lab09

April 26, 2022

Name: Allan Gongora

Section: 0131

1 Lab 9: Inference and Capital Punishment, Part 2

Welcome to Lab 9! This week, we will continue to investigate the relationship between murder rates and capital punishment in the United States. Please complete Lab 8 before starting Lab 9.

By the end of this lab, 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.

All of the concepts necessary for this project are found in the Inferential Thinking textbook. If you are stuck on a particular problem, reading through the relevant textbook section often will help clarify the concept.

Please complete this notebook by filling in the cells provided. Before you begin, execute the following cell to load the provided tests.

[2]:
```
pip install gofer-grader
```

```
Collecting gofer-grader
  Using cached gofer_grader-1.1.0-py3-none-any.whl (9.9 kB)
Requirement already satisfied: pygments in /opt/conda/lib/python3.7/site-
packages (from gofer-grader) (2.11.2)
Requirement already satisfied: tornado in /opt/conda/lib/python3.7/site-packages
(from gofer-grader) (6.1)
Requirement already satisfied: jinja2 in /opt/conda/lib/python3.7/site-packages
(from gofer-grader) (3.0.3)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.7/site-
packages (from jinja2->gofer-grader) (2.0.1)
Installing collected packages: gofer-grader
Successfully installed gofer-grader-1.1.0
Note: you may need to restart the kernel to use updated packages.
```

[3]:

```
from datascience import *
import numpy as np

%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')

from gofer.ok import check
```

Recommended Reading: * Visualization * Sampling and Empirical Distributions * Testing Hypotheses

1) For all problems that you must write explanations and sentences for, you **must** provide your answer in the designated space. This can include:
    A) Sentence reponses to questions that ask for an explanation
    B) Numeric responses to multiple choice questions
    C) Programming code
2) Moreover, throughout this lab and all future ones, please be sure to not re-assign variables throughout the notebook! For example, if you use `max_temperature` in your answer to one question, do not reassign it later on. Otherwise, you will fail tests that you thought you were passing previously!

Once you're finished, select "Save and Checkpoint" in the File menu. Your name and course section number should be in the first and last cell of the assignment. Be sure you have run all cells with code and that the output from that is showing. Then click "Print Preview" in the File menu. Print a copy from there in pdf format. (This means you right click and choose print and choose "save as pdf" from your printer options.) You will need to submit the pdf in Canvas by the deadline.

The gopher grader output and/or output from your coding are essential to helping your instructor grade your work correctly and in a timely manner.

Files submitted that are missing the required output will lose some to all points so double check your pdf before submitting.

1.1 Overview: Recap

In the United States, some jurisdictions execute some people who are convicted of particularly serious crimes, such as 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:

> Through our exploration, 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.

In Lab 8, we completed the following tasks:

1. In Section 1, we explored the data.
2. In Section 2, we tested a null hypothesis that murder rates were equally likely to go up or down each year, as if by a flip of a fair coin.

If you do not remember Lab 8, we highly recommend you go back and review it now. It will help you for this lab.

In Lab 9, we will complete the following tasks:

1. In Section 1, we'll apply a similar test to see whether US states that suddenly ended or reinstituted the death penalty were more likely to see murder rates increase than decrease.
2. In Section 2, we'll run some more tests to support claims that we had been investigating in previous sections.
3. In Section 3, we'll try to answer our question about deterrence using a visualization rather than a formal hypothesis test.

1.1.1 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 the 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.)

[4]:
```
murder_rates = Table.read_table('crime_rates.csv').select('State', 'Year',␣
 ↪'Population', 'Murder Rate')
murder_rates.set_format("Population", NumberFormatter)

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

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)
```

As a reminder of our hypothesis test, here are the null hypothesis, alternative hypothesis, and test statistic from Lab 8.

> **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.

> **Alternative Hypothesis**: State murder rates are more likely to increase over two-year periods.
>
> **Test Statistic**: The number of increases minus the number of decreases.

1.2 1. 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 presence of the death penalty affects the murder rates of a state.

Question 1.1 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? Be precise!

Write your answers below.

- Population: All 50 states in the US
- Control Group: States without the death penalty
- Treatment Group: States that have the death penalty
- Outcome: Whether or not the murder rates were affected by the death penalty

Question 1.2 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?

Not actualyl sure. I know this is a problem not why. Been trying to write up an answer for 20 minutes and i didnt like any of them

Your answers for Questions 1.1 and 1.2 are not graded. To check your answers, go onto the edX discussion forum for Lab 9 and check your answers with other students!

1.2.1 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 have some hope 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.

Our alternative hypothesis involves our suspicion that murder rates increase when the death penalty is eliminated.

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.

