

project3

April 27, 2016

1 Project 3 - Inference

Welcome to the third and final project in Data 8! 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

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 [91]: *# 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 plt
plt.style.use('fivethirtyeight')

import warnings
warnings.simplefilter(action="ignore", category=FutureWarning)

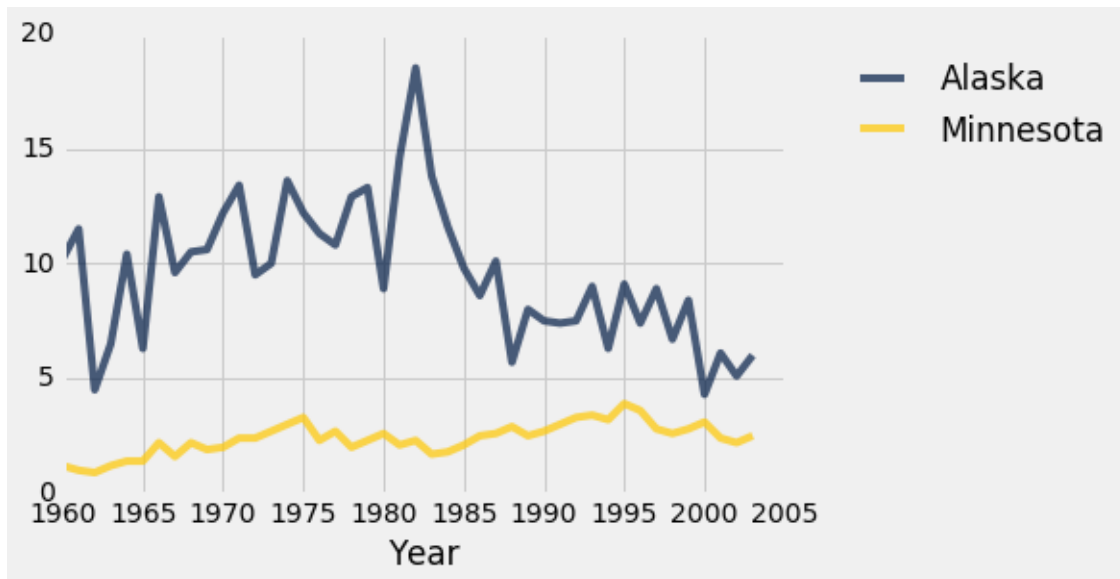
# These lines load the tests.
from client.api.assignment import load_assignment
project3 = load_assignment('project3.ok')
```

```
=====
Assignment: Project 3
OK, version v1.5.1
=====
```

2 1. Murder rates

Punishment for crime has many [philosophical justifications](#). An important one, of course, is that fear of punishment may deter people from committing crimes. Deterrence has been a focal point of the debate over


```
ak_mn = ak.join('Year', mn)
ak_mn.plot('Year')
```



```
In [94]: _ = project3.grade("q12")
```

~~~~~  
Running tests

-----  
Test summary

Passed: 1

Failed: 0

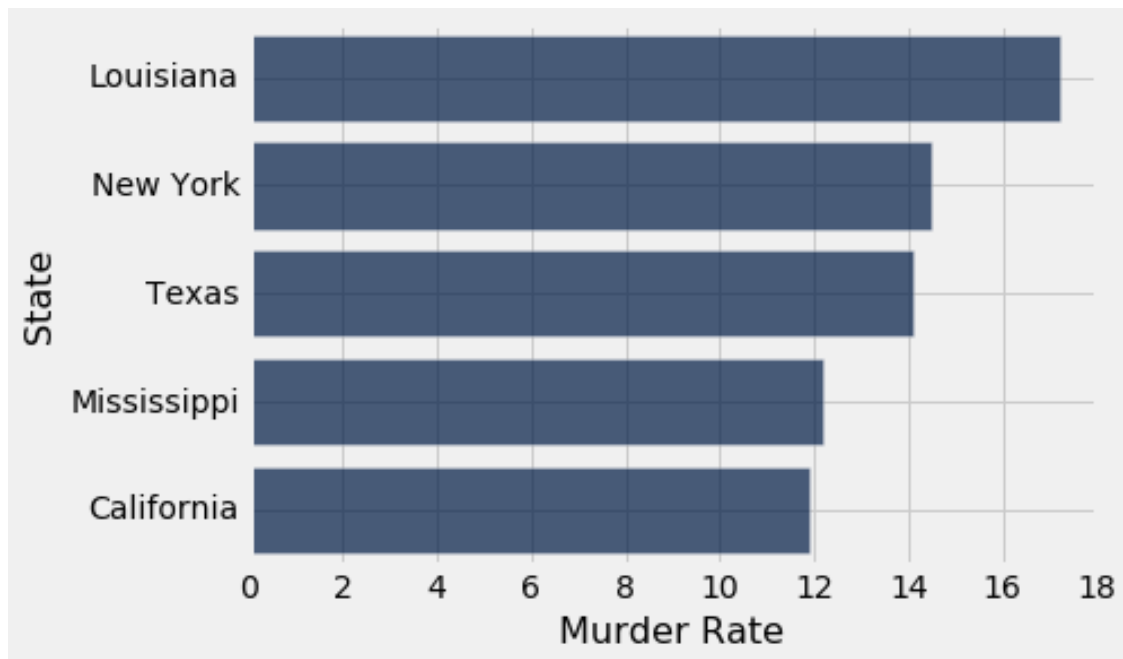
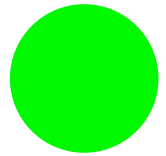
[ooooooooook] 100.0% passed

