INFO 370 Problem Set 4: Comparing Means - Housing

April 21, 2022

Instructions

The aim of this problem set is to work with **and interpret** hypothesis tests and t-tests. To do so appropriately, we will also need to be competent in data exploration, visualization, and transformations. In general, it broadly follows the logic of this week's lab, but focuses on comparing two different outcomes, instead of a single outcome and a null value.

Dataset

In this dataset you will use a sample of all AirBnB listings from Beijing and Seattle. The data is downloaded from AirBnB, http://insideairbnb.com/get-the-data.html. There is no documentation regarding the variables and what do they mean. So we cannot provide much guidance. However, only two variables are of interest for today:

- *city* (values as either 'Beijing' or 'Seattle')
- *price* (in USD)

1 Part One: Data Exploration (10 points)

In this section, you are asked to descriptively explore a sample of price data between Beijing and Seattle. Based off of this exploration, are the values different between cities? How so? (This is mostly a review as we've done similar work in prior problem sets.) After this hypothesis generation and exploration, we will make an informed decision about transforming the data to ease modeling and justify inferential statistical analysis.

1. load the data *beijing-seattle-airbnb-price.csv*.
2. Perform basic data cleaning checks on city and price. Report the number of observations (total and by city). Are there any missing, invalid, or otherwise suspicious entries?
3. Perform a basic descriptive analysis of the price (overall AND by city).
    - What is the mean, median, mode, standard deviation, and range? Output your results in a single, readable table.
    - Create a single histogram plot with three layers: one for Beijing prices, one for Seattle prices, and one for all. This plot needs to be understood without further comment and must be readable. Hence, it needs a clear title, and axes must be labeled. In addition,

each layer needs to be comparable and easily understood (so a legend will be necessary too.)

Hint: For each layer to be readable it needs a distinct `label` and `color`. `alpha` should also be adjusted to allow overlapping layers to be semi-transparent and readable. Finally, the `bins` for each layer need to be the same to be comparable. In this case, I'd suggest using the code `bins = np.linspace(0,4100, 100)` in your layers so that you have 100 evenly spaced bins covering the total price range in the dataset.

- In your judgment, which city has more expensive AirBnBs? What is the difference of mean prices between Beijing and Seattle?
- Is the price data roughly normal?

At this point, we should see an apparent difference in the mean prices between cities in our sample. Our hypothesis is then that there is a 'real' difference in the prices of AirBnBs between these cities (i.e. H_1 : *difference* $\geqslant 0$). What remains for this problem set is for us to determine if that difference is statistically noteworthy. Our **null hypothesis** in this case is that there is no difference in the mean prices of AirBnB rentals in the underlying population, i.e. 0 (i.e. H_0 : difference $= 0$). If this H_0 is correct, it implies the apparent difference in prices we see in our sample may be just due to sampling–in a limited sample we expect the averages for these two cities to differ just by chance.

---

2 Part Two: Data Transformation (10 points)

In Part Three, we will simulate samples of price data presuming H_0 is correct. This will allow us to get an idea of how unusual the apparent difference we observed in our original sample was due to random sampling variation alone. However, for us to simulate samples in a way that approximates a t-distribution, our data needs to be approximately normally distributed. Since our data is log-normal, not normal, we need to first transform our data so that it does. Luckily, log-normal distributions are transformable to a normal distribution!

We can transform log-normal data to normal by applying a logarithmic function to every value. $X_t = f(x)$, where $f(x)$ can be $\log_{10} x$, $\ln x$, $\log_2 x$, etc. This is then reversible by applying 10^x, e^x, 2^x, etc. to the subsequent values.

Important note! There is no value for x where ANY $\log x = 0$. Hence, you need to make sure no price data is equal to 0 is in your dataset (You can do this by further transforming your data by adding +1, +10, +7 to every value in x).

Interpreting log-transformed data is more difficult! Conversion is not necessarily a simple matter as $\log_{10} x$ - $\log_{10} y$ is NOT equivalent to $\log_{10}(x - y)$! When interpreting results, unless otherwise stated, you may use log-price instead of dollars. Conversion is a tricky matter and will only be requested for select questions.

1. Create a new variable with log-price. Use natural logs, not base-10, base-2, or any other log.
2. Recreate your analysis/visualization of 1.1.2, now using log-price instead of price. What is the shape of the distributions of log-price for Seattle, Beijing, and combined?
3. What is the difference between the mean log-prices of Beijing and Seattle?

   Hint: it should be approximately 0.739

---

3 Part Three: Brute-Force Approach (40 pts)

In this section, we will attempt a 'brute-force' method to determine how likely we are to see a price difference as large as we saw in 1.3, presuming the null hypothesis H_0 is correct. **This is equivalent to asking how likely we are to see a difference in log-prices as large as we observed in 2.3 due to chance!** This key fact will guide us through our brute-force approach.

To answer the question of how likely we are to see a difference in log-prices as large as 2.3 due to chance, we will use simulated new datasets of log-price price data where H_0 is true. This simulated data will mimic our airbnb dataset in numbers of observations (both total and by city). However, we will use a random number generator based off of a normal distribution set to the overall mean and standard deviation of our log-transformed price data. Hence, when we compare the differences in the simulated Beijing and Seattle log-prices, we **know** this variation is due to underlying sampling variability. If we do this enough times, we should get a general idea of how likely the difference in log-prices we observed in 2.3 could be due to chance alone (and hence 1.3 as well)! See also Lecture Notes, Ch 1.5.3 Comparing Distributions.

1. Our null hypothesis H_0 is that the difference in prices in the underlying population of AirBnBs in Beijing is 0. As previously discussed, this is equivalent to the statement that the difference in log-prices of the underlying population is 0. To create a viable distribution where this is the case, let's use the overall mean (μ_0) and standard deviation (σ_0) of the combined Beijing and Seattle log-prices. Please output these values here.

   Hint: the standard deviation is approximately 0.642.

2. Now create two sets of random normals, 'simulatedlogBeijing' and 'simulatedlogSeattle', both using the OVERALL mean μ_0 and standard deviation σ_0 of the entire dataset that you computed as part of your answer to 1.3. The number of observations for each city in your simulated data must be the same as in our original sample. What is the difference between the mean log-prices of these cities? Compare with your results in 2.3.

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

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

## array([4.94137804, 5.8311617 , 4.8655754 , 5.33380209, 4.58650271,
##        5.19472443, 4.21085937, 4.39921104, 4.73806996, 4.58584324,
##        5.40868757, 4.84519601, 4.45653038, 4.43146576, 4.13725904,
##        4.66155043, 4.90113539, 4.12533704, 4.30842263, 6.09261229])

simulatedS

## array([5.28385563, 5.01391037, 4.76576088, 5.27596889, 5.93039557,
##        4.38882754, 4.869364  , 4.60859117, 4.85587573, 4.57966272])
```

   Then, compute the mean difference:

```
np.mean(simulatedB) - np.mean(simulatedS)

## -0.15445502519469656
```

Now compare this number with what you see in the data (in Question 2.3).

3. How much does your simulated difference in mean log-prices reported above equate to in dollars?

   Hint: As stated before, the inverse of 10^x is $\log_{10} x$. However, $\log_{10} x$ - $\log_{10} y$ is NOT equivalent to $\log_{10}(x - y)$!

   Hint: So the suggested way forward would be to convert your simulated log-prices to prices by raising each value from say x to e^x. Then calculate the means of your converted prices by city and find the difference between those mean values.

4. Why do we use the same mean μ_0 and the same variance σ_0 for our simulated Beijing and simulated Seattle?

5. Now repeat simulations in question 3.2 a large number of times (pick, but it needs to be more than 1000) times. Each time store your results (i.e. the simulated log-prices for Beijing and Seattle, as well as the difference in the mean log-prices of the simulated cities.)

6. What is the mean of all the differences between the simulated city mean log-prices? What is the roughly equivalent difference in dollars?

   Hint: The mean value for all the differences recorded in 3.5 probably doesn't correspond exactly to an entry in your simulated dataset from 3.5. Hence, we will substitute with the value which is *closest* to that mean. We can do that by using a function to find the index of that closest value. If we have saved our difference in simulated mean log-prices in an array `fDiff`, then we can find the closest entry to the mean of this array using `idx = np.argmin(np.abs(A - x))`. We can then convert the simulated log-prices for this index to dollars, and find the roughly equivalent difference in dollars.

7. What is the largest difference (in absolute value) of mean log-prices between cities in your sample? What is this difference in standard prices? Hint: `np.abs` computes absolute value.

8. Find the 99% confidence interval (CI) of your sample of the differences of the mean of log-price based on quantiles. Does our observed apparent difference from 1.2.3 fall into that interval?

   Hint: use `np.percentile(0.5)` and a similar expression for the 99.5th percentile.

9. Convert your 99% CI above from log-prices to prices. Does our observed apparent difference from 1.1.2 fall into that interval?

10. Finally, based on the simulations, do you think that our apparent difference in AirBnB prices between cities is due to sample variation? Why?

If you successfully completed the above, congratulations! You have now performed Monte-Carlo estimation. It's delightful how much more prestigious the method sounds with a fancy name instead of 'brute force'. However, I'd suggest reading up and experimenting more with the technique before seriously considering updating your CV! :)

4 Part Four: t-test (30 points)

Above we spent a lot of effort with sampling, random numbers and such. The aim is to get you to understand the underlying 'logic' of how significance tests works. In practice, its usually much easier and more efficient to do a t-test. Below, we ask you to compute a t-test, without using any pre-existing function or package (i.e. writing the equation yourself)!

For the remainder of this PSet, we no longer ask you to convert from log-price differences to price differences. Please be aware of this in the interpretation of the remainder of your responses!

1. Compute standard error (SE) of the difference of mean log-prices between Beijing and Seattle.

   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)

   Hint 2: the answer should be ≈ 0.046.

2. Compute a 99% CI. Use the 1% two-tail significance level to look up the critical values in t-distribution table. OIS has such a table in Appendix C.2, and google can find a million more similar tables. A 99% CI is given by $\mu \pm t_{cr} \cdot SE$ where μ is the mean, SE is its standard error, and 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: Our null hypothesis is $H_0 = 0$, i.e. that there is no difference in prices. Our alternate hypothesis is $H_1 \neq 0$, i.e that there is a difference in mean prices between cities. However, that hypothesis makes no claim on *which* city is larger. Thus we want a two-sided, not one-sided, test.

   Hint 3: You can do this in two ways. Option 1: Compute a 99% CI around H_0 value and check if the apparent difference lies within that CI (this is what we simulated above). Option 2: Compute a 99% CI around the apparent difference between cities, and check of H_0 lies within that CI. DO NOT compute 99% around actual value and then check if the actual value fits in there. It always does!

3. Based on our CI, do you feel comfortable rejecting the null hypotheses H_0?

4. Now perform the opposite operation: compute the t-value of your apparent difference in prices. This gives you one estimate of the likelihood of the value occurring due to sampling variation, and not due to a difference in the underlying data. When you have mean μ and standard error SE, you can compute the t-value by

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

   Hint: answer is 15.98

5. What is the likelihood that such a t-value happens just by random chance? Consult the t-table, you may not get the precise number here, but interpret based on the information on the 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?

5 Part Five: Canned t-test Function (10 points)

Finally, we use a ready-made library: `scipy.stats.ttest_ind` contains ready-made t-test function. Remember: work with log price!

1. 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. Finally, state your conclusion: is Beijing more expensive than Seattle? Do all of your three methods: simulations, 99% CI, t-value and python's t-test agree?

---

6 Challenge (graded as extra credit, 10pt = 1EC point)

How long time do you need to simulate to get the difference in mean log-prices 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. First, time your simulations. Run 3.5 but for a larger number of repetitions, at least seven figures, and measure how long it takes on your computer. Your computer should run a least five seconds before proceeding (this will help with accuracy). Based on that figure, calculate how long it would take to run 10^{12} or so experiments.

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

2. Second, what is the probability to receive such enormous t-values? You need to calculate your t-values yourself, they will not be on any tables. 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
   ```

   Except 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. How many iterations do 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. 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.)