[5]:

```
non_death_penalty_states = make_array('Alaska', 'Hawaii', 'Maine', 'Michigan',␣
 ↪'Wisconsin', 'Minnesota')
def had_death_penalty_in_1971(state):
    """Returns True if the argument is the name of a state that had the death␣
 ↪penalty in 1971."""
    # 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

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

[5]:

```
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)
```

[6]:

```
num_death_penalty_states = death_penalty.where("Death Penalty", are.
 ↪equal_to(True)).num_rows
num_death_penalty_states
```

```
[6]: 44
```

```
[7]: murder_rates
```

```
[7]: 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)
```

Question 1.3 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	3,266,740	12.4
Alabama	1961	3,302,000	12.9

```
[8]: death_penalty_murder_rates = murder_rates.join("State", death_penalty).
       ↪where("Death Penalty", True).drop("Death Penalty")
     death_penalty_murder_rates
```

```
[8]: 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)
```

```
[9]: check("tests/q1_3.py")
```

```
[9]: <gofer.ok.OKTestsResult at 0x7f3a2feb3910>
```

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 we defined in Lab 8.

Question 1.4 Assign `test_stat_72` to the value of the test statistic for the years 1971 to 1973 using the states in `death_penalty_murder_rates`. As before, the test statistic is, "the number of increases minus the number of decreases."

Hint: It may be helpful to use multiple lines to work through each part of your solution. One way to proceed is to `group` using your `two_year_changes` function on a table that contains only the range of years you care about.

[12]:
```
def two_year_changes(rates):
    "Return the number of increases minus the number of decreases after two␣
 ↪years."
    changes = diff_n(rates, 2)
    return len(changes[changes > 0]) - len(changes[changes < 0])
```

[23]:
```
temp = death_penalty_murder_rates.where("Year", are.between_or_equal_to(1971,␣
 ↪1973)).group("State", collect = two_year_changes)
test_stat_72 = temp["Murder Rate two_year_changes"].sum()
print('Test statistic from 1971 to 1973:', test_stat_72)
```

```
Test statistic from 1971 to 1973: 22
```

[24]:
```
check("tests/q1_4.py")
```

[24]:
```
<gofer.ok.OKTestsResult at 0x7f3a30bb3050>
```

Question 1.5 Draw an empirical histogram of the statistic under the null hypothesis by simulating the test statistic 10,000 times.

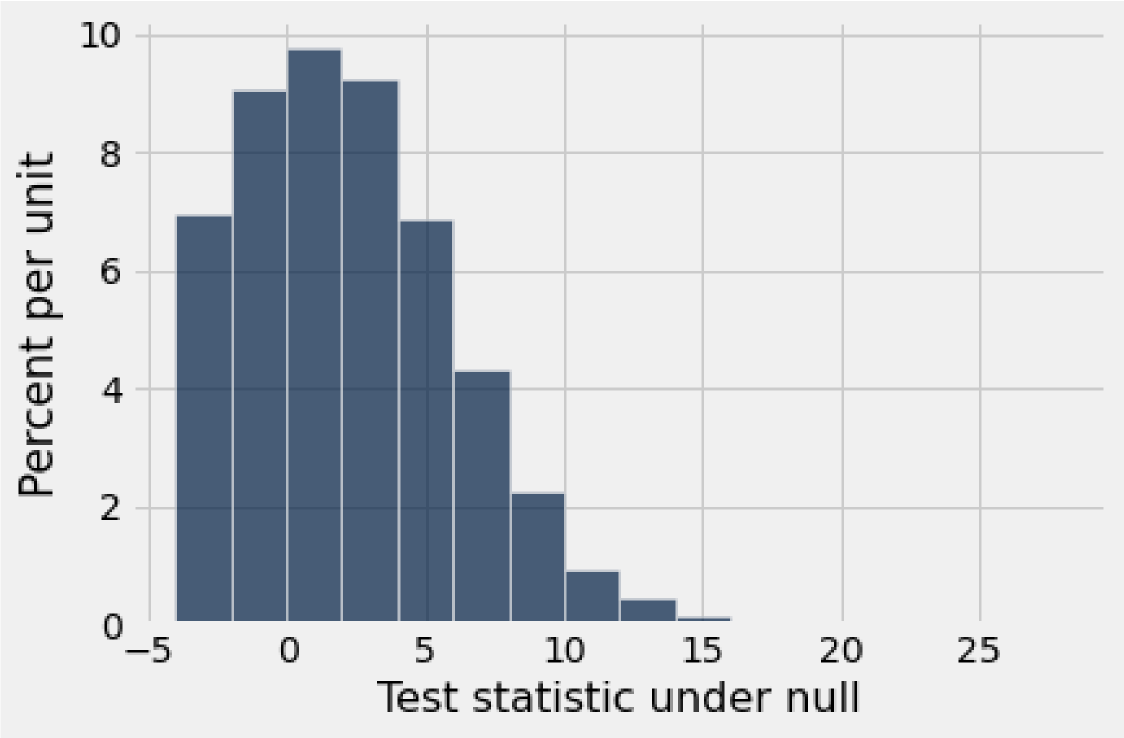
Hint: You may find the `simulate_under_null` function from Lab 8 useful.

[25]:
```
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.
    """
    uniform = Table().with_columns(
        "Change", make_array('Increase', 'Decrease'),
        "Chance", make_array(0.5,        0.5))
    sample = uniform.sample_from_distribution('Chance', num_chances_to_change)
    return sample.column(2).item(0) - sample.column(2).item(1)
```

[29]:
```
samples = []
for i in np.arange(10000):
    samples.append(simulate_under_null(test_stat_72))
```

