Data 8 Midterm Solutions

Summer 2017

First Name: ______________________________

Last Name: ______________________________

Email: ______________________________@berkeley.edu

Student ID: ______________________________

GSI: ______________________________

Name of person to your left: ______________________________

Name of person to your right: ______________________________

All the work on this exam is my own. **(please sign)**: ______________________________

Instructions:

- This exam contains 13 pages (including this cover page) and 4 questions.
- This exam must be completed in the **1:50 time** period ending at **12:00PM**.
- You may use a single handwritten (two-sided) cheat sheet and the two official study guides provided with this exam.
- Work quickly through each question. There are a total of 91 points on this exam.

This page intentionally left blank.

1 Counting Calories

1. (14 Pts) The `calories` table has a list of all food items from McDonald's and contains the following columns:

   - **Item**: A string, the name of the item.
   - **Size**: An int, the serving size of the item in grams.
   - **Cal**: An int, the number of calories the item has.
   - **FatCal**: An int, the number of calories from fat the item has. The number of calories from fat is always less than or equal to the number of calories.

`calories`:

Item	Size	Cal	FatCal
Egg McMuffin	136	300	120
Baked Apple Pie	77	250	110
McRib	208	500	240
Daily Double	190	430	200

`... (257 rows omitted)`

The `categories` table has a list of all food items from McDonald's and contains the following columns:

- **Item**: A string, the name of the item. The items in this column are the same as in `calories` but are not in the same order.
- **Category**: A string, the category of the item.

`categories`:

Item	Category
Egg McMuffin	Breakfast
McChicken	Chicken & Fish
Premium Southwest Salad (without Chicken)	Salads
Baked Apple Pie	Desserts

`... (257 rows omitted)`

This problem is continued on the next page.

Fill in the blanks of the Python expressions to compute the described values. You must use **all** and **only** the lines provided. The last (or only) line of each answer should evaluate to the value described. Assume that the statements `from datascience import *` and `import numpy as np` have been executed.

(a) (2 Pts) The total number of calories of all items on the McDonald's menu.

```
______________(calories.______________(___________________))
```

Solution:

```
sum(calories.column('Calories'))
```

(b) (3 Pts) The average serving size of all items with over 200 calories from fat.

```
______________(

  calories.__________(__________________, __________________)

    .______________(_____________________)
)
```

Solution:

```
np.mean(calories.where('FatCal', are.above(200))
                .column('Size'))
```

(c) (3 Pts) The serving size of the item with the largest serving size of all breakfast menu items. Use the provided `mcdonalds` table.

```
mcdonalds = calories.join('Item', categories)

______________(

  mcdonalds.__________(__________________, __________________)

    .______________(_____________________)
)
```

Solution:

```
mcdonalds = calories.join('Item', categories)
np.max(mcdonalds.where('Category', 'Breakfast')
        .column('Size'))
```

(d) (6 Pts) We have access to the following `menu` table which is a simpler combined version of the `calories` and `categories` tables. It just contains each food item, the number of calories, and the category for the item:

Item	Cal	Category
Egg McMuffin	300	Breakfast
Baked Apple Pie	250	Desserts
McRib	500	Beef & Pork
Daily Double	430	Beef & Pork

```
...  (257 rows omitted)
```

Which of the following Python expressions evaluates to the number of desserts with less than 200 calories? **Select all that apply.**

✓
```
(menu.where('Category', are.equal_to('Desserts'))
 .where('Calories', are.below(200))
 .num_rows)
```

✓
```
(menu.where('Calories', are.below(200))
 .where('Category', are.equal_to('Desserts'))
 .num_rows)
```

☐
```
(menu
 .with_column('Below200', menu.column('Calories') < 200)
 .pivot('Category', 'Below200')
 .num_rows)
```

☐
```
(menu
 .with_column('Below200', menu.column('Calories') < 200)
 .pivot('Category', 'Below200')
 .where('Below200', are.equal_to(True))
 .num_rows)
```

☐
```
(menu.group('Category')
 .where('Calories', are.below(200))
 .where('Category', are.equal_to('Desserts'))
 .column('count').item(0))
```

✓
```
(menu.where('Calories', are.below(200))
 .group('Category')
 .where('Category', are.equal_to('Desserts'))
 .column('count').item(0))
```

2 License Samples

2. (14 Pts) The table `license` contains the `name` and `age` of every person who got a new driver's license from the Berkeley DMV in 2016, sorted by `age` in descending order.

name	age
Arvind	49
Nanxi	38
Emma	25

```
...  (347 rows omitted)
```

Sample 1: `license.take(np.arange(30, 100))`
Sample 2: `license.take(np.arange(0, license.num_rows, 10))`
Sample 3:

```
start = np.random.choice(np.arange(10))
license.take(np.arange(start, license.num_rows, 10))
```

(a) (2 Pts) Select all true statements about the three samples.

- [✓ Sample 1 is a deterministic sample.]
- ☐ Sample 2 is a probability sample.
- ☐ Samples 2 and 3 are systematic samples.
- ☐ All three samples are taken at random without replacement.

(b) (2 Pts) What is the probability that Arvind is included in Sample 2? Assume that each name only appears once in the table. Select one choice.

- ○ 0
- ○ $\frac{1}{10}$
- ○ $\frac{1}{2}$
- ○ $\frac{1}{3}$
- [✓ 1]

(c) (2 Pts) What is the probability that Arvind is included in Sample 3? Assume that each name only appears once in the table. Select one choice.

- ○ 0
- [✓ $\frac{1}{10}$]
- ○ $\frac{1}{2}$
- ○ $\frac{1}{3}$
- ○ 1

(d) (2 Pts) Fred calculates the average age of all the people in sample 1. Select all choices that accurately complete the sentence. This value is:

- ☐ the population parameter.
- ✓ a statistic.
- ☐ an unbiased estimate of the population parameter.
- ✓ a biased estimate of the population parameter.

(e) (6 Pts) Fred writes the following code to take samples from `license`. For each of the questions in this part, select one choice.

```
num_trials = 1
mystery = make_array()
for i in np.arange(200):
    something = np.mean(
        license.sample(100, with_replacement=False)
               .column('age')
    )
    mystery = np.append(mystery, something)
```

i. How many samples did Fred take with this code?

- ◯ 1 sample.
- ◯ 100 samples.
- ✓ 200 samples.
- ◯ Can't tell from this code alone.

ii. How large was each sample?

- ◯ 1 person.
- ✓ 100 people.
- ◯ 200 people.
- ◯ Can't tell from this code alone.

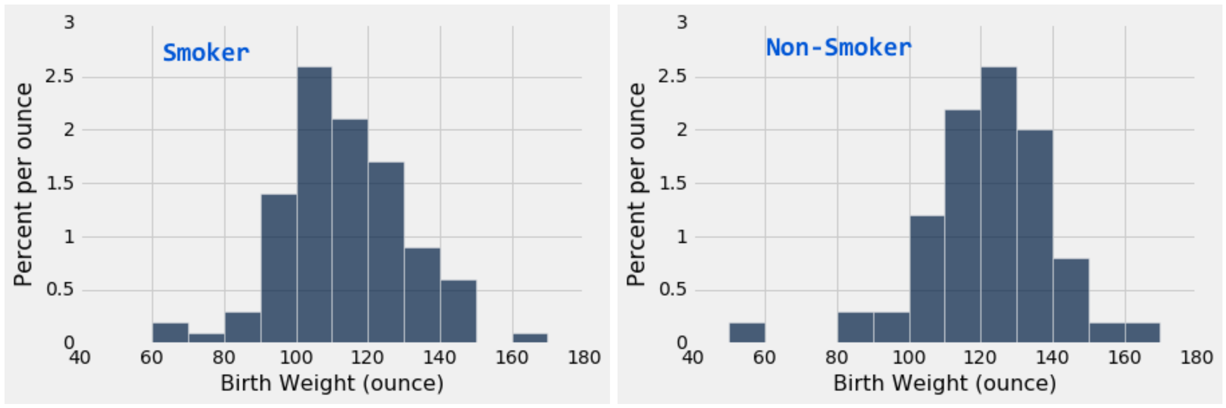
iii. At the end of this code, what does the `mystery` array contain?

- ✓ Unbiased estimates of the population mean.
- ◯ Biased estimates of the population mean.
- ◯ Unbiased estimates of Sample 2's mean.
- ◯ Biased estimates of Sample 2's mean.

3 Smoking Weights

3. (13 Pts) 600 women were surveyed after giving birth. Each woman was asked whether she regularly smoked and the weight of their newborns were recorded. There were 100 smokers and 500 non-smokers among those surveyed. The histograms below visualize the babies' weights of these two groups of women.

   Both histograms have bins that are multiples of 10. The bar heights of both histograms are multiples of 0.1 percent per ounce.



   (a) (8 Pts) For each pair of quantities, select one choice for the correct statement.
   You must briefly justify your answer in the space provided to receive full credit. When possible, use a mathematical justification.

   i. (I) The **proportion** of weights between 100 (inclusive) and 110 (exclusive) for smokers vs. (II) The **proportion** of weights between 100 (inclusive) and 110 (exclusive) for non-smokers.

      [✓ (I) is larger.]
      ◯ (II) is larger.
      ◯ (I) and (II) are roughly equal.
      ◯ There is not enough information to compare (I) and (II).

      Solution: $10 \times 2.6 > 10 \times 1.2$

ii. (I) The **number** of newborn weights between 80 (inclusive) and 90 (exclusive) for smokers vs. (II) The **number** of newborn weights between 80 (inclusive) and 90 (exclusive) for non-smokers.

- ◯ (I) is larger.
- ✓ (II) is larger.
- ◯ (I) and (II) are roughly equal.
- ◯ There is not enough information to compare (I) and (II).

Solution: B; $10 \times 0.3 \times 100 < 10 \times 0.3 \times 500$

iii. (I) The **proportion** of newborn weights under 100 (exclusive) for smokers vs. (II) The **proportion** of newborn weights above 130 (inclusive) for non-smokers.

- ◯ (I) is larger.
- ✓ (II) is larger.
- ◯ (I) and (II) are roughly equal.
- ◯ There is not enough information to compare (I) and (II).

Solution: B; $10(0.2 + 0.3 + 0.3) < 10(0.9 + 0.6 + 0.1)$

iv. (I) The **number** of newborn weights between 100 (inclusive) and 115 (exclusive) for smokers vs. (II) The **number** of newborn weights between 115 (inclusive) and 130 (exclusive) for smokers.

- ◯ (I) is larger.
- ◯ (II) is larger.
- ◯ (I) and (II) are roughly equal.
- ✓ There is not enough information to compare (I) and (II).

Solution: D; we don't know the distribution within a bin.

(b) (3 Pts) What proportion of all babies (with smoking and non-smoking mothers) were between 140 and 150 ounces? You may leave your answer as a mathematical expression.

Solution:

$$\frac{10 \times 0.8 \times 500 + 10 \times 0.6 \times 100}{600} \approx 7.7\%$$

4 Cookie Questions

4. Cookie Monster owns a cookie factory. His factory produces cookie flavors in the following proportions:

Sugar	Chocolate Chip	Trash
0.2	0.6	0.2

Sam is unhappy that his box of 10 cookies had 4 Trash-flavored cookies in it. However, he doesn't remember whether he got his cookies from Cookie Monster's factory or another factory. You decide to run a hypothesis test to figure out whether Sam's box came from Cookie Monster's factory or not.

(a) (3 Pts) Select the correct option from each set of choices to state an appropriate null hypothesis for this problem.

Null hypothesis: Sam's box [✓ did] ○ did not come from Cookie Monster's factory.

Thus, the chance that any one cookie in Sam's box was Trash-flavored is
[✓ 0.2.] ○ computable but not 0.2. ○ unknown given this information.

Because [✓ of random chance,] ○ the box didn't come from Cookie Monster's factory, Sam got a higher number of Trash-flavored cookies in his box.

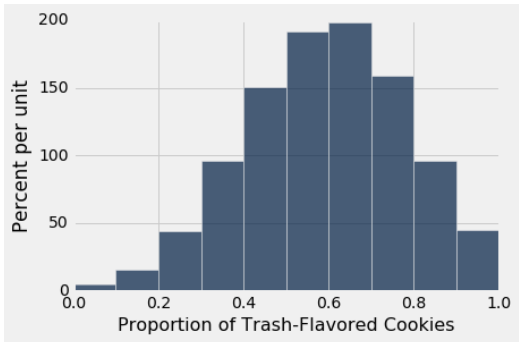
(b) (4 Pts) As a test statistic, you choose to compare **the proportion** of Trash-flavored cookies in boxes of 10 cookies. Complete the lines below to compute a single test statistic using a random sample from the population, storing the value into `test_stat`. You may write expressions as long as you need to on each line, but you may not use extra lines.

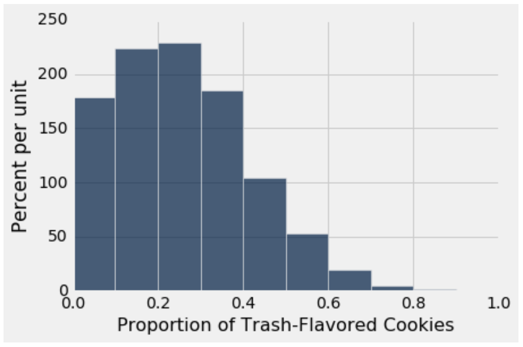
```
outcomes = make_array(

    ________________________________________________

    ________________________________________________
)

sample = np.random.choice(outcomes, ______________)

test_stat = ______________________________________
```

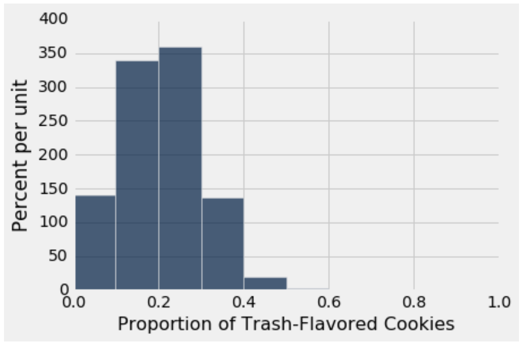
Solution:

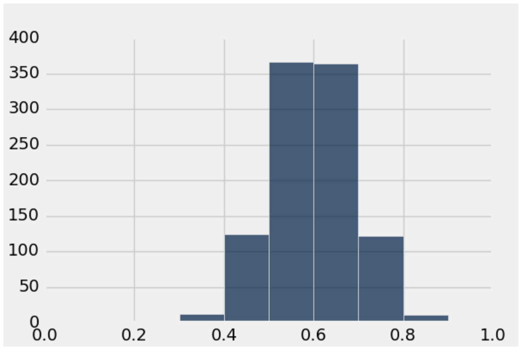
```
outcomes = make_array('Trash', 'Not Trash', 'Not Trash',
                      'Not Trash', 'Not Trash')
sample = np.random.choice(outcomes, 10)
test_stat = np.count_nonzero(sample == 'Trash') / 10
```

(c) (4 Pts) Given that your final computed p-value is less than 0.05, select all histograms corresponding to the possible probability distributions of the test statistic.

☐


☐


✓


☐


Briefly justify your answer below:

Solution: The bottom-left choice is the only correct choice.

The probability distribution for the statistic should be centered at the population proportion of Trash-flavored cookies, 0.2.

Of the histograms centered at 0.2, only the bottom-left has less than 5% of its area above 0.4, our sample statistic.