
Quiz 5 - DSC 10, Spring 2024

Full Name:

PID:

Quiz Time: ☐ 2PM ☐ 2:30PM ☐ 3PM ☐ 3:30PM ☐ 4PM

Instructions:

- This quiz consists of 2 questions. You have a total of 20 minutes to complete it.
 - Please write **clearly** in the provided answer boxes; we will not grade work that appears elsewhere. Completely fill in bubbles and square boxes; if we cannot tell which option(s) you selected, you may lose points.
 - ☐ A bubble means that you should only **select one choice**.
 - ☐ A square box means you should **select all that apply**.
 - If your answer is a string, make sure to put it in quotes. If your answer is a float, make sure to include a decimal point.
 - No aids are allowed (no notes, no calculators, and no computers).
-

By signing below, you are agreeing that you will behave honestly and fairly during and after this quiz.

Signature:

Version A

Please do not open your quiz until instructed to do so.

The DataFrame `bikes` contains a sample of 500 bikes for sale locally. Columns are:

- "serial number" (int): An integer that uniquely identifies the bicycle.
- "price" (int): The list price of the bicycle.
- "style" (str): Values are "electric", "mountain", "road", "hybrid", and "recumbent".

Question 1

You want to know if there is a significant difference in the sale prices of "road" and "hybrid" bikes using a permutation test. The hypotheses are:

- **Null:** The prices of "road" and "hybrid" bikes come from the same distribution.
- **Alt:** On average, "hybrid" bikes are more expensive than "road" bikes.

- a) Using the `bikes` DataFrame and the difference in group means (in the order "road" minus "hybrid") as your test statistic, fill in the blanks so the code below generates 10,000 simulated statistics for the permutation test.

```
def find_diff(df):  
    group_means = df.groupby("shuffled").mean().get("price")  
    return group_means.loc["road"] - group_means.loc["hybrid"]  
  
some_bikes = __ (x) __  
diffs = np.array([])  
for i in np.arange(10000):  
    shuffled_df = some_bikes.assign(shuffled = __ (y) __)  
    diffs = np.append(diffs, find_diff(shuffled_df))
```

(x):

(y):

- b) Do large values of the observed statistic make us lean towards the null or alternative hypothesis?

☐ Null Hypothesis

☐ Alternative Hypothesis

- c) Suppose the p-value for this test evaluates to 0.04. What can you conclude based on this? Select all that apply.

☐ Reject the null hypothesis at a significance level of 0.01.

☐ Fail to reject the null hypothesis at a significance level of 0.01.

☐ Reject the null hypothesis at a significance level of 0.05.

☐ Fail to reject the null hypothesis at a significance level of 0.05

Question 2

You want to determine if the bikes for sale locally have the following distribution of "style": "electric" (15%), "mountain" (20%), "road" (40%), "hybrid" (20%), and "recumbent" (5%). You want to use the 500 rows of `bikes` to test the following hypotheses:

- **Null:** Bikes for sale locally are randomly drawn from the proposed distribution.
- **Alt:** Bikes for sale locally are not randomly drawn from the proposed distribution.

- a) Suppose that in `bikes`, the "style" column is distributed as follows: "electric" (20%), "mountain" (20%), "road" (30%), "hybrid" (20%), and "recumbent" (10%). Let's do a hypothesis test with total variation distance (TVD) as the test statistic. Fill in the blanks below to complete a simulation that calculates 10,000 TVDs under the null hypothesis.

```
def total_variation_distance(distr1, distr2):
    return __ (a) __

proposed = np.array([0.15, 0.20, 0.40, 0.20, 0.05])
observed = np.array([0.20, 0.20, 0.30, 0.20, 0.10])
tvds = np.array([])
for i in np.arange(10000):
    simulated = np.random.multinomial(__ (b) __, __ (c) __) / 500
    new_tvd = total_variation_distance(__ (d) __, __ (e) __)
    tvds = np.append(tvds, new_tvd)
```

(a):

(b):

, (c):

(d):

, (e):

- b) Fill in the blanks below so that `observed_tvd` corresponds to the observed value of the test statistic for this hypothesis test.

```
observed_tvd = total_variation_distance(__ (f) __, __ (g) __)
```

(f):

, (g):

- c) Which of the following correctly calculates the p-value for this hypothesis test?

- ☐ `p_value = np.count_nonzero(tvds >= observed_tvd) / len(observed)`
☐ `p_value = np.count_nonzero(tvds >= observed_tvd) / len(tvds)`
☐ `p_value = np.count_nonzero(tvds <= observed_tvd) / len(observed)`
☐ `p_value = np.count_nonzero(tvds <= observed_tvd) / len(tvds)`

Before submitting your quiz, make sure your PID is on the front page and on the top right corner of page 3.