```
samples = np.array(samples)
```

[27]:
```
# Run this cell to draw an empirical histogram of the
# statistic under the null hypothesis.
Table().with_column('Test statistic under null', samples).hist(bins=np.
 ↪arange(-4, 28+2, 2))
```



1.2.2 Conclusion

Question 1.6 Complete the analysis as follows: 1. Compute a P-value. 2. Using a 5% P-value cutoff, draw a conclusion about the null and alternative hypotheses. 3. Describe your findings using simple, non-technical language. What does your analysis tell you about murder rates after the death penalty was suspended? What can you claim about causation from your statistical analysis?

P-value: 18

Conclusion about the hypotheses: Reject Null

Findings: The murder rates increased when death penalty was eliminated. They increased so much that we can say it was not due to chance.

[32]:
```
samples.max() * 0.95 <= test_stat_72 <= samples.max() * 1.05, f"P-value is␣
 ↪{samples.max()}"
```

```
[32]: (False, 'P-value is 18')
```

1.3 2. 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 requires 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`. It takes the following arguments:

- a table of murder `rates` for certain states, sorted by state and year like `murder_rates`
- 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.

`run_test` works regardless of the direction of association in your alternative hypothesis by accounting for whether `observed_test_statistic` is negative or positive.

Run the cell below to define `run_test`. Make sure you understand what it's doing and why.

```
[33]: def run_test(rates, start_year):
          """Prints out the observed test statistic and returns a P-value for this␣
      ↪statistic under the null hypothesis
          Parameters
          ----------
          rates : Table
              A table of murder rates for certain states, sorted by state and year␣
      ↪like murder_rates
          start_year : int
              The year when the analysis starts
          """
          end_year = start_year + 2
          observed_test_statistic = sum(rates.where('Year', are.
      ↪between_or_equal_to(start_year, end_year)).select('State', 'Murder Rate').
      ↪group('State', two_year_changes).column(1))
          print('Test statistic', start_year, 'to', end_year, ':',␣
      ↪observed_test_statistic)
```

```
    num_states = rates.group('State').num_rows

    samples = []
    for i in np.arange(5000):
        samples.append(simulate_under_null(num_states))
    samples = np.array(samples) # Convert from a list to an array

    if observed_test_statistic > 0:
        return sum(samples >= observed_test_statistic) / len(samples)
    elif observed_test_statistic < 0:
        return sum(samples <= observed_test_statistic) / len(samples)

run_test(death_penalty_murder_rates, 1971)
```

```
Test statistic 1971 to 1973 : 22
```

```
[33]: 0.0004
```

1.3.1 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 2.1 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**, set `reject_null_2` to whether their murder rates were also more likely to increase from 1971 to 1973.

Hint: Feel free to use additional lines for your solution.

```
[34]: non_death_penalty_murder_rates = murder_rates.where("State", are.
        ↪contained_in(non_death_penalty_states))
      non_death_penalty_murder_rates.show(10)
      run_test(non_death_penalty_murder_rates, 1971)
```

```
<IPython.core.display.HTML object>

Test statistic 1971 to 1973 : 1
```

```
[34]: 0.3528
```

```
[37]: reject_null = False
```

```
[38]: check("tests/q2_1.py")
```

```
[38]: <gofer.ok.OKTestsResult at 0x7f3a2e362710>
```

1.3.2 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.

[ ]:

```
print("Increases minus decreases from 1975 to 1977 (when the death penalty was␣
 ↪reinstated) among death penalty states:",
      sum(death_penalty_murder_rates.where('Year', are.
 ↪between_or_equal_to(1975, 1977))
                                      .group('State', two_year_changes)
                                      .column("Murder Rate two_year_changes")))
run_test(death_penalty_murder_rates, 1975)
```

Question 2.2 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?

1) Our results point toward the conclusion that the death penalty moratorium increased murder rates.
2) Our results point toward the conclusion that the death penalty moratorium increased murder rates, but we have not accounted for differences over time as a confounding factor.
3) Our results don't allow us to make any conclusion about murder rates and death penalties.
4) Our results point toward the conclusion that the death penalty moratorium didn't influence murder rates.
5) None of these conclusions are valid, or multiple of these conclusions are valid

Below, set `we_conclude` to a single number, corresponding to your answer.

[45]:

```
we_conclude = 2
```

[46]:

```
check("tests/q2_2.py")
```

[46]:

```
<gofer.ok.OKTestsResult at 0x7f3a2bff1ad0>
```

1.4 3. 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 3.1 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 in the states that had the death penalty in 1971, and * `No death penalty states`, the average murder rate in 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 1: It's easiest to use `pivot`. We don't expect you to account for differences in population.

Hint 2: You can reorder columns in a table using `.select`.

[57]:

```
# For reference, our solution used 5 method calls
average_murder_rates = Table().with_columns(
    "Year", range(death_penalty_murder_rates["Year"].min(),␣
 ↪death_penalty_murder_rates["Year"].max() + 1),
    "Death penalty states", death_penalty_murder_rates.group("Year", collect=np.
 ↪average)["Murder Rate average"],
    "No death penalty states", non_death_penalty_murder_rates.group("Year",␣
 ↪collect=np.average)["Murder Rate average"],
)

average_murder_rates
```

[57]:

```
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)
```

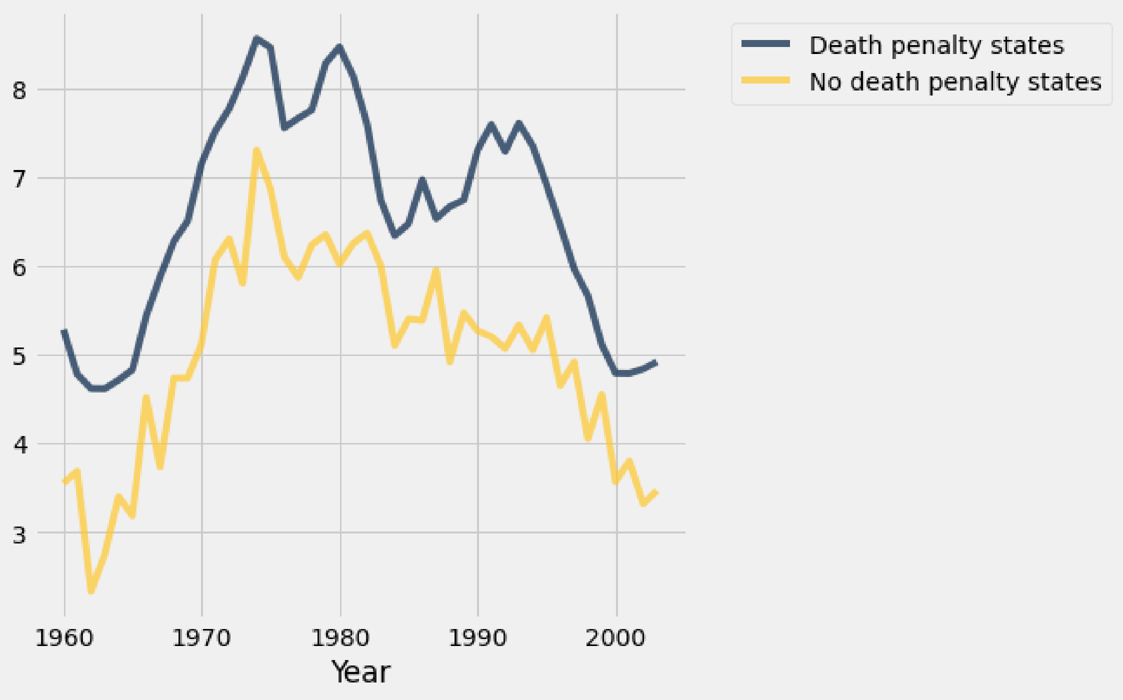
[58]:

```
check('tests/q3_1.py')
```

```
[58]: <gofer.ok.OKTestsResult at 0x7f3a2e3c5050>
```

Let’s take a look at the line plot below. Are the murder rates in these two groups of states related?

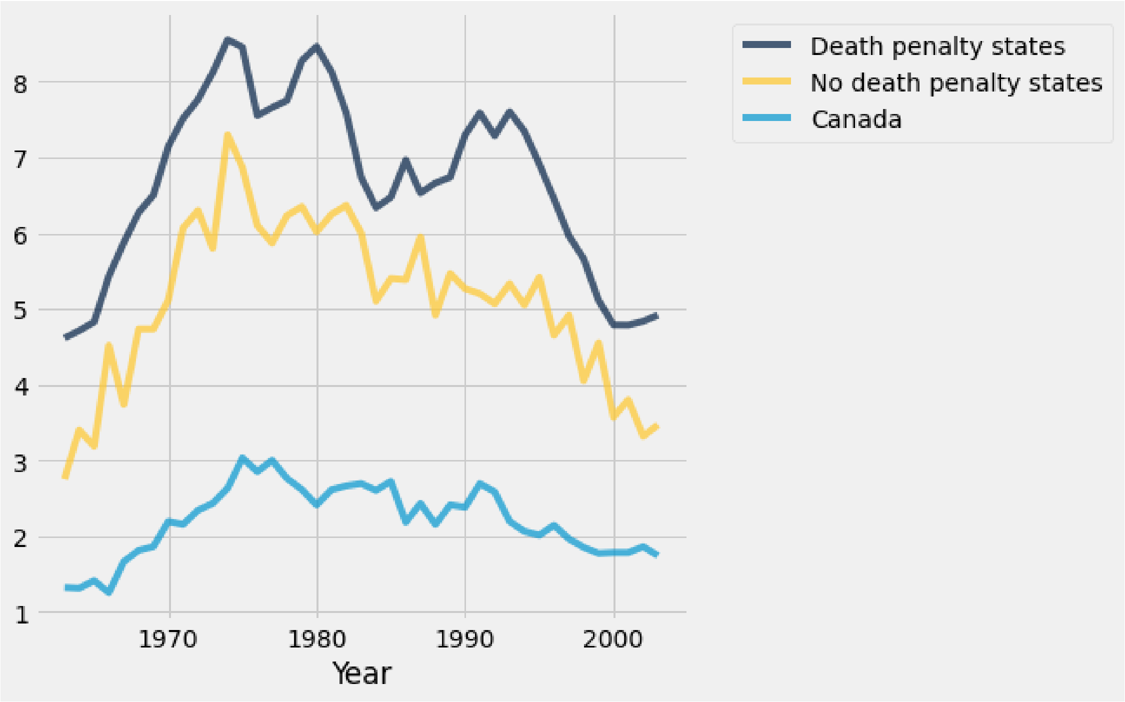
```
[59]: average_murder_rates.plot('Year')
```



Yes, murder rates in the two groups of states are related as we can see that they share a similar trend in the plot above.

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

```
[60]: canada = Table.read_table('canada.csv')
      murder_rates_with_canada = average_murder_rates.join("Year", canada.
        ↪select("Year", "Homicide").relabeled("Homicide", "Canada"))
      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 an 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.”

From the plots, we can conclude that murder rates in the two countries moved together even around points where death penalty legislation was passed. This seems to suggest that other changes over time determined murder rates in both countries, rather than one or the other’s legislation. This and the plots indicate that the death penalty is not an important factor in determining the murder rate. For more information, read the paper.

1.4.1 Conclusion

The authors that created this visualization argue that even though murder rates increased when the death penalty was taken away, and decreased when it was reintroduced, these changes were probably not caused by the death penalty itself.

1.5 4. Submission

Once you’re finished, select “Save and Checkpoint” in the File menu. Your name and course section number should be in the first and last cell of the assignment. Be sure you have run all cells with code and that the output from that is showing.

Double check that you have completed all of the free response questions as the autograder does NOT check that and YOU are responsible for knowing those questions are

there and completing them as part of the grade for this lab. When ready, click “Print Preview” in the File menu. Print a copy from there in pdf format. (This means you right click and choose print and choose “save as pdf” from your printer options.) You will need to submit the pdf in Canvas by the deadline.

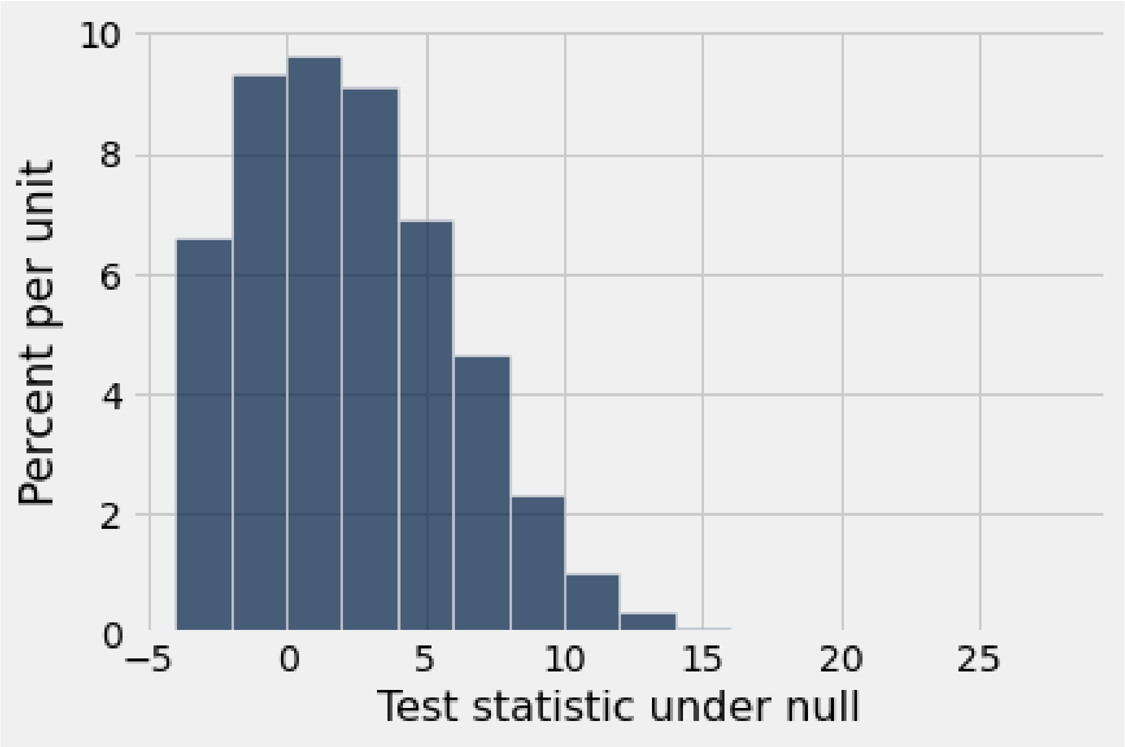
The gopher grader output and/or output from your coding are essential to helping your instructor grade your work correctly and in a timely manner.

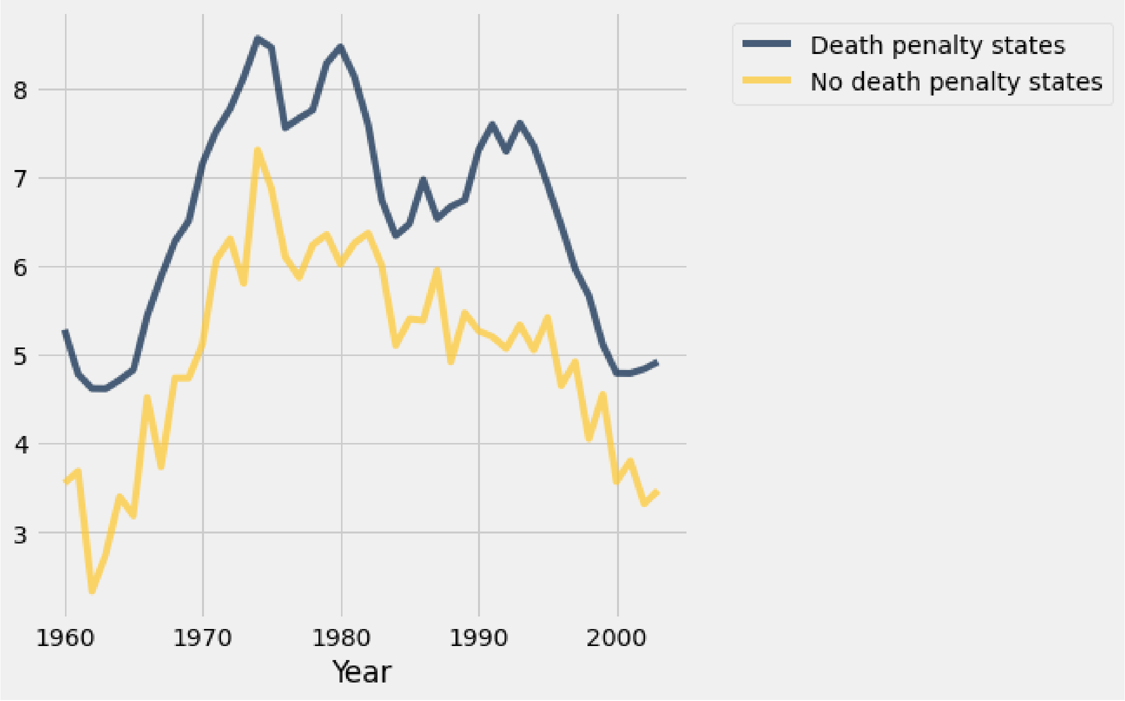
Files submitted that are missing the required output will lose some to all points so double check your pdf before submitting.

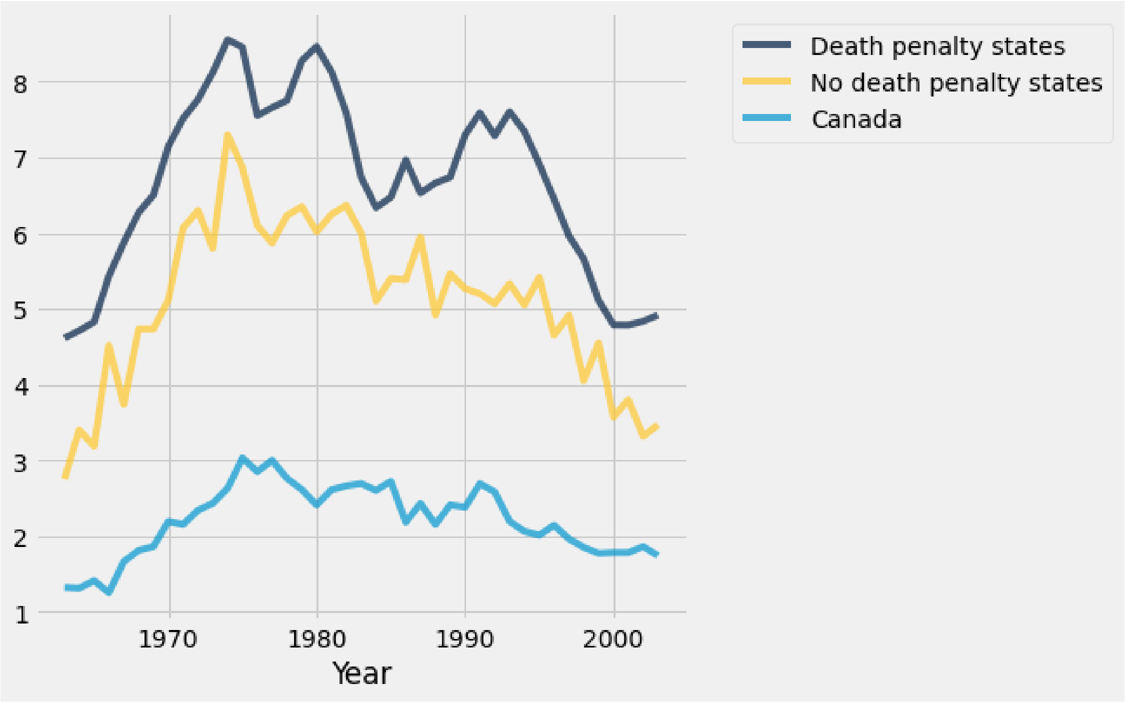
[61]:

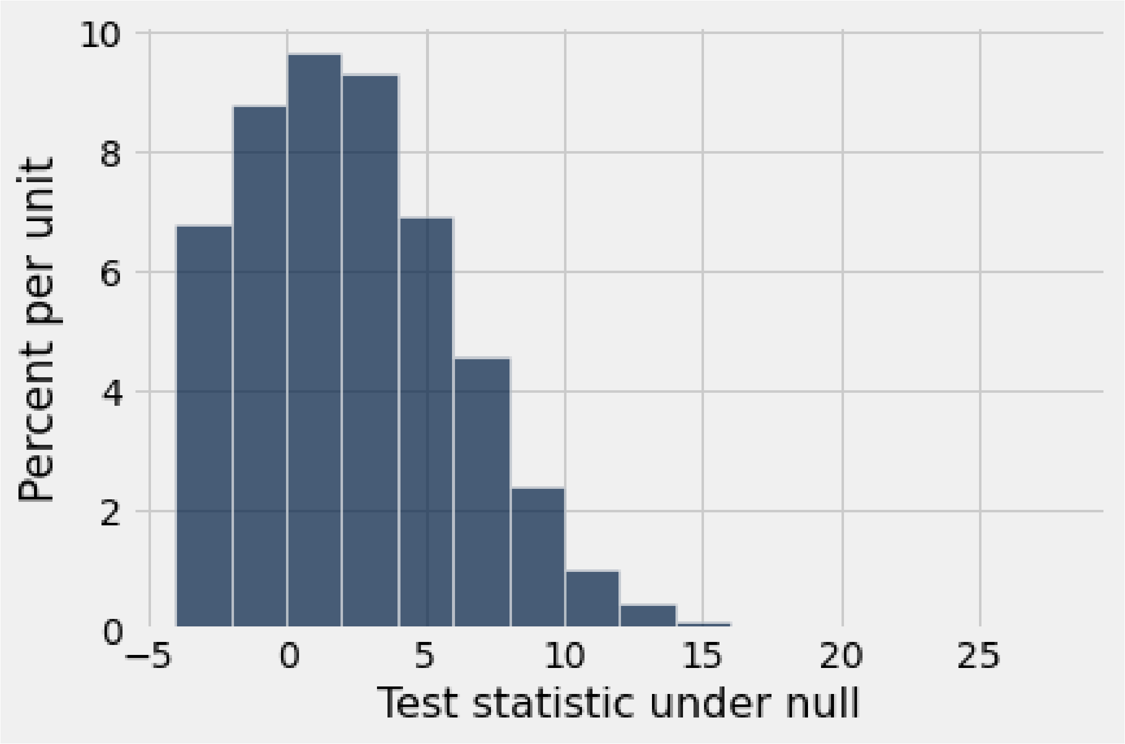
```
# For your convenience, you can run this cell to run all the tests at once!
import glob
from gofer.ok import grade_notebook
if not globals().get('__GOFER_GRADER__', False):
    display(grade_notebook('lab09.ipynb', sorted(glob.glob('tests/q*.py'))))
```

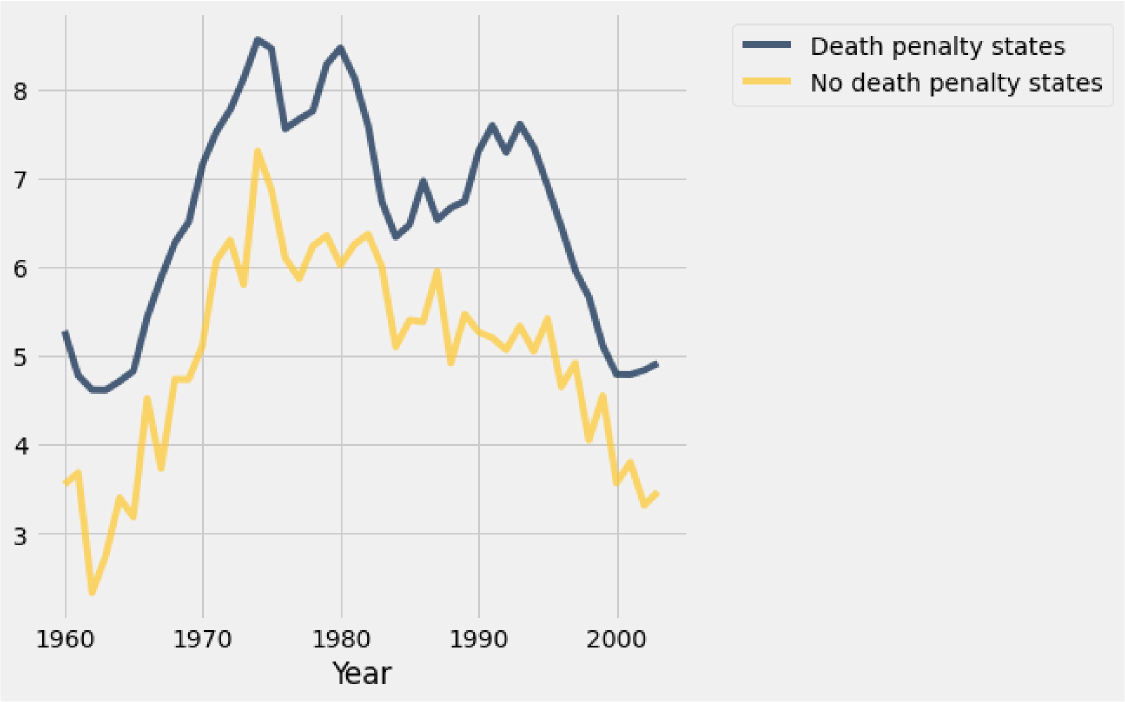
```
Test statistic from 1971 to 1973: 22
Test statistic 1971 to 1973 : 22
Test statistic 1971 to 1973 : 1
Increases minus decreases from 1975 to 1977 (when the death penalty was
reinstated) among death penalty states: -18
Test statistic 1975 to 1977 : -18
['tests/q1_3.py', 'tests/q1_4.py', 'tests/q2_1.py', 'tests/q2_2.py',
'tests/q3_1.py']
Question 1:

<gofer.ok.OKTestsResult at 0x7f3a2e4e0fd0>

Question 2:

<gofer.ok.OKTestsResult at 0x7f3a2e5bfd90>

Question 3:

<gofer.ok.OKTestsResult at 0x7f3a2e5e4190>

Question 4:

<gofer.ok.OKTestsResult at 0x7f3a2e66c8d0>

Question 5:

<gofer.ok.OKTestsResult at 0x7f3a2e66cdd0>

1.0
```

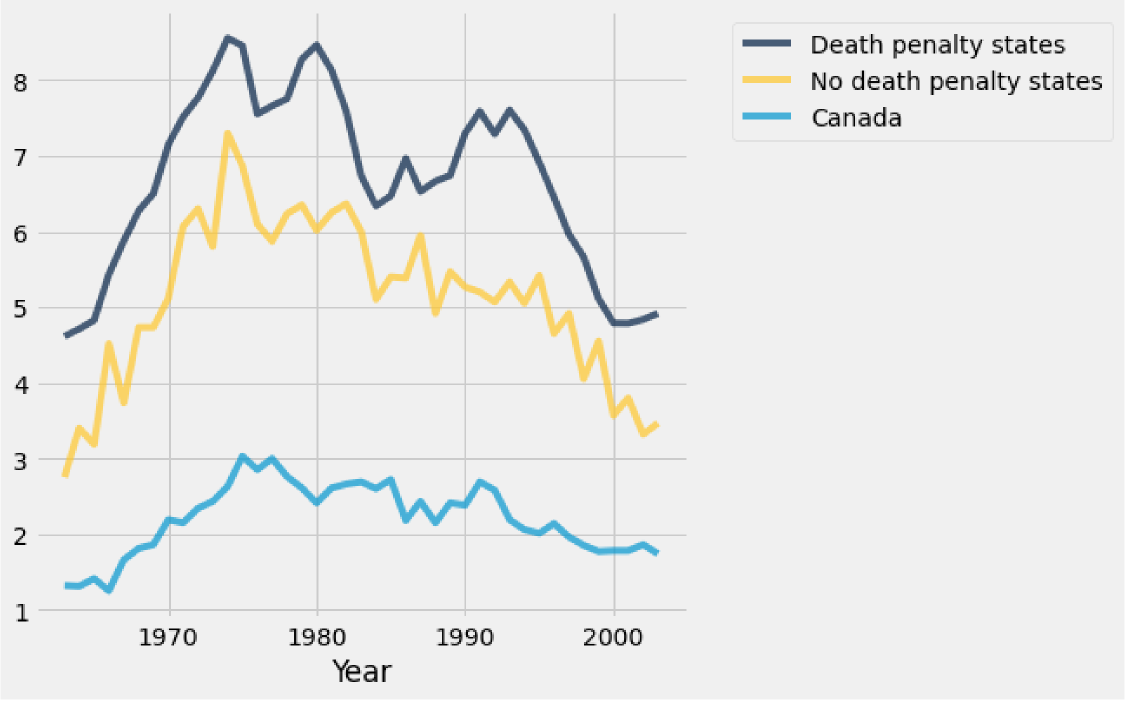

Percent per unit
10
8
6
4
2
0
−5
0
5
10
15
20
25
Test statistic under null


8
7
6
5
4
3
1960
1970
1980
1990
2000
Year
Death penalty states
No death penalty states


Death penalty states
No death penalty states
Canada
8
7
6
5
4
3
2
1
1970
1980
1990
2000
Year


Percent per unit
10
8
6
4
2
0
−5
0
5
10
15
20
25
Test statistic under null





Name: Allan Gongora

Section: 0131