**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 a list of the names of these states in order of increasing murder rate.

```
In [95]: 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.sort(murder_rates.column('Murder Rate'),descending=True)\
        .where('Year', year).take(np.arange(5))
    most.barh('State', 'Murder Rate')
    return list(most.sort(most.column('Murder Rate')).column('State'))

most_murderous(1990)
```

```
Out[95]: ['California', 'Mississippi', 'Texas', 'New York', 'Louisiana']
```



```
In [96]: _ = project3.grade("q13")
```

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

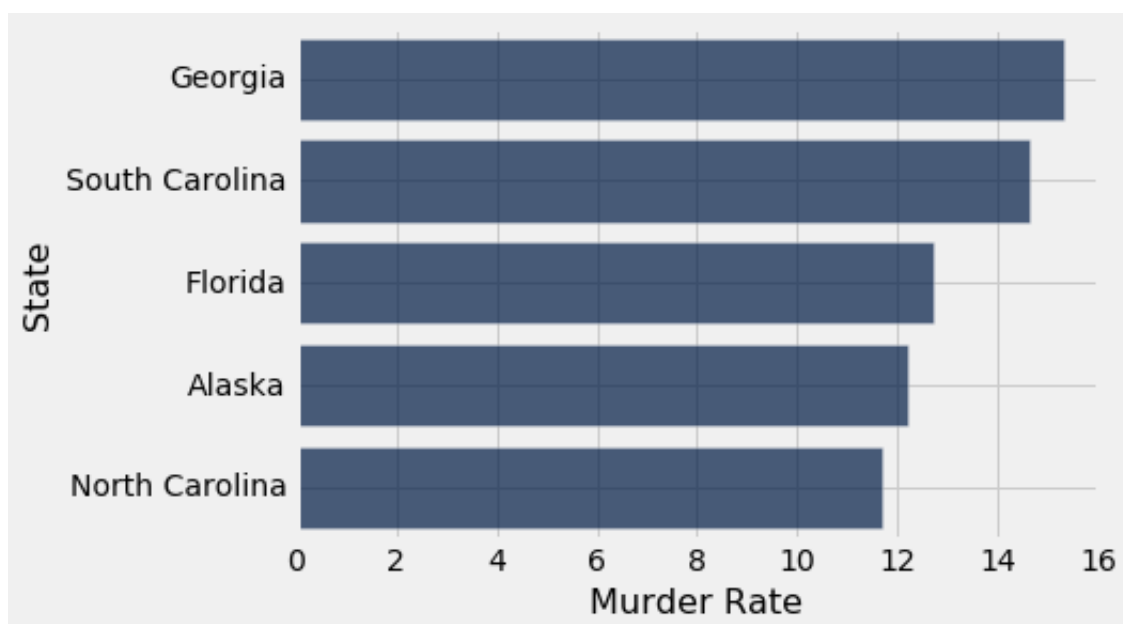
```

Test summary
```

```
 Passed: 1
```

```
 Failed: 0
```

```
[ooooooooook] 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.

```
In [97]: ca = murder_rates.where('State', 'California')
 ca_change = ca.where("Year", 1988).column(2).item(0) - ca.where('Year', 1975)\
 .column(2).item(0)

 np.round(ca_change)
```

```
Out[97]: 6983000
```

```
In [98]: _ = project3.grade("q14")
```

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

```
-----
Test summary
```

```
    Passed: 1
```

```
    Failed: 0
```

```
[ooooooooook] 100.0% passed
```

## 3 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, 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.

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

```
        diff_n([1, 10, 100, 1000, 10000], 2)
```

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

**Question 2.1.** Implement the function `two_year_changes` that takes a list 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 = [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 [100]: def two_year_changes(rates):
        "Return the number of increases minus the number of decreases after two years."
        return np.count_nonzero(diff_n(rates,2)>0)-np.count_nonzero(diff_n(rates,2)<0)

        print('Alaska:', two_year_changes(ak.column('Alaska')))
        print('Minnesota:', two_year_changes(mn.column('Minnesota')))
```

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

```
In [101]: _ = project3.grade("q21")
```

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

```

Test summary
```

```
Passed: 1
```

```
Failed: 0
```

