Data 8 Spring 2019

Foundations of Data Science

SUGGESTED SOLUTIONS

MIDTERM

INSTRUCTIONS

- The exam is worth 100 points. You have 100 minutes to complete it.
- The exam is closed book, closed notes, closed computer/phone/tablet, closed calculator, except the official midterm reference guide provided with the exam.
- Write/mark your answers on the exam in the space/bubbles provided. Answers written anywhere else will not be graded.
- If you need scratch paper, you are welcome to use the reference sheet and the back of this cover page. Scratch work will not be graded.
- For all Python code, you may assume that the statements `from datascience import *` and `import numpy as np` have been executed. Do not use features of the Python language that have not been described in this course.
- In any part, you are free to use any tables, arrays, or functions that have been defined in previous parts of the same question, and you may assume they have been defined correctly.

Last name	SUGGESTED SOLUTIONS
First name	
Student ID number	
CalCentral email (`_@berkeley.edu`)	
Lab GSI	
Name of the person to your left	
Name of the person to your right	
All the work on this exam is my own. **(please sign)**	

This page was intentionally left blank. You can use it for scratch work, but it will **not** be graded.

1. (10 points) Python Expressions

For each of the Python expressions below, write the output when the expression is evaluated. If an error occurs, write Error. Here is an example.

Example Expression: `make_array(1, 2, 3, 4, 5) == 3`

Example Answer: `array([False, False, True, False, False])`

(a) (2 pt) `make_array(1, 1) * np.arange(1, 10, 5)`

array([1, 6])

(b) (2 pt) `make_array(3, 4, 8) + np.arange(2, 7, 1)`

Error

(c) (2 pt) `np.average(np.arange(1, 10, 4))`

5.0

(d) (2 pt) `make_array(1, 2, 3, 4) + 2`

array([3, 4, 5, 6])

(e) (2 pt) `"I love Data " + 8`

Error

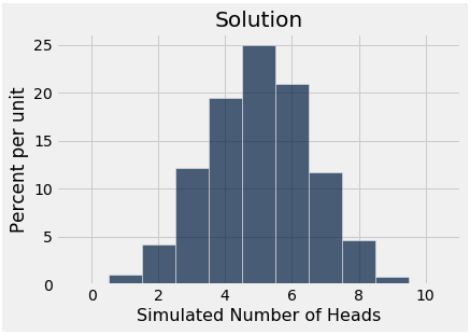
2. (8 points) Statistical Inference and Empirical Distributions

(a) (2 pt) A data scientist performs a statistical test. The null hypothesis is that a specified chance model is good and the alternative hypothesis is that the model is not good. The data scientist decides to use 3% as the cutoff for the P-value of the test.

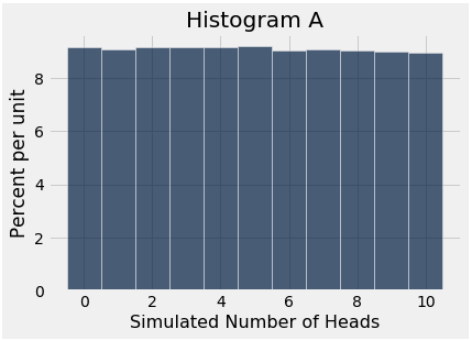
Pick **one** option: If the model is good, the chance that the test will conclude that the data are consistent with the model is:

- ○ 0%
- ○ about 1.5%
- ○ about 3%
- ○ 50%
- ○ about 94%
- ● about 97%
- ○ about 98.5%
- ○ 100%
- ○ not possible to approximate based on the information given

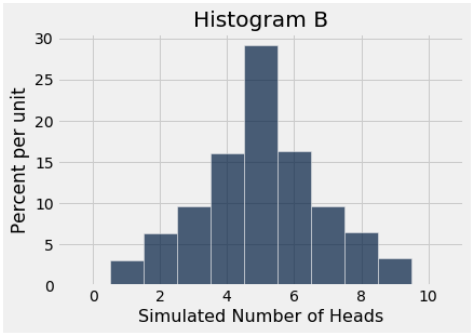
(b) (3 pt) Students in a data science class are asked to draw a histogram of 10,000 simulated values of the number of heads in 10 tosses of a fair coin. The answer in the solution set is below:



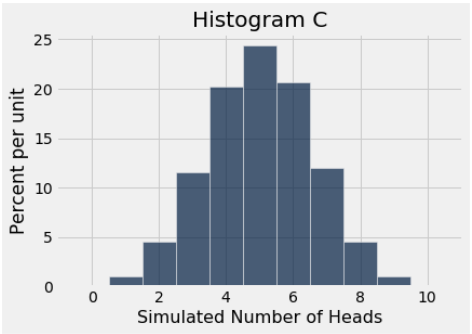
By mistake, a student drew a histogram of 100,000 simulated values instead of 10,000. One of Histograms A, B, and C is the student's answer. Which one is it?



○



○



●

(c) (3 pt) When the following code is run, the expression in the last line evaluates to a number that is approximately one of the values below. Which one?

```
counter = 0
for i in np.arange(100000):
    rolls = np.random.choice(np.arange(1, 7), 10)
    if np.count_nonzero(rolls > 4) == 0:
        counter = counter + 1
counter / 100000
```

○	`1/3`	○	`2/3`
○	`10*(1/3)`	○	`1 - 10*(1/3)`
○	`(1/3)**10`	○	`1 - (1/3)**10`
●	`(2/3)**10`	○	`1 - (2/3)**10`
○	`10*(2/3)`	○	`1 - 10*(2/3)`

3. (20 points) Counties

Every state in the United States is divided into counties that do not overlap with each other and together cover the whole state. A table `counties` contains one row for each county in the United States:

State	County	2010 Pop	2014 Pop
Alabama	Autauga County	54684	55395
Alabama	Baldwin County	183216	200111
Alabama	Barbour County	27336	26887

... (3139 rows omitted)

The table contains four columns:

- **State**: a string, the name of the state
- **County**: a string, the name of the county
- **2010 Pop**: an int, the population in 2010 (as estimated by the US Census Bureau)
- **2014 Pop**: an int, the population in 2014 (as estimated by the US Census Bureau)

In addition, we have the function `first` defined below:

```
def first(x):
    """x is an array"""
    return x.item(0)
```

In each part below, fill in the blanks of the Python expressions. **You must use ONLY the lines provided.** Some of the chained operations we might normally do in one line have been broken up into two or more lines, storing intermediate results in tables or arrays. Do not write any code outside the blanks provided. The expression in the last line should evaluate to the value described in the question.

(a) (2 pt) The name of the largest county in the United States (by 2014 population):

```
sorted = counties.sort("2014 Pop", descending = True)

first(sorted.column("County"))
```

(b) (2 pt) The number of counties in which the population grew by more than 10,000 people between 2010 and 2014:

```
counties_with_change = counties.with_column('Pop Change',

    counties.column("2014 Pop") - counties.column("2010 Pop") )

counties_with_change.where("Pop Change", are.above(10000)).num_rows
```

(c) (2 pt) A new table called `states` that has one row for each state. It should have three columns: the state's name, the total population of the state in 2010, and the total population of the state in 2014. It should not have any column corresponding to county names.

```
counties_3column = counties.drop("County")

states = counties_3column.group("State", sum)

states
```

(d) (4 pt) A new table called `biggest_county` that has one row for each state. Its columns should contain the state's name, the name of the largest county in the state (by 2014 population), the 2010 population of that county, and the 2014 population of that county. It doesn't matter what the column names are.

```
sorted = counties.sort("2014 Pop", descending = True)

biggest_county = sorted.group("State", first)

biggest_county
```

(e) (4 pt) The table `biggest_county` with an additional column called `'Pct in Largest County'`. The column should contain what percent of that state's population lived in the largest county in 2014. For example, if a state has only three counties and the populations in 2014 are 250,000, 200,000, and 50,000, then `'Pct in Largest County'` should be 50.

```
with_states = biggest_county.join("State", states)

biggest_county_with_pct = biggest_county.with_column('Pct in Largest County',

    100 * with_states.column('2014 Pop first') /with_states.column('2014 Pop sum'))

biggest_county_with_pct
```

(f) (6 pt) The table `states` with an additional column called `'Pop in Large Counties'`. The column should contain the total number of people in each state that lived in counties with population more than 100,000 in 2014. For example, if a state has only three counties and the populations in 2014 are 250,000, 200,000, and 50,000, then `'Pop in Large Counties'` should be 450,000. You may assume every state has at least one such county.

```
all_large = counties.where("2014 Pop", are.above(100000))

all_large = all_large.group("State", sum)

all_large = all_large.select('State','2014 Pop sum').relabeled(1, 'Pop in Large Counties')

states_with_large_pop = states.join("State", all_large)

states_with_large_pop
```

4. (25 points) Cereals

A table `cereal` contains one row with nutritional information for each of a set of cereal brands:

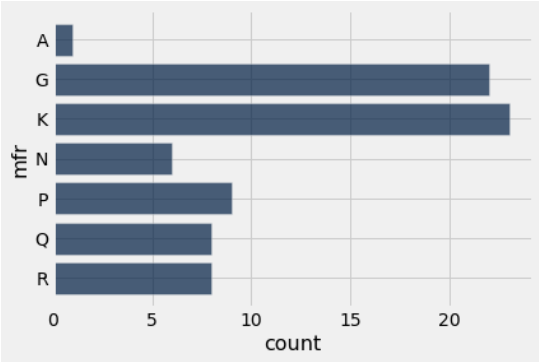
name	mfr	calories	protein	fat	fiber	sugars	shelf	rating
100% Bran	N	70	4	1	10	6	3	68.403
100% Natural Bran	Q	120	3	5	2	8	3	33.9837
All-Bran	K	70	4	1	9	5	3	59.4255

... (74 rows omitted)

The table contains nine columns:

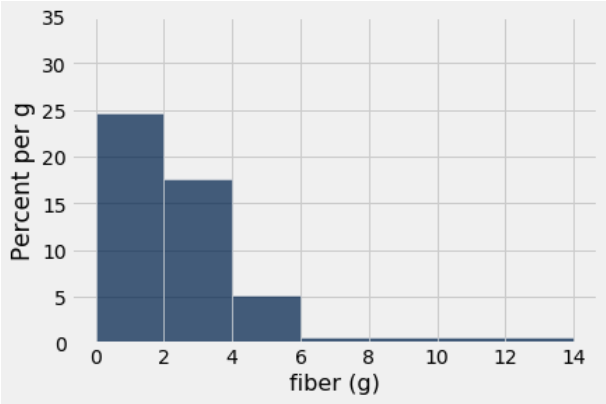
- **name**: a string, the name of the cereal
- **mfr**: a string, the initial of the manufacturer:
  - A: American Home Food Products
  - G: General Mills
  - K: Kelloggs
  - N: Nabisco
  - P: Post
  - Q: Quaker Oats
  - R: Ralston Purina
- **calories**: an int, calories per serving
- **protein**: an int, grams of protein per serving
- **fat**: an int, grams of fat per serving
- **fiber**: an int, grams of fiber per serving
- **sugars**: an int, grams of sugar per serving
- **shelf**: an int, what shelf the cereal is displayed on at a certain supermarket (1, 2, or 3)
- **rating**: a float, giving a rating of the cereal on a scale of 1–100 from Consumer Reports (CR)

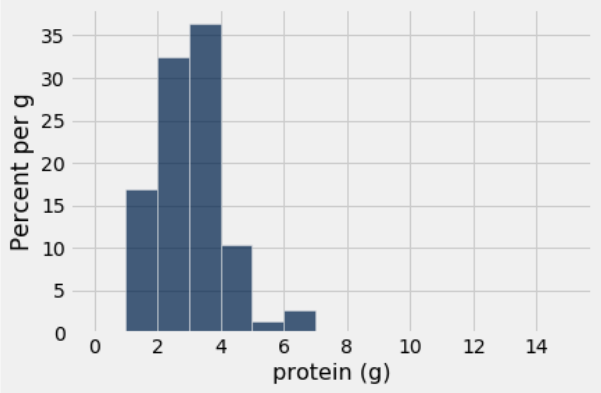
(a) (1 pt) Which of the following lines of code will produce the figure below?



- ○ cereal.hist('mfr')
- ○ cereal.barh('mfr')
- ○ cereal.hist('count')
- ◉ cereal.group('mfr').barh('mfr')
- ○ cereal.group('mfr').plot('mfr')

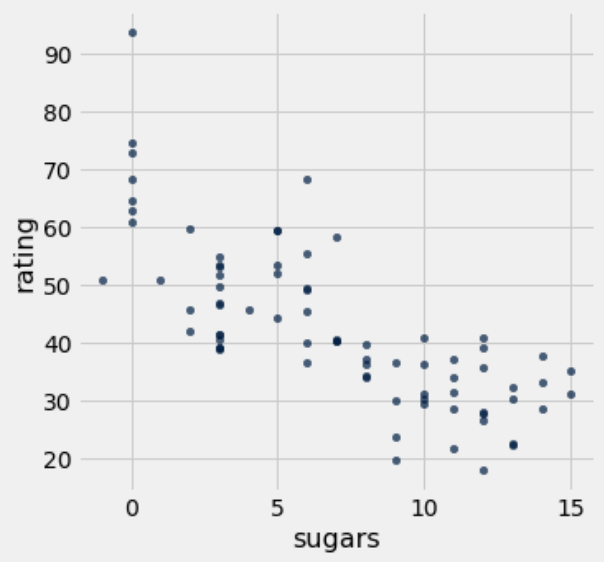
(b) (3 pt) All the cereals are represented in each of the two histograms below. Which conclusion or conclusions are justified by the histograms? Remember that all data are integers. **Bubble in all correct answers.**





- [x] About half the cereals have at least 2 grams of fiber per serving.
- [ ] Most cereals have more grams of protein than grams of fiber per serving.
- [ ] Most cereals have more grams of fiber than grams of protein per serving.
- [x] The number of cereals with at least 4 grams of fiber per serving is about the same as the number of cereals with at least 4 grams of protein per serving.
- [ ] Cereals with more fiber per serving tend to have more protein per serving.

(c) (3 pt) Which conclusion or conclusions are justified by the scatter plot below? **Bubble in all correct answers.**



- [ ] There is an apparent positive association between the amount of sugar per serving and the CR rating.
- [x] There is an apparent negative association between the amount of sugar per serving and the CR rating.
- [ ] There is no apparent association between the amount of sugar per serving and the CR rating.
- [ ] For most cereals, their CR ratings would likely be reduced if the manufacturer added more sugar.
- [ ] For most cereals, their CR ratings would likely be increased if the manufacturer added more sugar.
- [ ] For most cereals, their CR ratings would likely remain unchanged if the manufacturer added more sugar.

(d) **(4 pt)** Suppose we want to predict the CR rating for a new cereal based on its amount of sugar, by averaging the ratings of all cereals in `cereal` whose sugar content per serving differs from that of the new cereal by no more than 1 gram. Fill in the blanks below to define a function called `predict_rating` that takes in an amount of sugar in grams and returns the prediction for the CR rating.

```
def predict_rating(sugar):
    comparison_group = cereal.where("sugars", are.between(sugar - 1, sugar + 2))
    return np.average(comparison_group.column("rating"))
```

(e) **(4 pt)** Based on Parts (a) through (d), in approximately which range will the predicted rating be for a cereal with 1 gram of sugar per serving? You may assume `predict_rating` has been correctly defined.

- ○ between 30 and 35
- ○ between 40 and 45
- ○ between 50 and 55
- ● between 60 and 65

(f) **(4 pt)** Fill in the blanks below to add a new column to the table `cereal` containing the prediction from the `predict_rating` function for the `sugars` value in each row. The column should be called `'Predicted Rating'`.

```
predictions = cereal.apply(predict_rating, "sugars")
cereal = cereal.with_column("Predicted Rating", predictions)
```

(g) **(2 pt)** Next, we want to see how well our predictions are doing on average. Write a Python expression that evaluates to the average absolute error for all of our predictions. The absolute error is the absolute value of the difference between the predicted rating and the actual rating.

```
np.average(np.abs(cereal.column("Predicted Rating") - cereal.column("rating")))
```

(h) **(4 pt)** Complete the Python expression that evaluates to the table below. The table contains the average number of calories per serving in cereal brands grouped by manufacturer **and** shelf.

```
cereal.pivot("mrf", "shelf", "calories", np.average)
```

shelf	A	G	K	N	P	Q	R
1	0	106.667	107.5	86.6667	105	100	102.5
2	100	111.429	111.429	95	110	113.333	0
3	0	114.444	107.5	70	110	80	127.5

5. (12 points) Grabbing Socks

Professor Fithian stays up late laundering his socks and sleeps in one day when he is going to give his Data 8 lecture. While running out the door in a state of panic, he grabs two socks completely at random from the dryer, which contains 28 total socks: 16 black socks, 10 white socks, and 2 lime green socks.

The socks are not in pairs and haven't been touched since they were "shuffled" by the dryer the night before, so the two socks are like two draws at random without replacement.

Find the following probabilities. **Show your work!** Your answers should be math calculations, not Python code, but **you do not have to simplify any arithmetic**.

(a) (3 pt) The probability that both of the socks are black:

P(both of the socks are black) = (16/28) * (15/27)

(b) (3 pt) The probability that at least one sock is lime green:

P(at least one sock is lime green) = 1 - (26/28) * (25/27)

(c) (3 pt) The probability that one sock is black and one is white:

P(one sock is black and one is white) = 2 * (16/28) * (10/27)

(d) (3 pt) The probability that the two socks **are not** the same color:

P(two socks are not the same color) = 1 - [(16/28) * (15/27) + (10/28) * (9/27) + (2/28) * (1/27)]

6. (20 points) Matching Socks

After lecture, while trying to reassure Prof. Fithian about his mismatched socks (see Question 5), Prof. Adhikari tells him that the same thing happens to her all the time. In fact, she claims that every time she gives a lecture there is a 25% chance that her socks will be mismatched, regardless of whether they match on any other day.

You overhear this conversation and you suspect that she is just exaggerating to make Prof. Fithian feel better; that is, you believe her socks are mismatched less frequently than she claims. You decide to put her claim to the test by watching videos of the last 100 Data 8 lectures she has given.

In all of the questions below, a *mismatched lecture* is a lecture in which the professor's socks don't match, and a *matched lecture* is a lecture in which the professor's socks do match.

(a) (4 pt) Which option or options below are statements of the null hypothesis? **Bubble in all answers that are correct** (or that can be correct after blanks are filled in appropriately), and fill in all blanks for the answers you bubble in. **Do not** fill in blanks for answers you do not bubble in.

○ In 100 lectures, Prof. Adhikari is expected to wear mismatched socks about _____ times,

and any deviation from that is _____.

○ In any given lecture, Prof. Adhikari's socks are just as likely to be mismatched as they are to be matched, and any difference between the number of matched lectures and mismatched lectures

is _____.

○ Prof. Adhikari is just as likely to sleep in on the day of a lecture as Prof. Fithian is, and any difference between the number of times the two professors sleep in is

_____.

● If we label each of Prof. Adhikari's lectures as mismatched or matched, then 100 lectures will be like a random sample drawn __with__ replacement from a distribution that assigns a __25__ % chance to a lecture being mismatched and a __75__ % chance to being matched.

○ Prof. Adhikari wears mismatched socks as often as Prof. Fithian wears mismatched socks, and any difference in the fraction of mismatched lectures between the two professors

is _____.

○ In 100 past lectures of Data 8, Prof. Adhikari wore mismatched socks exactly _____ times.

(b) (4 pt) One of the choices below is the correct alternative hypothesis based on Prof. Adhikari's claim and what you believe. Bubble it in and fill in its blank.

○ Prof. Adhikari is less likely than Prof. Fithian

to _____.

○ Prof. Adhikari and Prof. Fithian are not equally likely

to _____.

● The probability Prof. Adhikari wears mismatched socks in a lecture is less than 25%.

○ The probability Prof. Adhikari wears mismatched socks in a lecture is more than _____.

○ The probability Prof. Adhikari wears mismatched socks in a lecture is different from _____.

(c) (2 pt) You watch the videos of Prof. Adhikari's last 100 lectures, and carefully count how many times her socks were mismatched. Which statistic should be used to decide between the two hypotheses?

● The number of lectures with mismatched socks

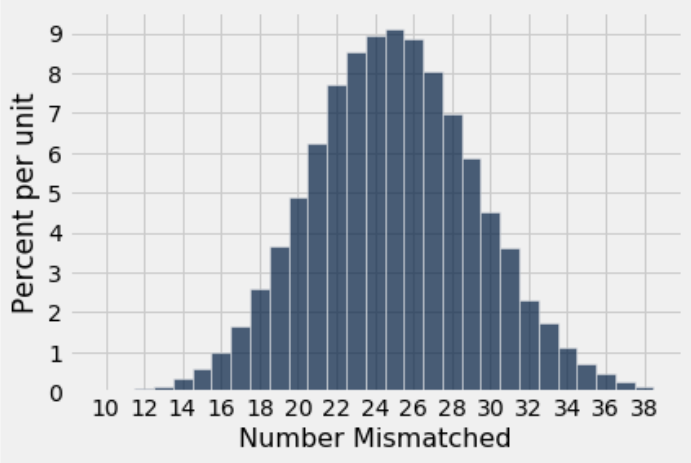
○ The distance between the number of lectures with mismatched socks and 25

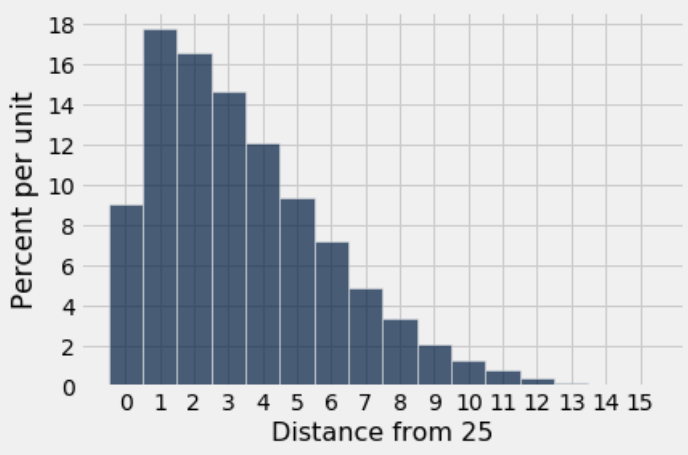
(d) (5 pt) Fill in the missing Python code below to simulate the value of the test statistic under the null hypothesis 10,000 times. The last line should evaluate to an array of 10,000 simulated values.

```
def sim_statistic():
""" Returns one value of the test statistic simulated under the null"""

    sim_proportion_under_null = sample_proportions(100, make_array(0.25, 0.75)).item(0)

    return sim_proportion_under_null * 100

statistics = make_array()

## simulate 10,000 values of the test statistic:

for i in np.arange(10000):

    new_statistic = sim_statistic()

    statistics = np.append(statistics, new_statistic)


statistics
```

(e) (3 pt)

The two histograms printed below show the simulated distributions of (left figure) the number of mismatched lectures and (right figure) the distance between 25 and the number of mismatched lectures. Each simulation is carried out under the null hypothesis, with 50,000 simulation runs of 100 lectures each. Remember that in both histograms, all of the simulated values are integers. The bins in both figures all have width 1 and are centered at integers.





In the 100 videos, Prof. Adhikari's socks are mismatched 16 times. Based on the figures above, which answer is closest to the P-value for the hypothesis test?

○ 0% ○ 1% ◉ 2% ○ 4% ○ 10%

(f) (2 pt)

If you use a 3% cutoff for the P-value, does the test favor your belief over Prof. Adhikari's claim?

○ No ◉ Yes

7. (5 points) Blood Pressure

In a randomized controlled experiment, 300 patients are randomized into the treatment group (Group A) and 200 to the control group (Group B). At the end of the experiment, the blood pressure of each patient is measured. The research team wants to test the null hypothesis that the treatment has no effect versus the alternative hypothesis that the treatment has an effect. As their test statistic they decide to use the absolute difference between the average blood pressures of Group A and Group B.

The table `data` has 500 rows, one for each patient in the experiment. The table has just one column. The column is labeled `'bp'` for blood pressure, and contains a numerical measurement of blood pressure for each patient. All measurements are in the same units.

The table `data` has no other information. If possible, fill in the blanks in the code below so that the last line evaluates to one value of the test statistic simulated under the null hypothesis. If this is not possible, pick the option below the code and give a brief explanation.

● This is possible based on the information given, using the following code:

```
shuffled = data.sample(with_replacement = False)

# Two tables

group_A_shuffled_table = shuffled.take(np.arange(300))

group_B_shuffled_table = shuffled.take(np.arange(300, 500))

# Two averages

shuffled_mean_A = np.average(group_A_shuffled_table.column('bp'))

shuffled_mean_B = np.average(group_B_shuffled_table.column('bp'))

# Test statistic

abs(shuffled_mean_A - shuffled_mean_B)
```

○ This is not possible with the information given. Explain below:

8. (0 points) **W**rite your name in the space provided on one side of every page of the exam. You're done!