INFO370 Problem Set

July 19, 2021

Instructions

The aim of this problem set is to learn about statistical hypotheses, hypothesis testing, and t-test. It follows broadly the approach of the lab 5 in the sense that you are first asked to generate random data under H_0, and later to use the corresponding formula. Hower, here we look at a slightly different task where the question is to compare two continuous outcomes, not a single proportion.

In this dataset you use AirBnB price data. The data originates from AirBnB, and it contains listings that are 1BR apartments with one bathroom, that accommodate two, and the price is for two persons.

1 Is Beijing more expensive than Seattle? (50pt)

You will proceed as follows: first, you compute the difference between the average log price in Beijing and Seattle. Thereafter you create two samples of random normal numbers as in the data above, using the mean and standard deviation over combined Beijing and Seattle listings. Call one of these samples "fake Beijing" and the other "fake Seattle". What is the difference of their means? And now you repeat this exercise many-many times and see if you can get as big a difference between the fake Beijing and fake Seattle as there is between real Beijing and real Seattle.

1. (2pt) load the data *beijing-seattle-airbnb-price.csv*. It contains two variable: *city* (Beijing/Seattle) and *price* (in USD).

   Perform basic description of it: what is the number of observations? Are there any missings or otherwise invalid entries?

2. (3pt) Describe the price: compute the mean, median, standard deviation, and range. According to these figures, which city is more expensive, Beijing or Seattle?

Below, we are going to do t-test. However, t-test works best if the data is normally distributed. However, price data is typically distributed approximately *log-normally*, not normally. (Log-normal is a distribution of such RV where distribution of log of it is normal.)

3. (3pt) Demonstrate that the data is approximately log-normally distributed: plot histograms of price and log price. Comment the shape of the histograms.

From now on we work only in log-price.

4. (1pt) Convert price to logs.

5. (4pt) Compute the mean difference between Beijing and Seattle listings. (The answer is 0.739.)

This was the basic description of the data. Now onward to the comparison. We proceed as follows: Imagine that there is no real difference between Beijing and Seattle prices. We call this null-hypothesis H_0. Hence whatever difference we see in the actual data is just random sampling noise. We would like to have a huge number of listings to test it, but unfortunately we do not have that. So we do this instead: we create fake Beijing listings and fake Seattle listings, both drawn from the same distribution. There must be as many fake ones as there are real ones in the data. Thereafter we compare the mean price: how much more expensive are the fake Beijing prices compared to fake Seattle prices? We repeat this process many times and at the end we report how often did we find a difference that is similar to what we observe in the real data. If this is a common occurrence, we cannot reject H_0.

6. (5pt) Let's state our H_0 again: Beijing and Seattle listings are of similar price (in average). Hence we have to create fake Beijing and fake Seattle prices using the same distribution. The obvious choice for this is the distribution of combined Beijing and Seattle log prices.

   Compute the overall mean μ_0 and standard deviation σ_0 of combined Beijing and Seattle prices.

   Hint: the standard deviation is 0.642.

7. (5pt) Now create two sets of random normals, "fake Beijing" and "fake Seattle", both with the same mean μ_0 and standard deviation σ_0 that you just computed above. The number of fake prices must be the same as the number of real prics for the corresponding city in the data.

   What is the difference between the average fake Beijing prices and average fake Seattle prices? Compare the result with the real difference you found above.

   Hint: say, the average is 5 and standard deviation is 0.5. You can create the corresponding normals like:

```
fakeB = np.random.normal(5, 0.5, size=20)  # create 20 fake
Beijing prices
fakeS = np.random.normal(5, 0.5, size=10)  # create 10 fake
Seattle prices
fakeB

## array([4.53725191, 4.85296321, 4.806473  , 4.41112139, 5.24563113,
##        5.31052972, 4.95880275, 5.26150267, 4.5434593 , 5.6461771 ,
##        4.77650392, 4.16534866, 4.34767097, 5.23501978, 5.62642792,
##        4.38070638, 4.47660532, 6.0679618 , 4.52303091, 4.92782542])

fakeS

## array([4.87591738, 5.28382483, 5.17267181, 4.78044328, 5.13577244,
##        4.54925033, 4.33785949, 4.76012063, 5.09110172, 4.96747069])
```

   compute the mean difference:

```
np.mean(fakeB) - np.mean(fakeS)

## 0.009607402537952225
```

   Now compare this number with what you see in data.

8. (5pt) Now repeat the previous question a large number R (1000 or more) times. Each time store the mean difference between fake Beijing and fake Seattle, so you end up with R different values for the mean difference.

9. (5pt) What is the mean of the mean differences? If you did your simulations correctly, it should be close to 0. Explain why do you get this result.

10. (4pt) What is the largest mean difference (in absolute value) in your sample?

    Hint: `np.abs` computes absolute value.

11. (7pt) find 95% confidence interval (CI) of your sample of mean differences based on sample quantiles. Does the difference in actual data, 0.739 in favor of Beijing, fall into the CI?

    Hint: use `np.percentile(2.5)` and a similar expression for the 97.5th percentile.

12. (7pt) Finally, based on the simulations, what is your conclusion: is the observed difference 0.739 just a random fluke, or are prices in Beijing really more expensive than in Seattle?

2 Now repeat the above with t-test (40pt)

Above we spent a lot of effort with sampling, random numbers and such. In practice, it is usually not possible to sample millions of listings. And even if feasible, it is much easier just to do a t-test. Below we ask you to *compute the* t*-value yourself*, do not use any pre-existing functions!

1. (10pt) Compute standard error SE of the Beijing-Seattle mean difference. Remember: we are still working in logarithms!

   Hint: read OIS 7.3, p 267. You probably have to walk back and read about various other concepts the book is using in 7.3.

2. (10pt) Compute 95% CI.

   Use the 5% two-tail significance level to look up $\mathtt{t_{cr}}$ values in t-distribution table. OIS has such a table in Appendix C.2, and google can find a million more similar tables.

   95% CI is given by $\mu \pm \mathtt{t_{cr}} \cdot \mathsf{SE}$ where μ is the mean, SE is its standard error, and $\mathtt{t_{cr}}$ is the critical value from the table.

   Hint 1: what is the *degrees of freedom* in current case? Consult OIS 7.3.

Hint 2: we need 2-tailed test as Beijing can be both cheaper and more expensive than Seattle.

Hint 3: you can do this in two ways. Compute 95% CI around H_0 value (i.e. difference is 0) and check if the actual difference fits in there (this is what we simulated above). Or compute 95% CI around the actual value, and check of H_0 value 0 fits in there (this is what we did in Lab05 Q2). DO NOT compute 95% around actual value and then check if the actual value fits in there. It always does!

3. (6pt) What will you conclude based on CI: can you reject H_0, Beijing and Seattle are of equally expensive, at 5% level?

4. (8pt) Now perform the opposite operation: compute the t-value. When the you have mean μ and standard error SE, you can compute the t-value by

$$t = \frac{\mu}{SE}$$

Hint: the answer is 15.98.

5. (6pt) What is the likelihood that such a t value happens just by random chance? Consult the t-table.

Hint: I have never seen t-tables that contain such large values. But where on the table would you write this value? What can you say about how likely it is to see such a value just by random chance?

3 Use canned t-test function (10pt)

Finally, we use a ready-made library: `scipy.stats.ttest_ind` contains ready-made t-test function.

Remember: work with log price!

1. (5pt) Compute t-value and the probability using `ttest_ind`.

Note: you have to specify `equal_var=False` to tell the function that Beijing and Seattle price may have different variance.

2. (5pt) Finally, state your conclusion: is Beijing more expensive than Seattle? Do all of your three methods: simulations, 95% CI, t-value and python's t-test agree?

4 Challenge (not graded)

How long time do you need to simulate to get the mean difference in log price between Beijing and Seattle that you actually observe in data, 0.739?

If you did the previous tasks well, you noticed that simulated differences are way smaller than the actual differences, and even millions of experiments do not bring you close. But how long time do you have to run the simulations to actually get close?

1. (3pt) First, time your simulations. Run a large number of repetitions `R`, say 1M, and measure how long it takes on your computer. You should aim for `R` that makes you computer to simulate at least 5 seconds for your measurements to be precise enough. Now you can easily calculate how long it would take to run 10^{12} or so experiments.

   Hint: check out `%timeit` and `%time` magic macros.

2. (3pt) Second, what is the probability to receive such enormous `t`-values? As these are off the `t` tables, you have to compute the corresponding probability yourself.

   Assume we are dealing with normal distribution. (Not quite but we are close.) You have to compute the probability you get a value larger than the `t` value you computed. This can be done along the lines:

   ```
   from scipy import stats
   norm = stats.norm()
   norm.cdf(-1.96)  # close to 0.025

   ## 0.024997895148220435
   ```

   where you replace 1.96 with your actual `t`-value.

   Explain: why does the example use `norm.cdf(-1.96)` instead of `norm.cdf(1.96)`?

3. (2pt) How many iterations you need? Let's do a shortcut–if probability `p` is small, you need roughly $3/p$ iterations. So if $p = 0.001$, you need 3000 iterations.

4. (2pt) Based on the timings you did above, how many years do you have to run the simulations?

   If one had started the computer the year your grandfather was born, would it be there now?

If the first Seattle inhabitants had started it when they moved here following the melting ice, 10,000 or so years ago?

If the last dinosaurs had started it 66,000,000 years ago? (But it must have been in Idaho or somewhere else, the land where Seattle is now did not exist back then.)