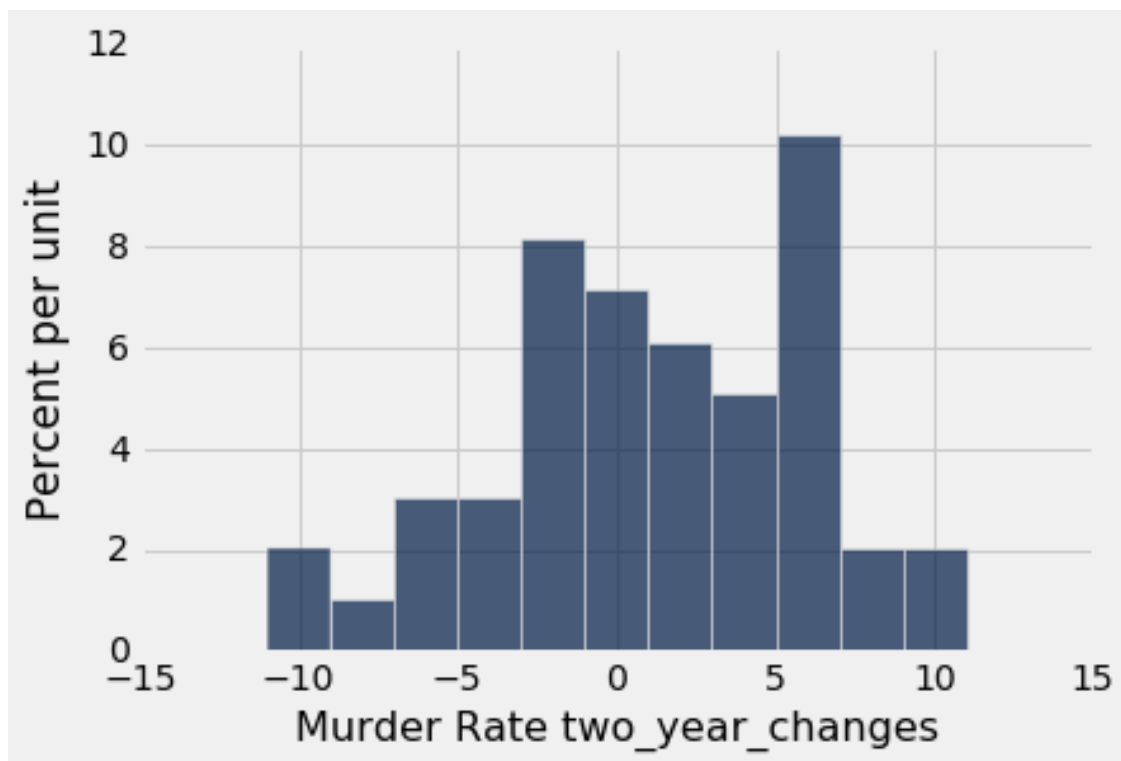
```
[ooooooooook] 100.0% passed
```

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 [102]: changes_by_state = murder_rates.group('State',two_year_changes).select([0,3])
changes_by_state.hist(1, bins=np.arange(-11, 12, 2))
```



```
In [103]: _ = project3.grade("q22")
```

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

```
-----  
Test summary
```

```
    Passed: 1
```

```
    Failed: 0
```

```
[ooooooooook] 100.0% passed
```

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 [104]: total_changes = sum(changes_by_state.column('Murder Rate two_year_changes'))  
         print('Total increases minus total decreases, across all states and years:', total_changes)
```

```
Total increases minus total decreases, across all states and years: 45
```

“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. 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.

```
In [105]: num_changes=len(murder_rates.column('Murder Rate'))-2*50  
         num_changes
```

```
Out[105]: 2100
```

```
In [106]: _ = project3.grade("q24")
```

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

```

Test summary
```

```
 Passed: 1
```

```
 Failed: 0
```

```
[ooooooooook] 100.0% passed
```

We now have enough information to perform a hypothesis test, using the difference between increases and decreases as a test statistic.

**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.

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.

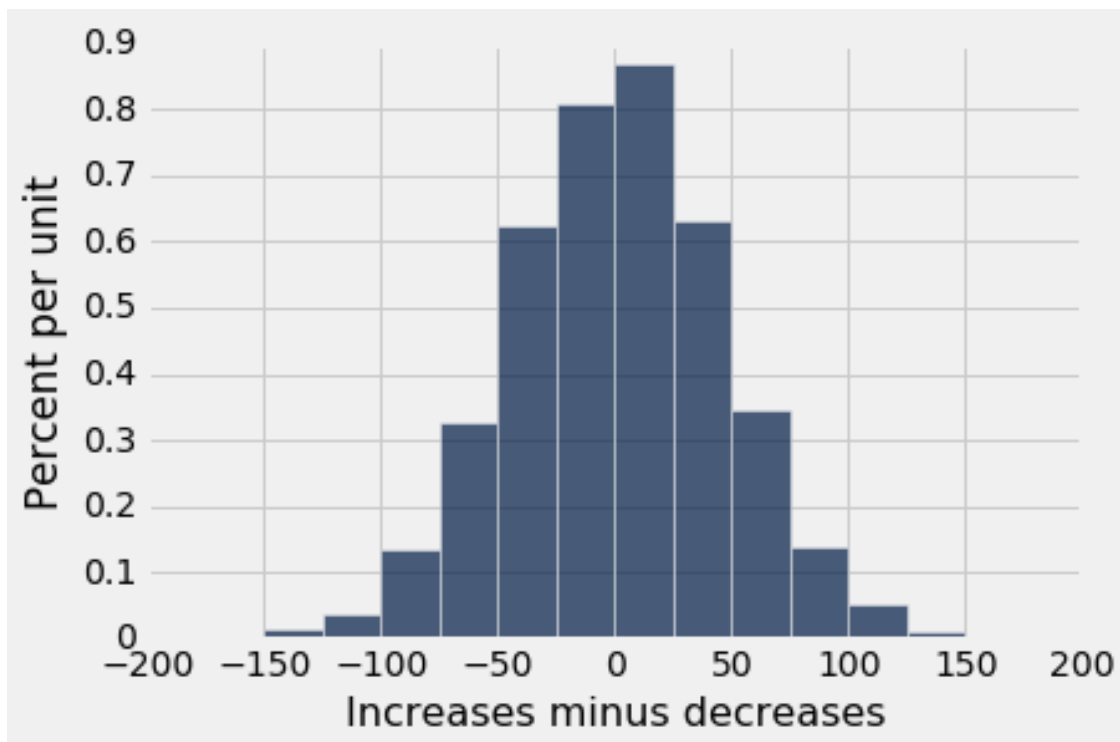
The cell below samples increases and decreases at random from a uniform distribution 100 times. The final column of the resulting table gives the number of increases and decreases that resulted from sampling in this way.

```
In [107]: uniform = Table(['Change', 'Chance']).with_rows(['Increase', 0.5], ['Decrease', 0.5])
 uniform.sample_from_distribution('Chance', 100)
```

```
Out[107]: Change | Chance | Chance sample
 Increase | 0.5 | 53
 Decrease | 0.5 | 47
```

**Question 2.5.** Complete the simulation below, which samples `num_changes` increases/decrease at random many times in order to form an empirical distribution of the difference between increases and decreases under the null hypothesis. The `uniform_samples` table should contain one row for each repetition that contains the difference between increases and decreases.

```
In [108]: uniform_samples = Table(['Increases minus decreases'])
 for i in np.arange(5000):
 sample = uniform.sample_from_distribution('Chance', num_changes)
 uniform_samples.append([sample.column('Chance sample').item(0)-sample.column\
 ('Chance sample').item(1)])
 uniform_samples.hist(0, bins=np.arange(-200, 201, 25))
```



```
In [109]: _ = project3.grade("q25")
```



~~~~~  
Running tests

-----  
Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

**Question 2.6.** Looking at this histogram, draw a conclusion about whether murder rates systematically increase more than they decrease, or whether the difference is consistent with the null hypothesis. (You do not need to compute a P-value for this question.)

The murder rates are roughly equal on either side which means the difference is consistent with the null hypothesis.

## 4 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: We are studying the people in each state in the United States.
- Control Group: The control group is the states do not have the death penalty.
- Treatment Group: The treatment group is the states that have enforced the death penalty.
- Outcome: The outcome will tell us if enforcing the death penalty affect the murder rate.

**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?

It is not sufficient to compare murder rates in places and times when the death penalty was in forced with places and times when it wasn't because that would only show an association, not a causation.

### 4.0.1 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 [110]: def had_death_penalty_in_1971(state):
 return state not in ['Alaska', 'Hawaii', 'Maine', 'Michigan', 'Wisconsin', 'Minnesota']

states = murder_rates.group('State').select('State')
death_penalty = states.with_column('Death Penalty', states.apply(had_death_penalty_in_1971, 0))
death_penalty
```

```
Out[110]: 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 [111]: num_death_penalty_states = death_penalty.where(1, True).num_rows
num_death_penalty_states
```

```
Out[111]: 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.

```
In [112]: death_penalty_murder_rates = murder_rates.join("State", death_penalty.\
 where('Death Penalty', True)).drop(4)

death_penalty_murder_rates
```

```
Out[112]: State | Year | Population | Murder Rate
Alabama | 1960 | 3266740 | 12.4
Alabama | 1961 | 3302000 | 12.9
Alabama | 1962 | 3358000 | 9.4
Alabama | 1963 | 3347000 | 10.2
Alabama | 1964 | 3407000 | 9.3
Alabama | 1965 | 3462000 | 11.4
Alabama | 1966 | 3517000 | 10.9
Alabama | 1967 | 3540000 | 11.7
Alabama | 1968 | 3566000 | 11.8
Alabama | 1969 | 3531000 | 13.7
... (1926 rows omitted)
```

```
In [113]: _ = project3.grade("q33")
```

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

```
-----  
Test summary
```

```
    Passed: 1
```

```
    Failed: 0
```

```
[ooooooooook] 100.0% passed
```

The null hypothesis doesn't specify how the murder rate changes; it only talks about increasing or decreasing. So, again we will use a test statistic that measures the difference between the number of increases and decreases to evaluate the null hypothesis.

**Test statistic:** For a collection of states and two years, compute the number of states where the murder rate increased, then subtract the number of states where the murder rate decreased.

A hypothesis test with this kind of statistic is called a sign test, since it only looks at the sign of a change. If the test statistic is very large (or very small), that means many more (or many fewer) states saw increases in murder rates than decreases. Values near 0 are consistent with the null hypothesis, and large positive or negative values are inconsistent.

**Question 3.4.** Complete the function called `sign_test_statistic` by implementing `increases_minus_decreases`. The `sign_test_statistic` function takes 3 arguments: \* `table`, a table, \* `before`, the label of a column of numbers in that table containing rates at the start of the two-year period, and \* `after`, the label of another column of numbers in that table containing rates at the end of the two-year period.

It should return the number of rows in which the `after` column has a higher value than the `before` column, minus the number of rows in which the `before` column has a higher value than the `after` column. In other words, it computes our test statistic.

```
In [114]: # This function is written for you, except that it relies on the  
# function increases_minus_decreases, which we have left for you  
# to finish.
```

```
def sign_test_statistic(table, before, after):  
    """The number of increases minus the number of decreases, for all rows in table."""  
    before_column = table.column(before)  
    after_column = table.column(after)  
    return increases_minus_decreases(before_column, after_column)  
  
def increases_minus_decreases(before_column, after_column):  
    """The number of increases minus the number of decreases between two parallel arrays."""  
    assert len(before_column) == len(after_column) \\  
    == len(after_column), 'Bad arguments'  
    # Fill in this function starting here.  
    return np.count_nonzero((after_column-before_column)>0) - \\  
    np.count_nonzero((after_column-before_column)<0)
```

```
In [115]: _ = project3.grade("q34")
```

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

```

Test summary
```

```
 Passed: 1
```

```
Failed: 0
[ooooooooook] 100.0% passed
```

**Question 3.5.** Assign `changes_72` to the value of our test statistic for the years 1971 to 1973 and the states in `death_penalty_murder_rates`.

```
In [116]: compare_72 = Table().with_columns([
 ["1971", death_penalty_murder_rates.where("Year", 1971).column(3)],
 ["1973", death_penalty_murder_rates.where("Year", 1973).column(3)]
])
```

```
changes_72 = sign_test_statistic(compare_72, '1971', '1973')
print('Increases minus decreases from 1971 to 1973:', changes_72)
```

```
Increases minus decreases from 1971 to 1973: 22
```

```
In [117]: _ = project3.grade("q35")
```

```
~~~~~
```

```
Running tests
```

```
-----
```

```
Test summary
```

```
Passed: 1
```

```
Failed: 0
```

```
[ooooooooook] 100.0% passed
```

**Question 3.6.** Write a function called `simulate_statistic_under_null`. It should take as its argument a sample size `n` (such as the number of death penalty states in 1971). It should simulate one sample of `n` states' murder rate changes under the null hypothesis and return the number of increases minus the number of decreases in that sample. In other words, it should simulate one value of `sign_test_statistic` that we'd see if the null hypothesis were true. That value should be an integer between `-n` and `n` (inclusive).

```
In [118]: def simulate_statistic_under_null(n):
            sample = uniform.sample_from_distribution('Chance', n)
            return sample.column(2).item(0) - sample.column(2).item(1)
```

```
simulate_statistic_under_null(10)
```

```
Out[118]: 0
```

```
In [119]: _ = project3.grade("q36")
```

```
~~~~~
```

