Homework 3/4

- Release Date: Friday, April 10th
- Due Date: Monday April 27, 8:00 PM

Introduction

Punishment for crime has many philosophical justifications (http://plato.stanford.edu/entries/punishment/#ThePun) (this reading is optional; just in case you're interested!). 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, 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.

Here is a roadmap for this homework:

1. In Question 1, we'll investigate the main dataset we'll be using.
2. In Question 2, we'll see how to test null hypotheses such as this: "For this set of U.S. states, the murder rate was equally likely to go up or down each year."
3. In Question 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 Question 4, we'll run some more tests to further claims we had been developing in previous sections.
5. In Question 5, we'll try to answer our question about deterrence using a visualization rather than a formal hypothesis test.

We will guide you through the problems step by step. However, we encourage you to discuss with us in Office Hours so that we can work together through these steps.

In [1]:
```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt

# Set some parameters in the packages
%matplotlib inline

plt.rcParams['figure.figsize'] = (9,6)

pd.options.display.max_rows = 20
pd.options.display.max_columns = 15
```

Data

The main data source for this project comes from a paper (http://cjlf.org/deathpenalty/DezRubShepDeterFinal.pdf) 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.)

In [2]:
```
murder_rates = pd.read_csv(os.path.expanduser('~/shared/crime_rates.csv'
))
murder_rates = murder_rates[['State', 'Year', 'Population', 'Murder Rat
e']]
murder_rates.head()
```

Out[2]:

	State	Year	Population	Murder Rate
0	Alaska	1960	226167	10.2
1	Alaska	1961	234000	11.5
2	Alaska	1962	246000	4.5
3	Alaska	1963	248000	6.5
4	Alaska	1964	250000	10.4

Question 1: Murder rates

So far, this looks like a dataset that lends itself to an observational study. In fact, the murder rates dataset isn'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!

(1a) What additional information will we need before we can check for that association? Assign `extra_info` to a Python list (i.e. [#] or [#, #, ...]) containing the number(s) for all of the additional facts below that we ***require*** in order to check for association.

1) What year(s) the death penalty was introduced in each state (if any).
2) Day to day data about when murders occurred.
3) What year(s) the death penalty was abolished in each state (if any).
4) Rates of other crimes in each state.

For example, if you think we need options 1, 2 & 4, write: `extra_info = [1,2,4]`

In [3]:
```
extra_info = [1, 3] # <- put the number(s) for the answer to the above question in this list!
```

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 few just to see the variety.

To start, we generate a table `ak_mn` with two columns of murder rates, in addition to a column of years:

In [4]:
```
ak = murder_rates[murder_rates["State"] == 'Alaska'][["Murder Rate", "Year"]].rename(columns={"Murder Rate":'Murder rate in Alaska'})
mn = murder_rates[murder_rates["State"] == 'Minnesota'][["Murder Rate", "Year"]].rename(columns={"Murder Rate":'Murder rate in Minnesota'})
ak_mn = pd.merge(ak, mn)
ak_mn.head()
```

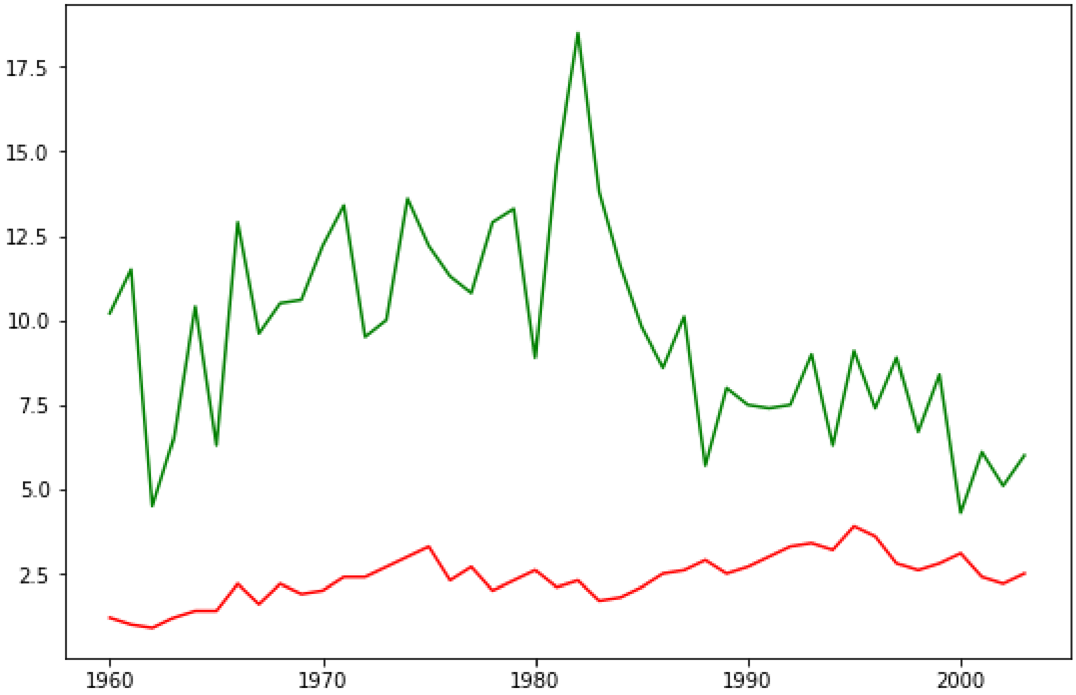
Out[4]:

	Murder rate in Alaska	Year	Murder rate in Minnesota
0	10.2	1960	1.2
1	11.5	1961	1.0
2	4.5	1962	0.9
3	6.5	1963	1.2
4	10.4	1964	1.4

(1b) Now, 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 using `plt.plot(x,y)` , where x is what data will be used for x axis and y is what data will be used for y axis.

In [5]:
```
# Draw your line plot here
plt.plot((ak_mn['Year']),(ak_mn['Murder rate in Alaska']), color = 'g')
plt.plot((ak_mn['Year']), (ak_mn['Murder rate in Minnesota']), color =
'r')
```

Out[5]:
```
[<matplotlib.lines.Line2D at 0x7fa6d07019d0>]
```



(1c) Now what about the murder rates of other states? Say, for example, California and New York? Complete the function `compare_state` by defining `state2_table`. This function takes as its input two states and creates a line graph simular to the above. `state2_table` should do the same as `state1_table` but for the second inputed state `state2` rather than for the first.

This will allow us to plot the murder rates of different pairs of states.

In [6]:
```
def compare_states(state1, state2):
    # Create the DataFrame for state1 by selecting only rows where the
 "State" column
    # equals state1, then pulling out just the "Murder Rate" and "Year"
 columns for
    # those states (for simplicity)
    state1_df = murder_rates[murder_rates["State"] == state1][["Murder R
ate", "Year"]]
    # Now the murder rate column is *specific* to state1, so we create a
new header for the column
    # indicating this
    state1_col_header = "Murder rate in " + str(state1)
    # Finally, we rename the generic "Murder Rate" column to have the st
ate1-specific column name
    state1_table.rename(columns={"Murder Rate": state1_col_header}, inpl
ace=True)
    # Your turn
    state2_table = murder_rates[murder_rates["State"] == state2] [["Murd
er Rate", "Year"]]
    state2_col_header = "Murder rate in " + str(state2)
    # And remember to rename the "Murder Rate" column here
    state2_table.rename(columns={"Murder Rate": state2_col_header}, inpl
ace=True)
    # This just combines state1_table and state2_table so we have all th
e info for
    # both states in a single DataFrame (called s1_s2)
    s1_s2 = pd.merge(state1_table, state2_table)
    plt.plot('Year')
    plt.show()
```

In [7]:
```
# You can compare the murder rates between different states
compare_states('California', 'New York')
```

```
----------------------------------------------------------------------
----
NameError                                 Traceback (most recent call l
ast)
<ipython-input-7-7afc6f1bccc6> in <module>
      1 # You can compare the murder rates between different states
----> 2 compare_states('California', 'New York')

<ipython-input-6-7c0595d18eb0> in compare_states(state1, state2)
      8     state1_col_header = "Murder rate in " + str(state1)
      9     # Finally, we rename the generic "Murder Rate" column to ha
ve the state1-specific column name
---> 10     state1_table.rename(columns={"Murder Rate": state1_col_head
er}, inplace=True)
     11     # Your turn
     12     state2_table = murder_rates[murder_rates["State"] == state2
] [["Murder Rate", "Year"]]

NameError: name 'state1_table' is not defined
```

Below we've provided a 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 [ ]: def most_murderous(year):
            # determine top 5 states
            data_for_year = murder_rates[murder_rates['Year'] == year]
            sorted_data = data_for_year.sort_values('Murder Rate', ascending=False)
            top = sorted_data[:5].set_index("State")

            # generate bar-chart
            top[['Murder Rate']].plot(kind = "barh");

            # determine list for output
            return list(top.index[::-1].values)
```

(1d) Using the above function, determine the 5 states with the highest murder rate in 1990. To td this, simply use `most_murderous` and assign the output to `top_5_1990` . (You should print the contents of the array at the end, to check that they are correct)

```
In [ ]: top_5_1990 = most_murderous(1990)
        top_5_1990
```

(1e) How many more people were murdered in New York in 1988 than in 1975? Assign `ny_change` to the answer.

Make sure you understand how murder rate is calculated. Recall the formula given at the beginning of this assignment:

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

Compute the change in the number of people murdered using the information we have calculated for you below. Every value in the above expression has been calculated for you, so by manipulating the formula and using the values you should be able to calculate the change in the number of people murdered.

In [ ]:

```
# DataFrames with information about New York in 1988 and 1975
ny_df = murder_rates[murder_rates['State'] == "New York"].copy()

# population
ny_1988_population = ny_df[ny_df["Year"] == 1988]["Population"].values[0
]
ny_1975_population = ny_df[ny_df["Year"] == 1975]["Population"].values[0
]

# murder rate
ny_1988_murder_rate = ny_df[ny_df["Year"] == 1988]["Murder Rate"].values
[0]
ny_1975_murder_rate = ny_df[ny_df["Year"] == 1975]["Murder Rate"].values
[0]
```

In [ ]:

```
# compute the change
ny_change = (((ny_1988_population)*(ny_1988_murder_rate))/100000) - (((n
y_1975_population)*(ny_1975_murder_rate))/100000)
ny_change
```

Question 2: Changes in Murder Rates

In this question, we'll see how to test null hypotheses such as this: "For this set of U.S. states, the murder rate was equally likely to go up or down each year."

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 changes -- *whether* they increased or decreased.**

Numpy provides an `np.diff()` function that takes a list/array of values as input and computes the differences between adjacent items of the list/array as such:

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

However, we may instead 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_2` function we provide below computes these two-position differences for a given list/array. Don't worry if the implementation uses unfamiliar features of Python, as long as you understand its behavior.

```
In [ ]: def diff_2(values):
            return np.array(values)[2:] - np.array(values)[:-2]

        diff_2([1, 10, 100, 1000, 10000])
```

(2a) Implement the function `two_year_changes` that takes an array of murder rates for a state, ordered by increasing year. It should:

1. Compute, for every **three-year period** in the array (e.g., [1960,1961,1962], then [1961,1962,1963], then [1962,1963,1964], and so on), the difference between the murder rate at the ***end*** of the period and the murder rate at the ***beginning*** of the period.
2. Return **the total number of increases (over *all* three-year periods in the dataset) minus the number of decreases.**

For example, the array `r = [10, 7, 12, 9, 13, 9, 11]` contains 3 increases (10 to 12, 7 to 9, and 12 to 13), 1 decrease (13 to 11), and 1 change that is neither an increase or decrease (9 to 9). Therefore, `two_year_changes(r)` would return 2, the difference between 3 increases and 1 decrease.

We have already started the function for you, with code that calculates the change in rates over two years using the `diff_2` function and assigns it to the variable `change_in_rates` (a Python list or NumPy array). Count the number of positive and negative changes in this list and assign them to the variables we created for you. Hint: what happens when we test if a numpy array is less than or greater than some value? The `np.count_nonzero` function may also be useful.

You are also free to not use the code and variables we gave you, so long as your function outputs the correct output.

```
In [ ]: def two_year_changes(rates):
            "Return the number of increases minus the number of decreases after
         two years."

            change_in_rates = diff_2(rates) # <- use this list in your code
            #Your code here using the variables we provided
            num_positive_changes = 0
            num_negative_changes = 0
            num_no_change = 0
            for num in change_in_rates:
                if num>0:
                    num_positive_changes = num_positive_changes + 1
                elif num<0:
                    num_negative_changes = num_negative_changes + 1
                else:
                    num_no_change = num_no_change + 1


            return num_positive_changes - num_negative_changes

        print('Alaska:',    two_year_changes(ak['Murder rate in Alaska']))
        print('Minnesota:', two_year_changes(mn['Murder rate in Minnesota']))
```

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.

In the code below we 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.

In [ ]:

```
changes_by_state = murder_rates[['State', "Murder Rate"]].groupby("State").agg(two_year_changes)
changes_by_state.head()
```

Question 2.2. Generate a histogram from `changes_by_state` using the column `Murder Rate` using the bins specified in `bins` . Your visualization should look something like the following.

In [ ]:

```
hist_bins = np.arange(-11, 19, 2)
# Write code to generate the histogram here, using hist_bins for the bins argument
plt.hist(changes_by_state["Murder Rate"], hist_bins)
plt.show()
```

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

(2c) We now have a column `Murder Rate` in the dataframe `changes_by_state` that has, for each state, the net number of times the murder rate increased over a two year period. What is the total for all states?

Specifically, assign `total_changes` to the total increases minus the total decreases for all two-year periods and all states in our data set. We want the total value for all the states together (meaning every observation in this dataframe) and to assign this value to the variable `total_changes` .

In [ ]:

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

Now one person might say...

> "More increases than decreases! Murder rates tend to go up across two-year periods. What dire times we live in."

While another person might say

> "Not so fast. 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. If state murder rates increase and decrease at random with equal probability, perhaps this difference was simply due to chance!"

(2d) Below we have a list of all years in the dataset `distinct_years` and a list of all states in the dataset `distinct_states`. What is the total number of distinct state and two-year period pairs in our dataset? Assign `num_changes` to this value. For example, Alaska during 1968 to 1970 would count as one distinct pair.

To illustrate, if we only had years 1990, 1991, 1992, 1993, 1994 we know there are five years but only three two year period pairs, which woud be [1990,1992], [1991,1993],[1992,1994]. If there were fifty states in the dataset we would therefore have 150 distinct state and two-year period pairs in our dataset.

Given the years and states we do have, what is this number?

Importantly, we know that every state has the same number of years in the dataset. We've also created lists of the unique states and years in the dataset that you should feel free to use in this calculation.

In [ ]:

```
distinct_states = murder_rates['State'].unique()
distinct_years = murder_rates['Year'].unique()
num_changes = len(distinct_states) * len(diff_2(distinct_years))
num_changes
```

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.

Murder rates can be more likely to go up or more likely to go down. Since we observed 45 more increases than decreases for all two year periods in our dataset, we formulate an alternative hypothesis in accordance with our suspicion:

> **Alternative Hypothesis**: State murder rates are more likely to increase over two-year periods.

If we had observed more decreases than increases, our alternative hypothesis would have been defined accordingly (that state murder rates are more likely to ***decrease***). This is typical in statistical testing - we first observe a trend in the data and then run a hypothesis test to confirm or reject that trend.

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.

Important requirements for our test statistic: We want to 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. This is because once we've determined the direction of our alternative hypothesis, we only care about the tail in that direction. If, for example, our p-value cutoff was 5%, we'd check to see if our observed test statistic fell within the largest 5% of values in our null hypothesis distribution.

Our test statistic should depend only on whether murder rates increased or decreased, not on the size of any change. Thus we choose:

> **Test Statistic**: The number of increases minus the number of decreases

(2e) The histogram below is an empirical distribution of the test statistic under the null hypothesis. Looking at this histogram, draw a conclusion about whether murder rates basically increase as often as they decrease. (Remember that we're only concerned with the ***postive direction*** because it supports our alternative hypothesis.) You **do not** need to compute a P-value for this question.

First, set `which_side` to `"Right"` or `"Left"` depending on which side of the histogram you need to look at to make your conclusion.

Then, set `reject_null` to `True` if rates increase more than they decrease, and we can reject the null hypothesis. Set `reject_null` to `False` if they do not systematically increase more than they decrease.

In [ ]:

```
which_side = "Right"
reject_null = False
```

Question 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 presence of the death penalty affects the murder rate of a state.

(3a) 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!

Population= All US States Control Group= States without the death penalty Treatment= States with the death penalty enforced Outcome measured= We can observe the changes in murder rates between states without the death penalty and those with the death penalty

(3b) 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?

I personally do not think this would be a fair test because it does not say much about the causality of the death penalty. I think the only thing you can really take away is observations about the population: some people in some states may have a lesser tendency to murder others? Maybe there are more urban areas in some states which have more murder crime as opposed to another state. There are too many inconsistencies amongst the states to make any conclusions. Beyond this, it is an observational study and we cannot make causal inferences without doing an experiement.

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 is in keeping with 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.

In [ ]:

```
non_death_penalty_states = np.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

death_penalty = pd.DataFrame(distinct_states,columns=['State'])  # Make sure you ran the cell where distinct_states is defined
death_penalty['Death Penalty'] = death_penalty.apply(had_death_penalty_in_1971,axis='columns',raw=True)
death_penalty.head()
```

In [ ]:

```
num_death_penalty_states = np.sum(death_penalty["Death Penalty"])
num_death_penalty_states
```

(3c) 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.

Firstly, we create a variable called `death_in_1971` that is `True` when a state had the death penalty in 1971, and `False` otherwise. Use that variable to subset the data.

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

State	Year	Population	Murder Rate

44|Alabama|1960|3266740|12.4| **45**|Alabama|1961|3302000|12.9|

In [ ]:

```
# non_death_penalty states, the bitwise not operator (~), and the pd.DataFrame function isin()
# are all good to know for data wrangling
death_in_1971 = ~murder_rates['State'].isin(non_death_penalty_states)

death_penalty_murder_rates = murder_rates[death_in_1971 == True]
death_penalty_murder_rates
```

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 Question 2: looking at the number of increases minus number of decreases for all states.

In the code below, we create a variable `murder_rate_changes_by_state`. This denotes whether, from 191 to 1973, the murder rate increased (1) or decreased (-1) for each state.

In [ ]:

```
# Keep only the rows of death_penalty_murder_rates where the "Year" variable is one of [1971,1972,1973]
from_71_to_73 = death_penalty_murder_rates[death_penalty_murder_rates['Year'].isin([1971,1972,1973])]
# And now calculate whether the murder rate increased or decreased during that three year period for each state
murder_rate_changes_by_state = from_71_to_73[['State', 'Murder Rate']].groupby('State').apply(two_year_changes)
murder_rate_changes_by_state
```

(3d) Given the `murder_rate_changes_by_state` variable we've now computed, it should be straightforward to use it to calculate the total number of changes. 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."

In [ ]:

```
test_stat_72 = sum(murder_rate_changes_by_state)
print('Test statistic from 1971 to 1973:', test_stat_72)
```

Next we're going to draw an empirical histogram of the statistic under the null hypothesis by simulating the test statistic 10,000 times. First we will define two helper functions for you:

- `simulate_state()` : This function simulates the 3-year murder rate "trajectory" of a a state by flipping two coins. The first coin represents whether the state experiences an increase or decrease in its murder rate after the first year in a three-year span (an increase if heads, a decrease if tails), and the second represents whether it experiences an increase or decrease in its murder rate between the second and third year in the same manner. It then returns the number of increases (how many coins landed heads) minus the number of decreases (how many coins landed tails). **You do not need to use this function directly, it is used in the below function.**
- `simulate_under_null(num_states)` : This function takes in a number of states `num_states` (which you'll need to define), then uses the `simulate_state` function to simulate three year spans for each state. It then sums up the test statistics from each individual state to compute a final "nationwide" test statistic of the number of increases minus the number of decreases over ***all*** `num_states` states. **You do need to use this function directly in your simulation code.**

In [ ]:
```
def simulate_state():
    # Flip 2 coins -- if heads (1) then state experiences an increase, o
therwise (-1) a decrease
    first_change = np.random.choice([-1,1])
    second_change = np.random.choice([-1,1])
    total_change = first_change + second_change
    return total_change

def simulate_under_null(num_states):
    # Simulate `num_states` states either increasing or decreasing their
murder rates
    # over a 2 year span, then sum to get the full test statistic
    state_stats = []
    for state_num in range(num_states):
        # Simulate this state increasing or decreasing twice
        current_state_stat = simulate_state()
        state_stats.append(current_state_stat)
    return sum(state_stats)
```

(3e): Now, write code using these two helper functions to perform 10,000 simulations of the 1971 to 1973 murder rate changes, thus producing 10,000 test statistics, and store these test statistics in a Python list called `simulated_test_statistics` . We've provided some starter code so you only need to replace the `...` lines:

Hint: Remember to provide an input to `simulate_under_null` of the correct amount of U.S. states.

In [ ]:
```
num_simulations = 10000
simulated_test_statistics = np.zeros(num_simulations) #<- use this np.ar
ray in your code
                                                      # Creating an "empt
y" np.array and changing the values inside is more efficient than append
ing to a np.array
for simulation_num in range(num_simulations):
    # Your code here.

    simulated_test_statistics[simulation_num] = simulate_under_null (50)
    #Hint: simulated_test_statistics[simulation_num] = ...

# Outputs the first 100 test statistics out of the 10,000 total simulati
ons
simulated_test_statistics[:100]
```

Now you can run this cell to draw an empirical histogram of the statistic under the null hypothesis.

In [ ]:

```
#plt.hist(samples['Test statistic under null'], bins=np.arange(-4, 28+2, 2))
test_stat_df = pd.DataFrame({'test_stat_under_null': simulated_test_statistics})
plt.hist(test_stat_df['test_stat_under_null'], bins=np.arange(-4, 28+2, 2))
plt.show()
```

Conclusion

(3f) Complete the analysis as follows:

First, lets calculate the p-value.

The variable `simulated_test_statistics` is a np.array, where each value is a number from a specific sample. Those numbers will be, for each simulation, the number of increases in murder rates over 1971 to 1973 if the null was true.

Using `simulated_test_statistics` , which is a np.array of what we simulated under the null, and `test_stat_72` which is our actual observed observation, calculate the p-value and assign it to the variable `sign_72_p_value` .

In [ ]:

```
sign_72_p_value = num_greater/10000
sign_72_p_value
```

(3g) Complete the analysis as follows:

1. Write the 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?

My p value is .0192 which is smaller than the .5 p value cut off. I believe this means we should reject the null. Murder rates probably icreased after the death penalty was gone. I do not think you can suggest causation however there is a positive association.

Question 4: Visualization

While our analysis appears to support the conclusion that the death penalty deters murder, a 2006 Stanford Law Review paper (http://users.nber.org/~jwolfers/papers/DeathPenalty%28SLR%29.pdf) 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!

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.

Here, we've "drawn" the pictures for you and will be asking for your takeaways from these plots.

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

(4a) The line plot below shows average murder rates per year, comparing states with the death penalty against states without the death penalty. Describe in **one short sentence** a high-level takeaway from this plot. Are the murder rates in these two groups of states related?

There is a correlation in trends of murders in no death penalty states and death penalty states

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

The line plot shown above is similar to a figure from the paper (http://users.nber.org/~jwolfers/papers/DeathPenalty%28SLR%29.pdf).

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."

(4b) 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.)

Murder rates in the US incrased even while the death penalty was being enforced. It also did not increase while there was an increase to abolish the death penalty either. Clearly there must be other confounding factors maybe like gun laws, social equality etc. that must be present in the data.

(4c) 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. Based on your analysis, what conclusion can you make? Address the steps you took throughout the project in your answer as well.

The paper focuses on murder rates over a long period of time whereas the analysis we did focuses on the change in murder rates specifically after a 1972 policy change. The analysis we did is too short a period of time to develop any sort of causation or important association. Our analysis might have missed out on different trends regarding murder rates beyond 2 years.

You're done! Congratulations!

In [ ]: