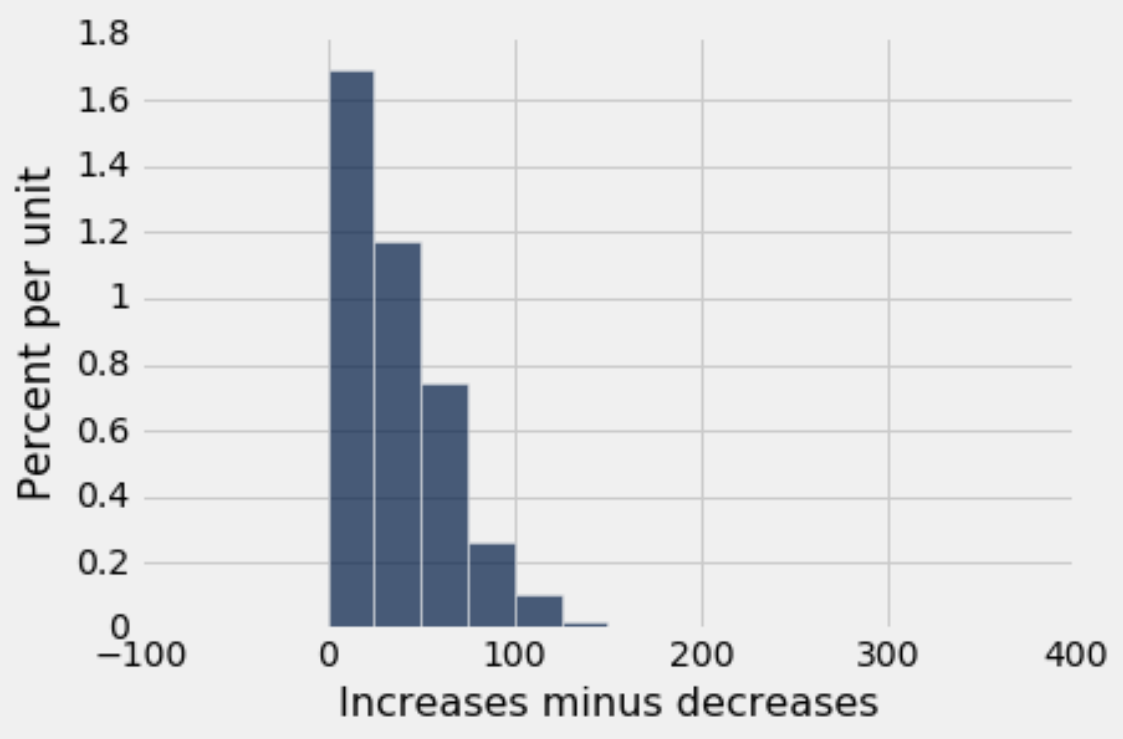
Absolute value of increases minus decreases.

The cell below samples increases and decreases at random from a uniform distribution 100 times. The final column of the resulting table gives the number of increases and decreases that resulted from sampling in this way.

```
In [219]: uniform = Table().with_columns(
              "Change", make_array('Increase', 'Decrease'),
              "Chance", make_array(0.5,        0.5))
          uniform.sample_from_distribution('Chance', 100)

Out[219]: Change   | Chance | Chance sample
          Increase | 0.5    | 58
          Decrease | 0.5    | 42
```

Question 2.6. Complete the simulation below, which samples `num_changes` increases/decreases at random many times and forms an empirical distribution of your test statistic under the null hypothesis. Your job is to * fill in the function `simulate_under_null`, which simulates a single sample under the null hypothesis, and * fill in its argument when it's called below.

```
In [220]: def simulate_under_null(num_chances_to_change):
              """Simulates some number changing several times, with an equal
              chance to increase or decrease.  Returns the value of your
              test statistic for these simulated changes.

              num_chances_to_change is the number of times the number changes.
              """
              sample = uniform.sample_from_distribution("Chance", num_chances_to_ch
              increases = sample.item(0)
              decreases = sample.item(1)
              return abs(increases - decreases)

          uniform_samples = make_array()
          for i in np.arange(5000):
              uniform_samples = np.append(uniform_samples, simulate_under_null(num_
          Table().with_column('Increases minus decreases', uniform_samples).hist(0,
```



Question 2.7. Looking at this histogram, draw a conclusion about whether murder rates basically increase as often as they decrease. (You *do not* need to compute a P-value for this question.)