```
Running tests
```

```

```

```
Test summary
```

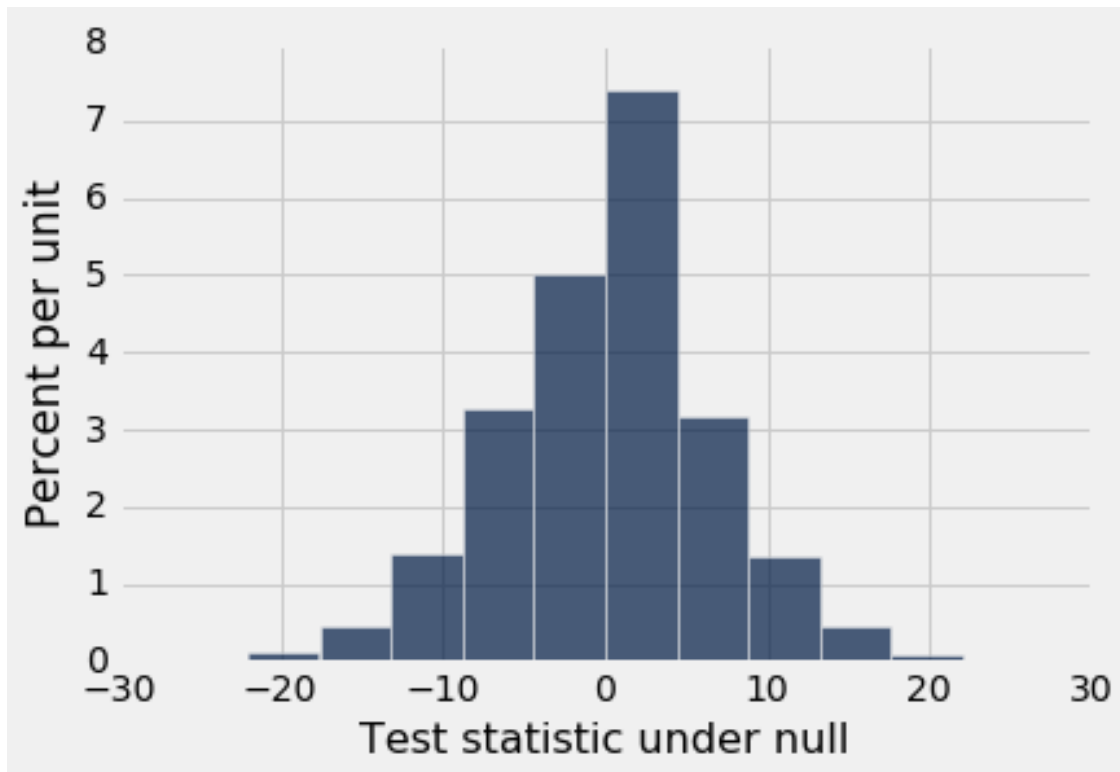
```
Passed: 1
```

```
Failed: 0
```

```
[ooooooooook] 100.0% passed
```

**Question 3.7.:** Draw an empirical histogram of the statistic under the null hypothesis by simulating the test statistic 5000 times.

```
In [120]: samples = Table(['Test statistic under null'])
 for i in np.arange(5000):
 samples.append([simulate_statistic_under_null(num_death_penalty_states)])
 samples.hist(0)
```



```
In [121]: _ = project3.grade("q37")
```

```
~~~~~
```

```
Running tests
```

```
-----
```

```
Test summary
```

```
    Passed: 1
```

```
    Failed: 0
```

```
[ooooooooook] 100.0% passed
```

#### 4.0.2 Conclusion

To complete our analysis, we can compare this empirical distribution to the observed test statistic for the two-year period from 1971 to 1973.

The P-value of the observation is the proportion of simulated test statistics that are at least as extreme as the observed value. Either a very positive or a very negative observation is inconsistent with this empirical distribution. In this case, we include in the P-value proportion all simulated statistics that are at least as large as `changes_72` or at least as small as `-changes_72` (that's negative one times `changes_72`). Including both positive and negative extremes in the P-value proportion is called a two-tailed test.

**Question 3.8.** Assign `sign_72_p_value` to the two-tailed P-value of the observed `changes_72` under the null hypothesis.

```
In [122]: sign_72_p_value = np.count_nonzero(samples.column(0)>= changes_72)/samples.num_rows +\
        np.count_nonzero(samples.column(0)<= -changes_72)/samples.num_rows
        print('1971 - 1973 P-value:', sign_72_p_value)
```

1971 - 1973 P-value: 0.0012000000000000001

**Question 3.9.** Draw an inferential conclusion about the changes in murder rates from 1971 to 1973 in states where the death penalty was originally in force.

The changes in murder rates from 1971 to 1973 in states where the death penalty was originally in force is inconsistent (p-value is low) therefore we reject the null hypothesis.

## 5 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 sign tests to strengthen our claim.

Conducting a sign 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 difference between increases and decreases,
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 two-tailed P-value.

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

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

- A table of murder `rates` for certain states, and
- A `before` year and an `after` year.

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

Hint: You can use functions you already defined, such as `increases_minus_decreases` and `simulate_statistic_under_null`.

```
In [123]: def sign_test(rates, before, after):
        """Return a two-tailed P-value for the observed difference between increases and decreases.
        table=Table().with_columns([
            ['before', rates.where(rates.column("Year")==before).column('Murder Rate')],
            ['after', rates.where(rates.column("Year")==after).column('Murder Rate')]
        ])
        changes = sign_test_statistic(table, 'before', 'after')
        print('Increases minus decreases', before, 'to', after, ':', changes)
        num_states = rates.group('State').num_rows

        samples = Table(['Test statistic under null'])
        for i in np.arange(5000):
            samples.append([simulate_statistic_under_null(num_states)])
        return np.count_nonzero(samples.column(0)>= np.abs(changes))/samples.num_rows\
            + np.count_nonzero(samples.column(0)<= -np.abs(changes))/samples.num_rows

        sign_test(death_penalty_murder_rates, 1971, 1973)
```

Increases minus decreases 1971 to 1973 : 22

```
Out[123]: 0.001
```

```
In [124]: _ = project3.grade("q41")
```

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

```

Test summary
```

```
 Passed: 1
```

```
 Failed: 0
```

```
[ooooooooook] 100.0% passed
```

### 5.0.3 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 sign test on this table. In one sentence, conclude whether their murder rates also increased from 1971 to 1973.

```
In [125]: non_death_penalty_murder_rates = murder_rates.\
 join("State", death_penalty.where('Death Penalty', False)).drop(4)
 sign_test(non_death_penalty_murder_rates, 1971, 1973)
```

```
Increases minus decreases 1971 to 1973 : 1
```

```
Out[125]: 0.6986
```

The null hypothesis is supported because the p-value is high.

```
In [126]: _ = project3.grade("q42")
```

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

```
-----
Test summary
```

```
    Passed: 1
```

```
    Failed: 0
```

```
[ooooooooook] 100.0% passed
```

### 5.0.4 The death penalty reinstated

In 1976, the Supreme Court repealed its ban on the death penalty in its rulings for [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 [127]: sign_test(death_penalty_murder_rates, 1975, 1977)
```

```
Increases minus decreases 1975 to 1977 : -18
```

```
Out[127]: 0.009600000000000001
```

Hint: You should see above an observed statistic of -18; there were 18 more decreases than increases among the 44 states that had the death penalties in force. You should also have found a P-value near 0.01. If your P-value is very different, go back and inspect your `sign_test` implementation to make sure that it correctly handles negative observed test statistics.

**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? Is all the evidence consistent, or is there a contradiction?

States that had already outlawed execution at the time of the ruling were consistent with the null hypothesis that the murder rate was based on chance. However, when looking at the states where the death penalty went away and came back, we see that the murder rate was not based on chance. The evidence is not consistent and the change in murder rate is not due to chance.

## 5.1 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 [128]: oui = murder_rates.join("State",death_penalty).where("Death Penalty", True)\
        .group("Year", np.mean)
        non= murder_rates.join("State",death_penalty).where("Death Penalty",False)\
        .group("Year", np.mean)
        average_murder_rates=Table().with_columns([
            "Year", oui.column(0),
            "Death penalty states", oui.column(3),
            "No death penalty states", non.column(3),
        ])
        average_murder_rates
```

```
Out[128]: 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)

```

```
In [129]: _ = project3.grade("q51")
```

~~~~~  
Running tests

Test summary

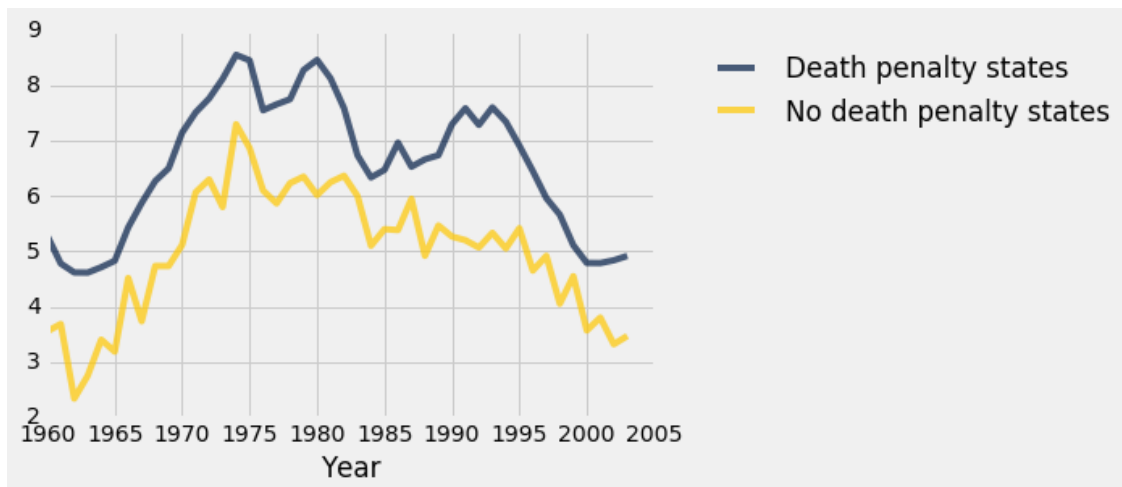
Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

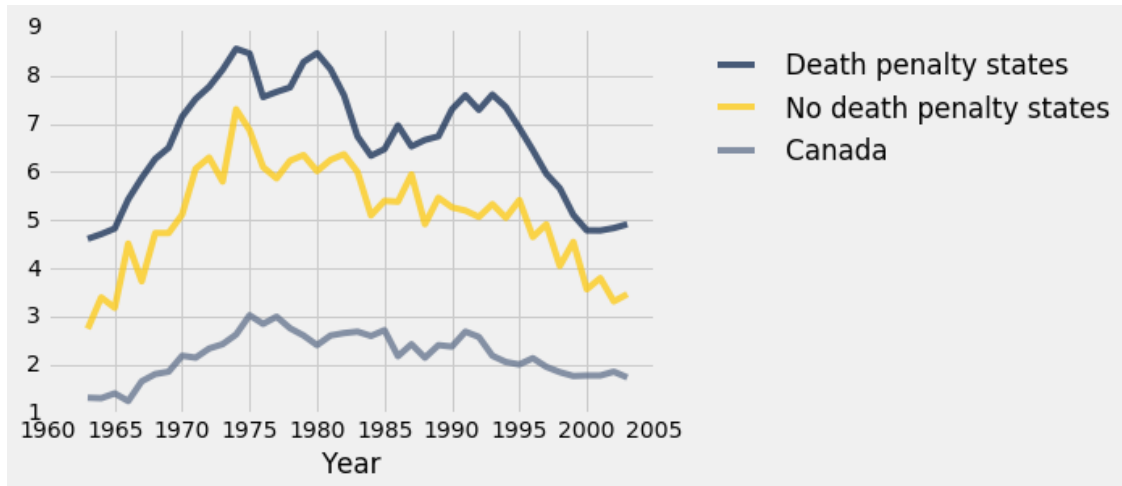
Question 5.2. Describe in **one short sentence** a high-level takeaway from the line plot below. Are the murder rates in these two groups of states related?

```
In [130]: average_murder_rates.plot('Year')
```



The murder rates in these two groups of states seem to be related.
Let's bring in another source of information: Canada.

```
In [131]: canada = Table.read_table('canada.csv')
murder_rates_with_canada = average_murder_rates.join\
("Year", canada.select(["Year", "Homicide"]).relabel("Homicide", "Canada"))
murder_rates_with_canada.plot('Year')
```



The line plot we generated above is similar to a figure from the [paper](#).

Paper plot

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](#).)

Although the homicide rate in Canada seems to fluctuate similarly to the rate in the US, it is significantly lower than the rate in the US. They haven’t executed a criminal since 1962 and their murder rate has stayed relatively constant. The plots for the US State with and the US states without the death penalty fluctuate in the same way which means that the death penalty does not have much effect on the murder rate

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 assumed that all other factors remained constant and did not affect the murder rates throughout the years such as external factors like the socioeconomic differences of populations in different years, which might change the murder rates regardless of the death penalty.

You’re done! Congratulations.

```
In [132]: # For your convenience, you can run this cell to run all the tests at once!
import os
print("Running all tests...")
_ = [project3.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

[oooooooook] 100.0% passed

Running tests

Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~  
Running tests

-----  
Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~  
Running tests

Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~  
Running tests

-----  
Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~  
Running tests

Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~  
Running tests

-----  
Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~

Running tests

Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~  
Running tests

-----  
Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~  
Running tests

Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~  
Running tests

-----  
Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~  
Running tests

Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~  
Running tests

-----  
Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~

Running tests

Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

~~~~~  
Running tests

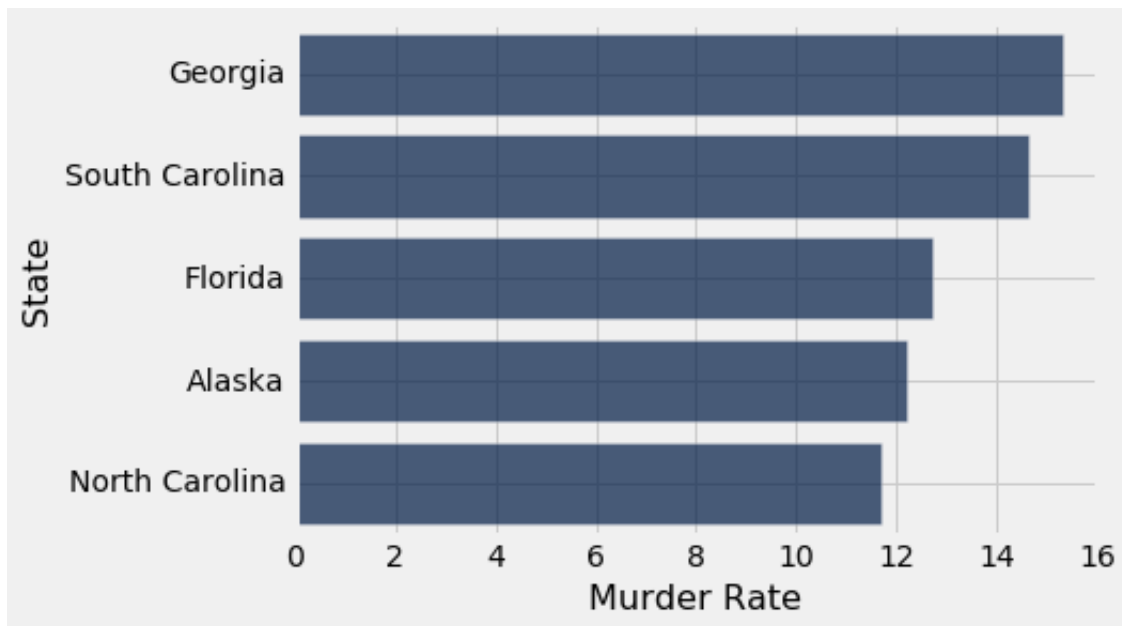
-----  
Test summary

Passed: 1

Failed: 0

[ooooooooook] 100.0% passed

Finished running all tests.



In [ ]: