Py hypothesis test education

June 15, 2023

1 Hypothesis testing

Throughout the following exercises, you will learn to use Python to conduct a two-sample hypothesis test. Before starting on this programming exercise, we strongly recommend watching the video lecture and completing the IVQ for the associated topics.

All the information you need for solving this assignment is in this notebook, and all the code you will be implementing will take place within this notebook.

As we move forward, you can find instructions on how to install required libraries as they arise in this notebook. Before we begin with the exercises and analyzing the data, we need to import all libraries and extensions required for this programming exercise. Throughout the course, we will be using pandas and scipy stats for operations.

[1]:
```
import pandas as pd
from scipy import stats
```

[2]:
```
education_districtwise = pd.read_csv("education_districtwise.csv")
education_districtwise = education_districtwise.dropna()
```

We'll continue with our scenario from an earlier part of the course, in which you're a data professional working for the department of education of a large nation. Recall that you're analyzing data on the literacy rate for each district.

Now imagine that the department of education asks you to collect data on mean district literacy rates for two of the nation's largest states: STATE21 and STATE28. STATE28 has almost 40 districts, and STATE21 has more than 70. Due to limited time and resources, you are only able to survey 20 randomly chosen districts in each state. The department asks you to determine if the difference between the two mean district literacy rates is statistically significant, or due to chance. This will help the department decide how to distribute government funding to improve literacy. If there is a statistically significant difference, the state with the lower literacy rate may receive more funding.

You can use Python to simulate taking a random sample of 20 districts in each state, and conduct a two-sample t-test based on the sample data.

1.0.1 Organize your data

To start, filter your data frame for the district literacy rate data from the states STATE21 and STATE28.

First, name a new variable: `state21`. Then, use the relational operator for equals (`==`) to get the relevant data from the `STATNAME` column.

[4]:
```
state21 = education_districtwise[education_districtwise['STATNAME'] ==␣
↪"STATE21"]
```

Next, name another variable: `state28`. Follow the same procedure to get the relevant data from the `STATNAME` column.

[5]:
```
state28 = education_districtwise[education_districtwise['STATNAME'] ==␣
↪"STATE28"]
```

1.0.2 Simulate random sampling

Now that you've organized your data, use the `sample()` function to take a random sample of 20 districts from each state. First, name a new variable: `sampled_state21`. Then, enter the arguments of the `sample()` function.

- `n`: Your sample size is `20`.
- `replace`: Choose `True` because you are sampling with replacement.
- `random_state`: Choose an arbitrary number for the random seed – how about `13490`. .

[6]:
```
sampled_state21 = state21.sample(n=20, replace = True, random_state=13490)
```

Now, name another variable: `sampled_state28`. Follow the same procedure, but this time choose a different number for the random seed - how about 39,103.

[7]:
```
sampled_state28 = state28.sample(n=20, replace = True, random_state=39103)
```

1.0.3 Compute the sample means

You now have two random samples of 20 districts, one sample for each state. Next, use `mean()` to compute the mean district literacy rate for both STATE21 and STATE28.

[8]:
```
sampled_state21['OVERALL_LI'].mean()
```

[8]:
```
70.82900000000001
```

[9]:
```
sampled_state28['OVERALL_LI'].mean()
```

[9]:
```
64.60100000000001
```

STATE21 has a mean district literacy rate of about 70.8%, while STATE28 has a mean district literacy rate of about 64.6%.

Based on your sample data, the observed difference between the mean district literacy rates of STATE21 and STATE28 is 6.2 percentage points (70.8% - 64.6%).

Note: At this point, you might be tempted to conclude that STATE21 has a higher overall literacy rate than STATE28. However, due to sampling variability, this observed difference might simply be due to chance - rather than an actual difference in the corresponding population means. A hypothesis test can help you determine whether or not your results are statistically significant.

1.1 Conduct a hypothesis test

Now that you've organized your data and simulated random sampling, you're ready to conduct your hypothesis test. Recall that the two-sample t-test is the standard approach for comparing the means of two independent samples. Let's review the steps for conducting a hypothesis test:

1. State the null hypothesis and the alternative hypothesis
2. Choose a significance level
3. Find the p-value
4. Reject or fail to reject the null hypothesis

1.1.1 Step 1: State the null hypothesis and the alternative hypothesis

The **null hypothesis** is a statement that is assumed to be true unless there is convincing evidence to the contrary. The **alternative hypothesis** is a statement that contradicts the null hypothesis, and is accepted as true only if there is convincing evidence for it.

In a two-sample t-test, the null hypothesis states that there is no difference between the means of your two groups. The alternative hypothesis states the contrary claim: there is a difference between the means of your two groups.

We use H_0 to denote the null hypothesis, and H_A to denote the alternative hypothesis.

- H_0: There is no difference in the mean district literacy rates between STATE21 and STATE28
- H_A: There is a difference in the mean district literacy rates between STATE21 and STATE28

1.1.2 Step 2: Choose a significance level

The **significance level** is the threshold at which you will consider a result statistically significant. This is the probability of rejecting the null hypothesis when it is true. The education department asks you to use their standard level of 5%, or 0.05.

1.1.3 Step 3: Find the p-value

P-value refers to the probability of observing results as or more extreme than those observed when the null hypothesis is true.

Based on your sample data, the difference between the mean district literacy rates of STATE21 and STATE28 is 6.2 percentage points. Your null hypothesis claims that this difference is due to chance. Your p-value is the probability of observing an absolute difference in sample means that is 6.2 or greater *if* the null hypothesis is true. If the probability of this outcome is very unlikely - in particular, if your p-value is *less than* your significance level of 5% – then you will reject the null hypothesis.

`scipy.stats.ttest_ind()` For a two-sample t-test, you can use `scipy.stats.ttest_ind()` to compute your p-value. This function includes the following arguments:

- `a`: Observations from the first sample.
- `b`: Observations from the second sample.
- `equal_var`: A boolean, or true/false statement, which indicates whether the population variance of the two samples is assumed to be equal. In our example, you don't have access to data for the entire population, so you don't want to assume anything about the variance. To avoid making a wrong assumption, set this argument to `False`.

Reference: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html.

Now you're ready to write your code and enter the relevant arguments:

- `a`: Your first sample refers to the district literacy rate data for STATE21, which is stored in the `OVERALL_LI` column of your variable `sampled_ state21`.
- `b`: Your second sample refers to the district literacy rate data for STATE28, which is stored in the `OVERALL_LI` column of your variable `sampled_ state28`.
- `equal_var`: Set to `False` because you don't want to assume that the two samples have the same variance.

[10]:
```
stats.ttest_ind(a=sampled_state21['OVERALL_LI'],␣
↪b=sampled_state28['OVERALL_LI'], equal_var=False)
```

[10]:
```
Ttest_indResult(statistic=2.8980444277268735, pvalue=0.006421719142765231)
```

Your p-value is about 0.0064, or 0.64%.

This means there is only a 0.64% probability that the absolute difference between the two mean district literacy rates would be 6.2 percentage points or greater if the null hypothesis is true. In other words, it's highly unlikely that the difference in the two means is due to chance.

1.1.4 Step 4: Reject or fail to reject the null hypothesis

To draw a conclusion, compare your p-value with the significance level.

- If the p-value is less than the significance level, you conclude there is a statistically significant difference in the mean district literacy rates between STATE21 and STATE28. In other words, you reject the null hypothesis H_0.
- If the p-value is greater than the significance level, you conclude there is *not* a statistically significant difference in the mean district literacy rates between STATE21 and STATE28. In other words, you fail to reject the null hypothesis H_0.

Your p-value of 0.0064, or 0.64%, is less than the significance level of 0.05, or 5%. So, you *reject* the null hypothesis, and conclude that there is a statistically significant difference between the mean district literacy rates of the two states STATE21 and STATE28.

Your analysis will help the education department decide how to distribute government resources. Since there is a statistically significant difference in mean district literacy rates, the state with the lower literacy rate, STATE28, will likely receive more resources to improve literacy.

If you have successfully completed the material above, congratulations! You now understand how to use Python to conduct a two-sample hypothesis test. Going forward, you can start using Python to conduct hypothesis tests on your own data.