INFO370 Problem Set: Are students with research experience more likely to be admitted?

January 31, 2022

Instructions

The aim of this problem set is to play with statistical hypotheses, hypothesis testing, and t-test. It follows broadly the approach of lab 3 in the sense that you are first asked to generate random data under H_0, and later the corresponding formulas. 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 college admission data. The dataset is downloaded from Kaggle. It is in places referred to as "UCLA Graduate Dataset", or "inspired by the UCLA Graduate Dataset". It is used in at least two papers, Acharya *et al.* (2019) and Rajagopal (2020), but none of these really describes what is data. So unfortunately we have no idea what do these numbers represent, we cannot even tell you if a line in the data file represents a student, a school, or are these synthetic data. However, the little documentation there is lists the variables as:
Serial No. : case id
GRE Score : (out of 340)
TOEFL Score : (out of 120)
University Rating : (out of 5)
SOP : statement of purpose (out of 5).
LOR : letter of recommendation (out of 5)
CGPA : undergraduate GPA (out of 10)
Research : research experience (0 for none, 1 for experience)
Chance of Admit : admission probability
Note: variable names contain spaces!

But below we imagine that these cases describe individual students, and somehow they they have measured their "admission probability" to actual schools. The task is to analyze if students with research experience are more likely to be admitted. You will only use two variables here, *Research* and *Chance to Admit*.

1 Simulations (50pt)

You will proceed as follows: first, you compute the difference between the average admission chance for students with and without research experience. Thereafter you create two samples of random normal numbers, similar to data above, using the mean and standard deviation over all students (whatever their research experience). Call one of these samples "fake researchers" and the other "fake non-researchers". What is the difference of means of these two groups? And now you repeat this exercise many-many times and see if you can get as big a difference between the fake researchers and fake non-researchers as there is between real researchers and real non-researchers.

1. (2pt) load data *college-admissions.csv*. You only need variables *Research* and *Chance to Admit*. You are welcome to delete all other variables right here.

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

2. (3pt) Describe the admission probability: compute its mean, median, standard deviation, and range. According to these figures, which students are more likely to be admitted–researchers or non-researchers?

3. (3pt) Below, we are going to do t-test. However, t-test works best if the data is normally distributed. Analyze the shape of the distributions for both researchers and non-researchers on a histogram. Comment its shape. Do you think the chances are normally distributed?

4. (5pt) Compute the mean difference in the admission rate between researchers and non-researchers.

   Hint: 0.158

This was the basic description of the data. Now onward to the comparison. We proceed as follows: imagine that there is no real difference in the admission chances for researchers and non-researchers. 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 students' data to test it, but unfortunately we only have what we have. So we do this instead: we create fake researchers' admission chances, and fake non-researchers' admission chances, both drawn from the same normal distribution. There must be as many fake ones as there are real ones in the data. Thereafter we compare the mean chances: how much more likely are the fake researchers to be admitted, compared to the fake non-researchers? 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.

5. (2pt) Let's state our H_0 again: *researchers and non-researchers have similar admission chances (in average)*. Hence we have to create fake researchers and fake non-researchers using the same distribution. The obvious choice for this is the distribution of all students combined.

   Compute the overall mean μ_0 and standard deviation σ_0 of admission chances across all students in the data.

   Hint: standard deviation is 0.143.

6. (5pt) Now create two sets of random normals, "fake researchers" and "fake non-researchers", both with the same mean μ_0 and standard deviation σ_0 that you just computed above. The number of fake students must be the same as the number of real students for the corresponding group.

   What is the difference in the mean admission chances of the fake researchers and fake non-researchers? Compare the result with the real difference you found above.

   Hint: say, the average is 0.5 and standard deviation is 0.2. You can create the corresponding normals like:

```
faker = np.random.normal(0.5, 0.2, size=5)  # create 5 fake researchers
faken = np.random.normal(0.5, 0.2, size=6)  # create 6 fake non-researchers
faker
```

```
## array([0.85772569, 0.58730197, 0.51929949, 0.12730146, 0.44452236])

faken

## array([0.4290482 , 0.4834517 , 0.37459986, 0.49123637, 0.40455639,
##        0.23722705])
```

And you can compute the mean difference like this:

```
np.mean(faker) - np.mean(faken)

## 0.10387693178786389
```

Now compare this number with what you see in data.

7. (3pt) Why do we use the same mean μ_0 for both fake researchers and fake non-researchers?
8. (5pt) Now repeat the previous question a large number R (1000 or more) times. Each time store the mean difference between fake researchers and fake non-researchers, 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, 0.158 in favor of researchers, 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 0.158 just a random fluke, or do students with research experience have better admission chances?

---

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 gather data about millions of students. 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 nonresearchers-researchers 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 t_{cr} values in t-distribution table. OIS has such a table in Appendix C.2, and google image search finds a thousand similar tables.

   95% 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: we need 2-tailed test as nonresearchers can have both better and worse admission chances than researchers.

   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. 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 that nonresearchers and researchers have equal admission chance 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 13.38.

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.

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

   Note: you have to specify `equal_var=False` to tell the function that non-researchers and researchers chances may have different variance.

2. (5pt) Finally, state your conclusion: do researchers have better admission chances than non-researchers?

   Do all of your three methods: simulations, 95% CI, t-value and python's t-test agree?

   Hint: they should!

---

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

How long time do you need to simulate to find a realization in the mean difference in fake non-researchers and fake researchers admission chances that is similar to what you actually see in data, 0.158?

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 run for 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 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. (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.)

References

Acharya, M. S., Armaan, A. and Antony, A. S. (2019) A comparison of regression models for prediction of graduate admissions, in *Second International Conference on Computational Intelligence in Data Science*.

Rajagopal, S. K. P. (2020) Predicting student university admission using logistic regression, *European Journal of Computer Science and Information Technology*, **8**, 46–56.