Answer: The distribution is centered around 0, stretching out into the positive values because the statistic was calculated as an absolute value, which suggests that murder rates basically increase as often as they decrease with minor variation due to chance.

3 3. The death penalty

Some US states have the death penalty, and others don't, and laws have changed over time. In addition to changes in murder rates, we will also consider whether the death penalty was in force in each state and each year.

Using this information, we would like to investigate how the death penalty affects the murder rate of a state.

Question 3.1. Describe this investigation in terms of an experiment. What population are we studying? What is the control group? What is the treatment group? What outcome are we measuring?

Write your answers below.

- Population: The population in the 50 US states
- Control Group: The group of states when the death penalty was not instituted
- Treatment Group: The group of states when the death penalty was instituted
- Outcome: Whether there was an increase or decrease in the murder rate given the implementation or lack of implementation of the death penalty.

Question 3.2. We want to know whether the death penalty *causes* a change in the murder rate. Why is it not sufficient to compare murder rates in places and times when the death penalty was in force with places and times when it wasn't?

There could be other forces at play other than simply the murder rate and the death penalty. For example, there could be specific policies in play that affect the issue or seasonal issues that might not be in place in every state, that could impact the results of the study.

3.0.1 A Natural Experiment

In order to attempt to investigate the causal relationship between the death penalty and murder rates, we're going to take advantage of a *natural experiment*. A natural experiment happens when something other than experimental design applies a treatment to one group and not to another (control) group, and we can reasonably expect that the treatment and control groups don't have any other systematic differences.

Our natural experiment is this: in 1972, a Supreme Court decision called *Furman v. Georgia* banned the death penalty throughout the US. Suddenly, many states went from having the death penalty to not having the death penalty.

As a first step, let's see how murder rates changed before and after the court decision. We'll define the test as follows:

> **Population:** All the states that had the death penalty before the 1972 abolition. (There is no control group for the states that already lacked the death penalty in 1972, so we must omit them.) This includes all US states **except** Alaska, Hawaii, Maine, Michigan, Wisconsin, and Minnesota.
>
> **Treatment group:** The states in that population, in the year after 1972.
>
> **Control group:** The states in that population, in the year before 1972.
>
> **Null hypothesis:** Each state's murder rate was equally likely to be higher or lower in the treatment period than in the control period. (Whether the murder rate increased or decreased in each state was like the flip of a fair coin.)
>
> **Alternative hypothesis:** The murder rate was more likely to increase *or* more likely to decrease.

Technical Note: It's not clear that the murder rates were a "sample" from any larger population. Again, it's useful to imagine that our data could have come out differently and to test the null hypothesis that the murder rates were equally likely to move up or down.

The `death_penalty` table below describes whether each state allowed the death penalty in 1971.

```
In [221]: non_death_penalty_states = make_array('Alaska', 'Hawaii', 'Maine', 'Michi
          def had_death_penalty_in_1971(state):
              """Returns True if the argument is the name of a state that had the d
              # The implementation of this function uses a bit of syntax
              # we haven't seen before.  Just trust that it behaves as its
              # documentation claims.
              return state not in non_death_penalty_states
```

```
In [221]: states = murder_rates.group('State').select('State')
          death_penalty = states.with_column('Death Penalty', states.apply(had_deat
          death_penalty

Out[221]: State       | Death Penalty
          Alabama     | True
          Alaska      | False
          Arizona     | True
          Arkansas    | True
          California  | True
          Colorado    | True
          Connecticut | True
          Delaware    | True
          Florida     | True
          Georgia     | True
          ... (40 rows omitted)

In [222]: num_death_penalty_states = death_penalty.where("Death Penalty", are.equal
          num_death_penalty_states

Out[222]: 44
```

Question 3.3. Assign `death_penalty_murder_rates` to a table with the same columns and data as `murder_rates`, but that has only the rows for states that had the death penalty in 1971.

The first 2 rows of your table should look like this:

State	Year	Population	Murder Rate
Alabama	1960	3266740	12.4
Alabama	1961	3302000	12.9

```
In [223]: # The staff solution used 3 lines of code.
          penalty_states = death_penalty.where("Death Penalty", are.equal_to(True))
          death_penalty_murder_rates = murder_rates.where("State", are.equal_to(per
          death_penalty_murder_rates

Out[223]: State   | Year | Population | Murder Rate
          Alabama | 1960 | 3,266,740  | 12.4
          Alabama | 1961 | 3,302,000  | 12.9
          Alabama | 1962 | 3,358,000  | 9.4
          Alabama | 1963 | 3,347,000  | 10.2
          Alabama | 1964 | 3,407,000  | 9.3
          Alabama | 1965 | 3,462,000  | 11.4
          Alabama | 1966 | 3,517,000  | 10.9
          Alabama | 1967 | 3,540,000  | 11.7
          Alabama | 1968 | 3,566,000  | 11.8
```

```
Alabama | 1969 | 3,531,000  | 13.7
... (1926 rows omitted)
```

The null hypothesis doesn't specify *how* the murder rate changes; it only talks about increasing or decreasing. So, we will use the same test statistic you defined in section 2.

Question 3.4. Assign `changes_72` to the value of the test statistic for the years 1971 to 1973 and the states in `death_penalty_murder_rates`.

Hint: You have already written nearly the same code in a previous part of this project.

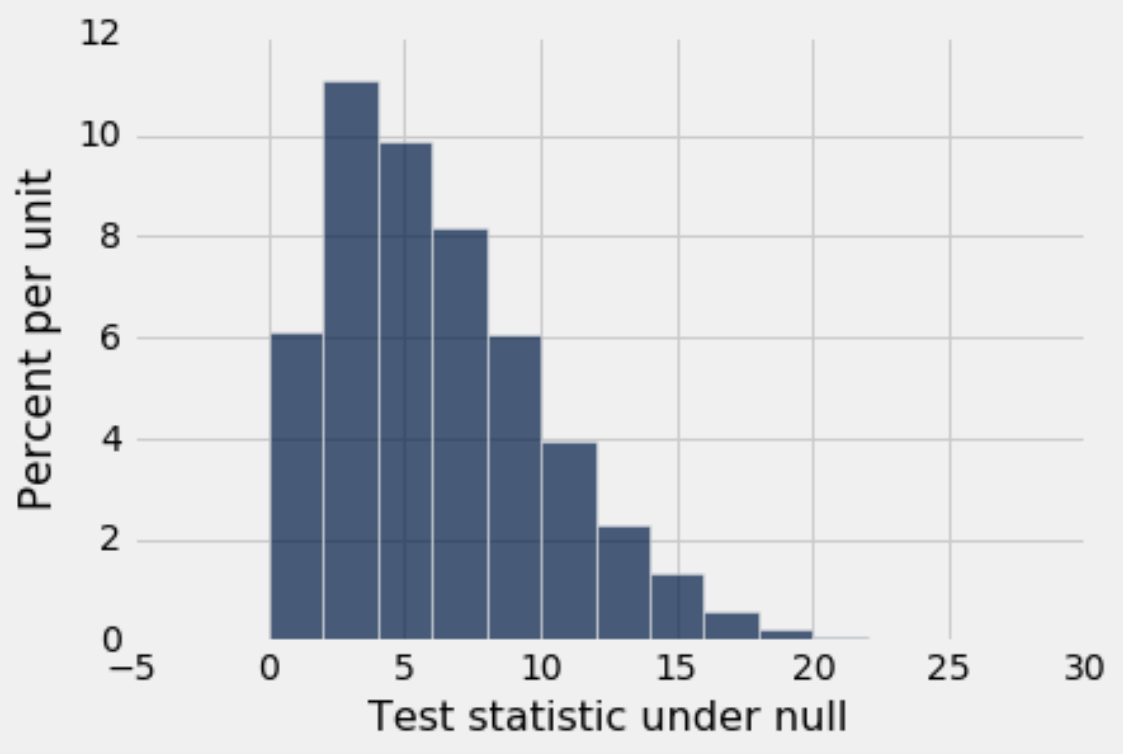
```
In [224]: # The staff solution took 5 lines of code.
          yr_1972 = death_penalty_murder_rates.where("Year", are.between(1971,1974)
          #states_72 = yr_1972.group("State", two_year_changes())
          test_stat_72 = make_array()
          for state in yr_1972.group("State").column(0):
              changes = two_year_changes(yr_1972.where("State",
                      are.equal_to(state)).column("Murder Rate"))
              test_stat_72 = sum(np.append(test_stat_72, changes))


          print('Increases minus decreases from 1971 to 1973:', test_stat_72)

Increases minus decreases from 1971 to 1973: 22.0
```

Look at the data (or perhaps a random sample!) to verify that your answer is correct.

Question 3.5.: Draw an empirical histogram of the statistic under the null hypothesis by simulating the test statistic 5,000 times.

Hint: In a previous part of this project, you have already written a function that runs such a simulation once.

```
In [225]: samples = make_array()
          for i in np.arange(10000):
              samples = np.append(samples, simulate_under_null(num_death_penalty_st
          # Feel free to change the bins if they don't make sense for your test sta
          Table().with_column('Test statistic under null', samples).hist(bins=np.ar
```




```
In [226]: _ = tests.grade('q1_3_5')

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed
```

3.0.2 Conclusion

Question 3.6. Complete the analysis as follows: 1. Compute a P-value. 2. Draw a conclusion about the null and alternative hypotheses. 3. Describe your findings using simple, non-technical language. Be careful not to claim that the statistical analysis has established more than it really has.

P-value: 0.0011

Conclusion about the hypotheses: We accept the alternative hypothesis because the value is extreme as the p value denotes. Given a p-value cutoff of 1%, we would fail to reject the null, and thus accept the alternative.

Findings: This means that the change in murder rates is not due to chance and that murder rates are related to whether or not the death penalty was instituted.

```
In [227]: np.count_nonzero(samples>=22)/10000
```

```
Out[227]: 0.0011
```

4 4. Further evidence

So far, we have discovered evidence that when executions were outlawed, the murder rate increased in many more states than we would expect from random chance. We have also seen that across all states and all recent years, the murder rate goes up about as much as it goes down over two-year periods.

These discoveries seem to support the claim that eliminating the death penalty increases the murder rate. Should we be convinced? Let's conduct some more tests to strengthen our claim.

Conducting a test for this data set required the following steps:

1. Select a table containing murder rates for certain states and all years,
2. Choose two years and compute the observed value of the test statistic,
3. Simulate the test statistic under the null hypothesis that increases and decreases are drawn uniformly at random, then
4. Compare the observed difference to the empirical distribution to compute a P-value.

This entire process can be expressed in a single function, called `run_test`.

Question 4.1. Implement `run_test`, which takes the following arguments:

- A table of murder `rates` for certain states, sorted by state and year like `murder_rates`, and
- the year when the analysis starts. (The comparison group is two years later.)

It prints out the observed test statistic and returns the P-value for this statistic under the null hypothesis.

Hint 1: You can complete most of this question by copying code you wrote earlier.

Hint 2: This problem might seem daumting. Start by writing out the different steps involved in running a test.

```
In [228]: proper_dates = death_penalty_murder_rates.where("Year", are.between(1971,
                                        1973+ 1))
          two_year_changes(proper_dates.where("State", are.equal_to(state)).column
```

```
Out[228]: 1
```

```
In [229]: def run_test(rates, start_year):
              """Return a P-value for the observed difference between increases and
              end_year = start_year + 2
              observed_test_statistic = make_array()
              proper_dates = rates.where("Year", are.between(start_year,
                                        end_year + 1))
```

```
            for state in proper_dates.group("State").column('State'):
                start_rate = proper_dates.where("State",
                    are.equal_to(state)).column("Murder Rate")
                end_rate = proper_dates.where("State",
                    are.equal_to(state)).column("Murder Rate")
                changes = two_year_changes(proper_dates.where("State", are.equal_
                observed_test_statistic = np.append(observed_test_statistic, char

            test_stat = abs(np.count_nonzero(observed_test_statistic > 0) - np.co

            print('Test statistic', start_year, 'to', end_year, ':', test_stat)
            num_states = proper_dates.group('State').num_rows


            samples = make_array()
            for i in np.arange(5000):
                samples = np.append(samples, simulate_under_null(num_states))

            return np.count_nonzero(samples>=test_stat)/5000

        run_test(death_penalty_murder_rates, 1971)
```

```
Test statistic 1971 to 1973 : 22


Out[229]: 0.001
```

```
In [230]: _ = tests.grade('q1_4_1')

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed
```

4.0.3 The rest of the states

We found a dramatic increase in murder rates for those states affected by the 1972 Supreme Court ruling, but what about the rest of the states? There were six states that had already outlawed execution at the time of the ruling.

Question 4.2. Create a table called `non_death_penalty_murder_rates` with the same columns as `murder_rates` but only containing rows for the six states without the death penalty in 1971. Perform the same test on this table. **Then**, in one sentence, conclude whether their murder rates were also more likely to increase from 1971 to 1973.

```
In [231]: non_death_penalty = death_penalty.where("Death Penalty", are.equal_to(Fal
          non_death_penalty_murder_rates = murder_rates.join('State', non_death_per
          run_test(non_death_penalty_murder_rates, 1971)

Test statistic 1971 to 1973 : 1


Out[231]: 0.6948
```

Murder rates are not more likely to increase from 1971 to 1973.

```
In [232]: _ = tests.grade('q1_4_2')

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed
```

4.0.4 The death penalty reinstated

In 1976, the Supreme Court repealed its ban on the death penalty in its rulings on a series of cases including Gregg v. Georgia, so the death penalty was reinstated where it was previously banned. This generated a second natural experiment. To the extent that the death penalty deters murder, reinstating it should decrease murder rates, just as banning it should increase them. Let's see what happened.

```
In [233]: print("Increases minus decreases from 1975 to 1977 (when the death penalt
                sum(death_penalty_murder_rates.where('Year', are.between_or_equal_t
                                              .group('State', two_year_changes)
                                              .column("Murder Rate two_year_changes
          run_test(death_penalty_murder_rates, 1975)

Increases minus decreases from 1975 to 1977 (when the death penalty was reinstated)
Test statistic 1975 to 1977 : 18


Out[233]: 0.0094
```

Hint: To check your results, figure out what your test statistic should be when there are 18 more decreases than increases, and verify that that's the test statistic that was printed. Also, you should have found a P-value near 0.01. If your P-value is very different, go back and inspect your `run_test` implementation and your test statistic to make sure that it correctly produces low P-values when there are many more decreases than increases.

Question 4.3. Now we've analyzed states where the death penalty went away and came back, as well as states where the death penalty was outlawed all along. What do you conclude from the results of the tests we have conducted so far? Does all the evidence consistently point toward one conclusion, or is there a contradiction?

Because when the death penalty was reinstated, there was a net sum decrease in the murder rates among death penalty states, we fail to accept the null that the increases or decreases were due to random chance. We then accept the alternative and can find the connection between instituting the death penalty and murder rates decreasing.

4.1 5. Visualization

While our analysis appears to support the conclusion that the death penalty deters murder, a 2006 Stanford Law Review paper argues the opposite: that historical murder rates do **not** provide evidence that the death penalty deters murderers.

To understand their argument, we will draw a picture. In fact, we've gone at this whole analysis rather backward; typically we should draw a picture first and ask precise statistical questions later!

What plot should we draw?

We know that we want to compare murder rates of states with and without the death penalty. We know we should focus on the period around the two natural experiments of 1972 and 1976, and we want to understand the evolution of murder rates over time for those groups of states. It might be useful to look at other time periods, so let's plot them all for good measure.

Question 5.1. Create a table called `average_murder_rates` with 1 row for each year in `murder_rates`. It should have 3 columns: * `Year`, the year, * `Death penalty states`, the average murder rate of the states that had the death penalty in 1971, and * `No death penalty states`, the average murder rate of the other states.

`average_murder_rates` should be sorted in increasing order by year. Its first three rows should look like:

Year	Death penalty states	No death penalty states
1960		
1961		
1962		

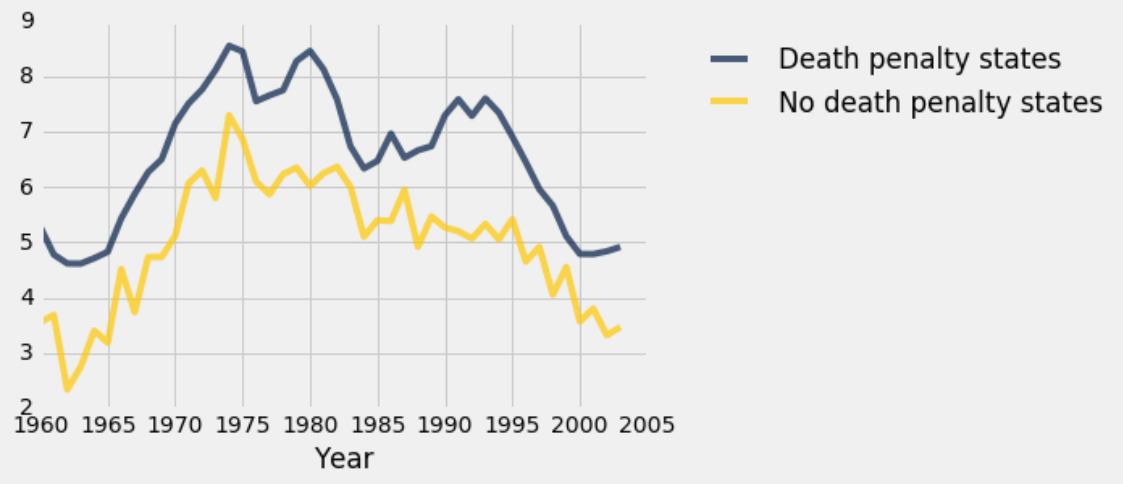
Hint: Use `pivot`. To compute average murder rates across states, just average the murder rates; you do not need to account for differences in population.

```
In [247]: # For reference, our solution used 5 method calls
          penalty_rates = death_penalty_murder_rates.group("Year", np.mean)
          non_penalty = non_death_penalty_murder_rates.group("Year", np.mean)
          full_average_murder_rates = penalty_rates.join("Year", non_penalty, "Yea
          average_murder_rates = full_average_murder_rates.drop(1,2,4,5).relabeled
                                 'Death penalty states').relabeled('Murder Rate me
          average_murder_rates

Out[247]: Year | Death penalty states | No death penalty states
          1960 | 5.27955              | 3.55
```

```
1961 | 4.77727              | 3.68333
1962 | 4.61591              | 2.33333
1963 | 4.61364              | 2.75
1964 | 4.71136              | 3.4
1965 | 4.82727              | 3.18333
1966 | 5.43182              | 4.51667
1967 | 5.875                | 3.73333
1968 | 6.27045              | 4.73333
1969 | 6.50227              | 4.73333
... (34 rows omitted)
```

Question 5.2. Describe in **one short sentence** a high-level takeaway from the line plot below. Are the murder rates in these two groups of states related?

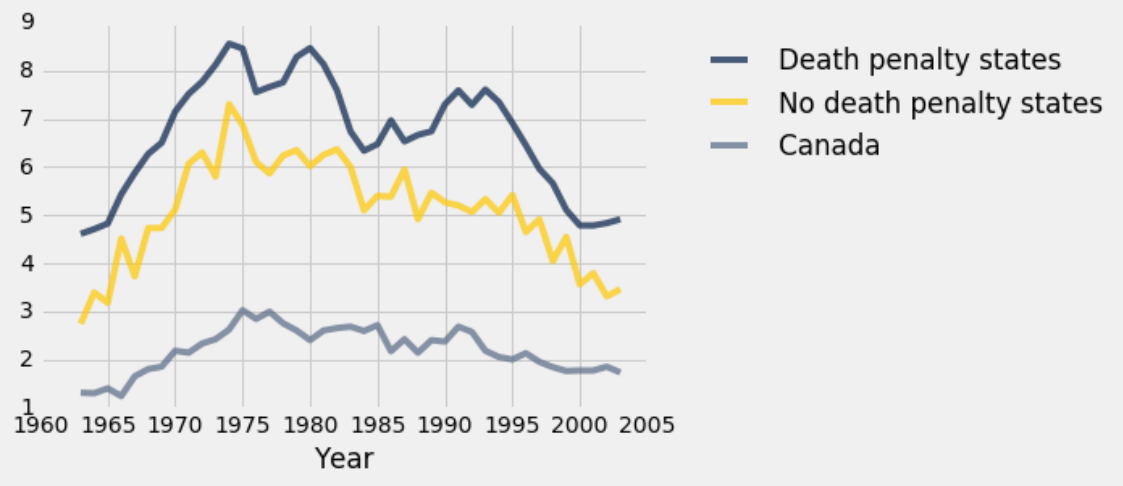
```
In [248]: average_murder_rates.plot('Year')
```



Death penalty states have always had a higher average murder rate than those states where no death penalty was enacted, and follow the same trend line regardless of the institution of the penalty or not.

Let's bring in another source of information: Canada.

```
In [249]: canada = Table.read_table('canada.csv')
          murder_rates_with_canada = average_murder_rates.join("Year", canada.selec
          murder_rates_with_canada.plot('Year')
```



The line plot we generated above is similar to a figure from the paper.

Canada has not executed a criminal since 1962. Since 1967, the only crime that can be punished by execution in Canada is the murder of on-duty law enforcement personnel. The paper states, "The most striking finding is that the homicide rate in Canada has moved in virtual lockstep with the rate in the United States."

Question 5.4. Complete their argument in 2-3 sentences; what features of these plots indicate that the death penalty is not an important factor in determining the murder rate? (If you're stuck, read the paper.)

Regardless of whether or not there is a death penalty instituted or not, the murder rates rise and fall at approxamately the same rate. This demonstrates that the change in murder rate is not dependent on the death penalty policy, but rather, external factors. It is also interesting to note that although the US and Canada follow different timelines of policy, they still follow the same trends.

Question 5.5. What assumption(s) did we make in Parts 1 through 4 of the project that led us to believe that the death penalty deterred murder, when in fact the line plots tell a different story?

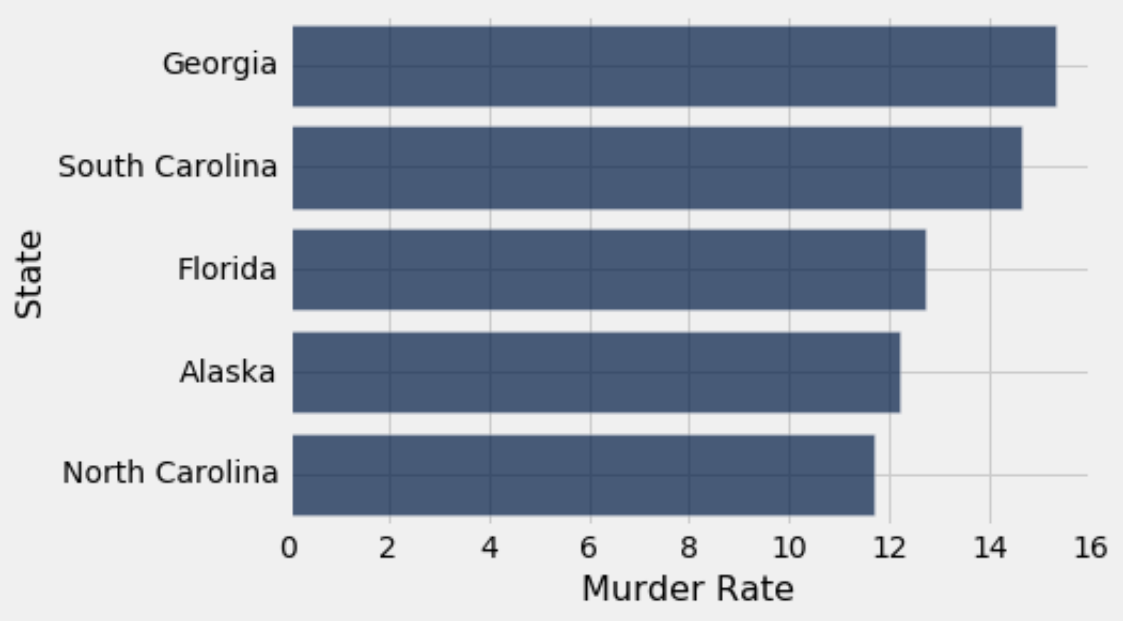
We previously assumed that murder rates rose and fell as a factor of whether or not the death penalty was instituted and failed to consider other confounding factors other than the policy implementation. Although we had a small p value and thus failed to accept the null, we assumed this was due to the death penalty policy and that predominately, when in reality, it was due to other confounding factors, as seen in the line plots.

You're done! Congratulations.

```
In [250]: # For your convenience, you can run this cell to run all the tests at onc
          import os
          print("Running all tests...")
          _ = [tests.grade(q[:-3]) for q in os.listdir("tests") if q.startswith('q'
          print("Finished running all tests.")

Running all tests...
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
```

```
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

----------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

----------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

----------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

----------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

----------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
```

```
[oooooooooook] 100.0% passed

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

----------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

----------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

----------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

----------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[oooooooooook] 100.0% passed

Finished running all tests.
```


Georgia
South Carolina
Florida
Alaska
North Carolina
State
0
2
4
6
8
10
12
14
16
Murder Rate

In [ ]: