Data 8 Foundations of Data Science
Fall 2017

MIDTERM SOLUTIONS

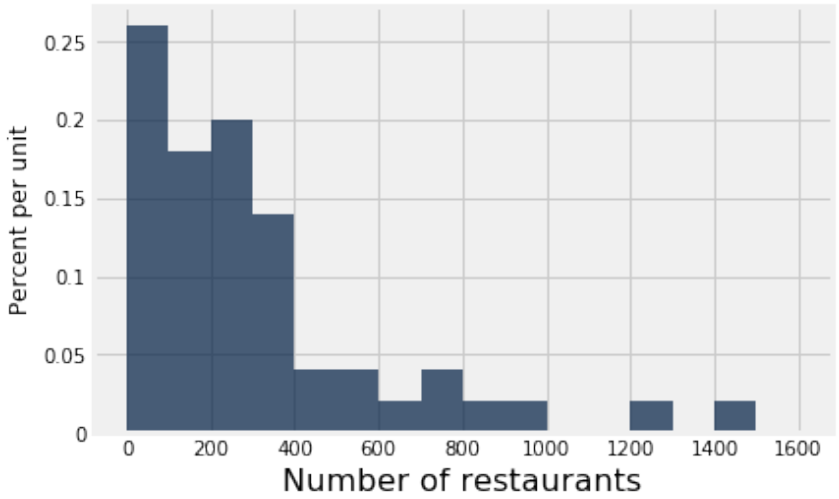
INSTRUCTIONS

- You have 45 minutes to complete the exam.
- The exam is closed book, closed notes, closed computer, closed calculator, except one hand-written 8.5" × 11" sheet of notes of your own creation and the official midterm exam reference guide provided with the exam.
- Mark your answers **on the exam itself**. We will *not* grade answers written on scratch paper.

Last name	
First name	
Student ID number	
BearFacts email (`_@berkeley.edu`)	
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)**	

1. (12 points) Distributions

We counted the number of McDonald's restaurants in each of the 50 US states, and plotted below a histogram of these 50 counts. **All bars are 100 wide.** The areas of the bars sum to 100%.



Calculate each quantity described below or write *Unknown* if there is not enough information above to express the quantity as a single number (not a range). It's OK to leave answers unsimplified. Show your work.

(a) (2 pt) The **percentage** of states that have fewer than 100 McDonald's restaurants.

$0.26 \times 100 = 26\%$ *Note:* You can tell in advance that an odd percentage is not right, because with 50 states and bar widths of 100, each state will raise its bar by 2%.

(b) (2 pt) The **number** of states that have fewer than 100 McDonald's restaurants.

$0.26 \times 50 = 13$ *Note:* You can tell in advance that a fractional answer like 12.5 is not right, because each state either has fewer than 100 McDonald's restaurants or it doesn't. A number of states must be a whole number.

(c) (2 pt) The **number** of states that have fewer than 150 McDonald's restaurants.

Unknown (because we don't know how the states with 100-199 McDonald's are distributed within the bar—they might all be 100, or might all be 199)

(d) (4 pt) The **number** of states that have 200 or more McDonald's restaurants.

$50 \times (1 - 0.26 - 0.18) = 28$ *Note:* You can tell in advance that a fractional answer like 27.5 is not right, because each state either has 200+ McDonald's restaurants or it doesn't. A number of states must be a whole number.

(e) (2 pt) The **number** of states that have 2000 or more McDonald's restaurants.

0 (because the histogram includes all 50 counts and the area sums to 100%)

2. (18 points) Expressions

A table named `seat` contains a row for each time a student submitted the attendance form in lecture on September 18th, 20th, or 22nd. The table contains four columns.

- **Email**: a string, the email address of the student
- **Row**: a string, the letter of the row in which they claim to be seated
- **Seat**: an int, the number of the seat in which they claim to be seated
- **Date**: an int, the date of the submission, either 18, 20, or 22.

A second table `roster` contains a row for each student enrolled in Data 8. It has two columns: the **Email** column contains student emails and the **Name** column contains student names. Both columns contain strings.

Email	Row	Seat	Date
sulu@berkeley.edu	C	102	20
mccoy@berkeley.edu	A	3	18
kirk@berkeley.edu	R	110	20

... (1747 rows omitted)

Email	Name
kirk@berkeley.edu	James
uhura@berkeley.edu	Nyota
sulu@berkeley.edu	Hikaru

... (1023 rows omitted)

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.
You may not use features of the Python language that have not been described in this course.
Assume that the statements `from datascience import *` and `import numpy as np` have been executed.

(a) (2 pt) The largest seat number in the `seat` table.

```
max(seat.column('Seat'))
```
or
```
seat.sort('Seat', descending=True).column('Seat').item(0)
```

(b) (4 pt) The row letter that appears most often in the `Row` column of the `seat` table.

```
t = seat.group('Row').sort('count', descending=True)


t.column('Row').item(0)
```

(c) (4 pt) The total number of attendance submissions for September 20th in rows A, B, C, D, or E.

```
u = seat.where('Row', are.below('F'))


u.where('Date', 20).num_rows
```

(d) (4 pt) The name of the student who submitted attendance the most times. (Assume no ties.)
Note: Every student has one unique email address, but two students might have the same name.

```
v = seat.group('Email').join('Email', roster)


v.sort('count', descending=True).column('Name').item(0)
```

(e) (4 pt) The total number of repeat submissions in the `seat` table (those that are *not* the first submission from a student on a particular day).

```
seat.num_rows - seat.group(['Email', 'Date']).num_rows
```
or
```
sum(seat.group(['Email', 'Date']).column('count') - 1)
```

3. (10 points) Probability

(a) (4 pt) We roll two fair six-sided dice. If the sum is not 7, we re-roll just the second die (once). After rolling the two dice and possibly re-rolling the second, what is the probability that the sum of the two dice is now 7? It's OK to leave your answer unsimplified. *Examples*: **(A)** We roll 2 and 5: the sum is 2+5=7. **(B)** We roll 1 and 4, so we re-roll the second die and it comes up 6: the sum is 1+6=7.

$1/6 + 5/6 \times 1/6 = 11/36$: The chance of totaling 7 the first time (6 of 36) + the chance of not totaling 7 the first time (30 of 36) and then totaling 7 the second time (1 of 6).

or

$1 - (5/6 * 5/6) = 11/36$: The chance that the following does not happen: Not totaling 7 the first time (30 of 36) and not totaling 7 the second time either (5 of 6).

(b) (6 pt) Implement the `estimate` function, which returns an estimate of this probability generated by simulating the above process repeatedly, `trials` times, and returning the fraction of trials where the sum is 7. In each trial, we simulate rolling two fair six-sided dice and re-rolling the second if necessary.

```
def estimate(trials):
    sevens = 0
    faces = np.arange(1, 7)

    for i in np.arange(trials):
        first = np.random.choice(faces)
        second = np.random.choice(faces)

        if first + second != 7:
            second = np.random.choice(faces)

        if first + second == 7:
            sevens = sevens + 1

    return sevens / trials
```

4. (5 points) Causality

In 2013, a Berkeley chemical engineering professor, Jay Keasling, discovered a synthetic version of artemisinin — a chemical crucial to producing antimalarial drugs. A group of researchers and statisticians come together to test the effectiveness of this new chemical, which is being used to treat malaria in developing nations.

They run the following *experiment*: A random sample of 500 malaria patients are randomly assigned to take either the antimalarial drug or a placebo pill. It turns out that among the patients who took the drug, a much larger proportion recovered from malaria than among those who took the placebo.

Meanwhile, cases of malaria are decreasing in developing nations, and some suggest this might be due to the increased effectiveness of mosquito nets being distributed.

Circle True or False for each of the following statements. Don't justify your answer.

(a) (1 pt) True or False: This is a randomized controlled experiment.

(b) (2 pt) True or False: The experiment indicates that taking the drug *causes* an increase in recovery rates among malaria patients.

(c) (2 pt) True or False: The improvements in mosquito nets are a confounding factor in this experiment.