Data 8 Summer 2018 — Foundations of Data Science

MIDTERM SOLUTIONS

INSTRUCTIONS

- You have 50 minutes to complete the exam.
- The exam is closed book, closed notes, closed computer, closed calculator, except for the official reference guide provided with the exam.
- Mark your answers **on the exam itself**. We will *not* grade answers written on scratch paper.
- You may leave numerical calculations unsimplified throughout the exam.
- You may assume that the statements `import numpy as np` and `from datascience import *` have been executed throughout the exam.

Last name	
First name	
Student ID number	
CalCentral email (`_@berkeley.edu`)	
GSI name and Lab Time	
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)**	

Question 0 (1 point) Write your name and SID (Student ID Number) in the space provided on one side of every page of the exam.

1. (10 points) Potpourri

For the first two independent subproblems, assume you are drawing with replacement from a bucket of tickets. There are 12 tickets; 3 of which are red, 5 of which are blue, 1 of which is green, and the rest are yellow.

(a) (2 pt) You draw from the bucket 40 times, as specified above. **Mark all** of the Python expressions which evaluate to the probability of picking a green ticket at least once in 40 draws.

- ◯ `1 - ((11/12) ** 40)`
- ◯ `(1/12) ** 40`
- ◯ `1 - ((1/12) ** 40)`
- ◯ `(11/12) ** 40`
- ◯ None of the above

Option 1

(b) (2 pt) This time, you only draw up to two tickets. If your first ticket is red, you do not draw again. If your first ticket is not red, you put your ticket back and draw again one more time. **Mark all** of the Python expressions below which evaluate to the probability of picking a red ticket.

- ◯ `1/4 + (3/4 * 1/4)`
- ◯ `2 * (3/4 * 1/4)`
- ◯ `(1/4) ** 2`
- ◯ `3/16`
- ◯ None of the above

Option 1

(c) (2 pt) Mark the following statement as **True** or **False**.

Specifying a control group and a treatment group as part of an experiment is enough to establish causality through the experiment.

- ◯ True
- ◯ False

False

(d) (2 pt) Which of the following statements about randomization are true? **Mark all that apply.**

- ◯ Assigning individuals randomly to treatment and control groups ensures that the two groups are likely to be similar (besides the treatment).
- ◯ Randomization ensures that individuals don't know whether they are in the treatment group or the control group.
- ◯ Randomization ensures that experimenters don't know whether their participants are in the treatment group or the control group.
- ◯ Randomization in an experiment intends to eliminate any potential confounding factors.
- ◯ None of the above

1, 4

(e) (2 pt) Which of the following statements are true? **Mark all that apply.**

- ◯ A parameter does not change depending on the values in a sample.
- ◯ The population average is a statistic.
- ◯ A statistic does not change depending on the values in a sample.
- ◯ A `for` loop can be used to create the probability distribution of a statistic. The purpose of this simulation is to be used as an approximation for the empirical distribution.
- ◯ None of the above

Option 1

2. (11 points) World Cup

The table `soccer` contains one row for each international soccer game that has ever been played before the 2018 World Cup. Each game has several attributes as shown below in the first few rows of the table.

year	home_team	away_team	home_score	away_score	tournament	home_team_result
1872	Scotland	England	0	0	Friendly	Draw
1873	England	Scotland	4	2	Friendly	Win
1874	Scotland	England	2	1	Friendly	Win
1875	England	Scotland	2	2	Friendly	Draw
1876	Scotland	England	3	0	Friendly	Win
1876	Scotland	Wales	4	0	Friendly	Win

In the table above, assume that the entries in the columns **year**, **home_score** and **away_score** are of type `int`, while the rest are of type `string`. The only values found in the **home_team_result** column are 'Draw', 'Win', and 'Loss'.

Fill in the blanks of the Python expressions to compute the described values. You must use only the lines provided to get full credit. The last line of each answer should evaluate to the value described. Assume that the statements `import numpy as np` and `from datascience import *` have been executed. You may enter anything you would like in the blanks below, but you may not add code outside of the blanks.

(a) (2 pt) The total number of goals that team 'USA' has scored when they were the home team.

```
united_states = ____________________________________________________________________

____________________________________________________________________________________

united_states = soccer.where('home_team', 'USA')

sum(united_states.column('home_score'))
```

(b) (2 pt) The total number of games the country 'Belgium' has played, both home and away.

```
____________________________________________________________________________________

soccer.where(1, 'Belgium').num_rows + soccer.where(2, 'Belgium').num_rows
```

(c) (3 pt) The best away team; i.e. the away team which has won the most games.

```
winners = soccer.where(____________________________________________________________)

____________________________________________________________________________________

winners = soccer.where('home_team_result', 'Loss')

winners.group('away_team').sort(1, descending = True).column(0).item(0)
```

(d) (4 pt) The away team against whom Brazil has the most wins, when Brazil is playing at home. To help, we have started off a function `num_wins`, which takes in an array of strings and returns the number of times the word 'Win' occurs. This should be useful for your total expression.

```
def num_wins(arr):

       return ____________________________________________________________________________

wins = soccer.pivot(_____________________________________________________________________)

wins.select('away_team', _____________).____________________________________________
```

```
def num_wins(arr)

    return sum(arr == 'Win')

wins = soccer.pivot('home_team', 'away_team', 'home_team_result', num_wins):

wins.select('away_team', 'Brazil').sort('Brazil', descending = True).column(0).item(0)
```

3. (8 points) Fun with Functions

Define a function `simulate_fn`, which takes in the following arguments:

- `tbl`: A one column table of numbers representing the population we want to sample from
- `sample_size`: The number of items we want to take in a sample (without replacement)
- `fn`: A function which takes in an array of numbers and returns back one value
- `reps`: The number of times we want to simulate taking a sample

The function `simulate_fn` should simulate taking `reps` number of samples of size `sample_size` from `tbl`. Each time you take a sample, keep track of the value of the function `fn` applied to the numbers in the sample. `simulate_fn` should return the number of these values that are greater than the smallest number in `tbl`.

```
def simulate_fn(tbl, sample_size, fn, reps):

        vals = ____________________________________________________________________


        for i in ____________________________________________________________________:


                samp = ____________________________________________________________


                val = _____________________________________________________________


                vals = ____________________________________________________________


        return ____________________________________________________________________
```

```
def simulate_fn(tbl, sample_size, fn, reps):
        vals = make_array()

        for i in np.arange(reps):

                samp = tbl.sample(sample_size, with_replacement = False)

                val = fn(samp.column(0))

                vals = np.append(vals, val)

        return sum(vals >= min(tbl.column(0)))
```

4. (20 points) Sampling and Visualizations

We have a distribution of 200 individuals' heights and weights in a sample from a population of size 2000. This sample is found in the `hw_200` table.

(a) (2 pt) Assume we label each individual in the population with a label from 1 to 2000. Which of the following sampling techniques will generate a probability sample? **Mark all that apply.**

- ◯ Taking every 10th individual from the population, starting with individual 1
- ◯ Taking every 10th individual from the population, starting with some individual in range 1 to 10 randomly with equal chance
- ◯ Shuffling all the labels of individuals and picking the new individuals labeled 1 to 200
- ◯ Shuffling all the labels of individuals and picking the original individuals labeled 1 to 200
- ◯ None of the above

Options 2 and 3

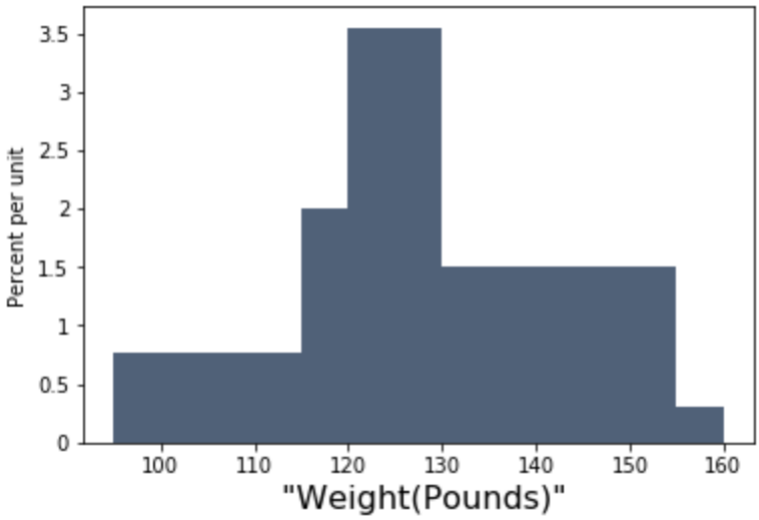
(b) (2 pt) Which of the following visualizations should be used to properly view if there is an association between heights and weights of individuals? **Mark all that apply**.

- ◯ Line Graph
- ◯ Bar Graph
- ◯ Scatter Plot
- ◯ Histogram
- ◯ None of the above

Option 3

We decide to discard the heights from our sample and only look at a distribution of weights. This distribution is visualized with the histogram below using the following code:

```
hw_200.hist('Weight(Pounds)', bins = make_array(95, 115, 120, 130, 155, 160))
```

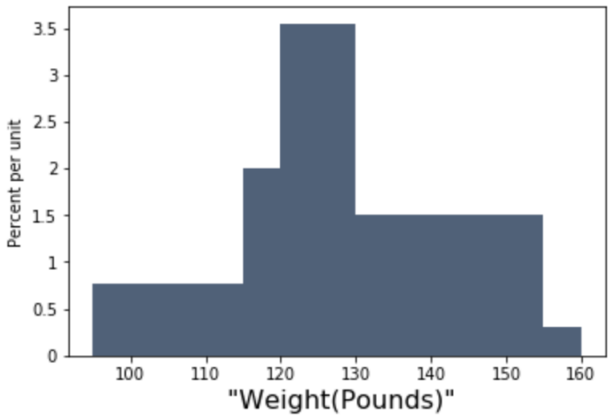


(c) (2 pt) Which of the following most accurately describes the distribution above, with respect to the population of all 2,000 weights?

- ◯ Deterministic distribution
- ◯ Empirical distribution
- ◯ Probability distribution
- ◯ Normal distribution
- ◯ None of the above

Option 2

For the following questions, choose the option corresponding to the **larger value**. If you cannot identify an answer using the given information, choose the "Need more information" option. Assume the intervals are inclusive on the lower end and exclusive on the upper end. The histogram is repeated for your convenience.



(d) **(2 pt)** Option A: The percentage of individuals in the sample weighing between 95 lbs and 115 lbs
Option B: The percentage of individuals in the sample weighing between 115 lbs and 120 lbs

- ◯ Option A
- ◯ Option B
- ◯ Need more information

Option A

(e) **(2 pt)** Option A: The percentage of individuals in the sample weighing between 95 lbs and 120 lbs
Option B: The percentage of individuals in the sample weighing between 120 lbs and 130 lbs

- ◯ Option A
- ◯ Option B
- ◯ Need more information

Option B

(f) **(3 pt)** Option A: 3%
Option B: The percentage of individuals in the sample weighing between 115 lbs and 117.5 lbs

- ◯ Option A
- ◯ Option B
- ◯ Need more information

Need more information

(g) **(3 pt)** Option A: 20%
Option B: The percentage of individuals in the sample weighing between 115 lbs and 117.5 lbs

- ⬭ Option A
- ⬭ Option B
- ⬭ Need more information

Option A

(h) **(4 pt)** We decide to break up the 120-130 bin, which has height 3.5, into two bins of equal width. Once divided, we see that the height of the 120-125 bin is twice as large as the height of the 125-130 bin. If possible, solve for the height of the 120-125 bin. If not possible, explain what more information you might need to solve this problem. Show your work and circle or box your final answer. You may leave your final answer as an arithmetic expression.

Total area should be 35. We know that the width of both is equal, and we know that area of 1 plus area of 2 is 35. The area of both is some height multiplied by 5, so we divide 5 on both sides. the sum of the two heights should be 7. One of the heights is twice as large as the other heights, so we get the larger height is 14/3

5. (20 points) Video Game Distributions

The table `vg` displays a sample of racing video games and shooter video games along with their global sales (in millions of dollars) in 2016. The first four rows are shown below:

Genre	Global_Sales
Racing	35.82
Shooter	28.31
Racing	23.42
Racing	14.98

How does the genre of a video game affect its global sales in 2016? Using the `vg` sample, we would like to know whether or not the population distribution of global sales for all racing video games in 2016 is the same as the population distribution of global sales for all shooter video games in 2016.

To begin with, we are told that the two do come from the same population distribution. However, we would like to test that claim as we actually believe the two come from different population distributions (Note: We don't know what the two different distributions look like, we just think that the distributions are different in some way).

(a) (4 pt) State the null and alternative hypotheses that should be used to answer this question.

Null:

Alternative:

Null: The underlying distribution from which the population of global sales for the shooting genre came from is the same underlying distribution from which the global sales for the racing genre was picked from. [The distributions in the sample are different due to chance.]

Alternative: The underlying distributions of global sales of the two genres are different.The difference in our sample is due to something other than chance.

After we formulate our two hypotheses, we calculate our sample average global sales for our two video game genres in 2016:

Genre	Global_Sales mean
Racing	0.586101
Shooter	0.791885

(b) (1 pt) Which of the following lines of code could have produced the table above? Mark all that apply.

- ◯ `vg.group('Global_Sales', np.mean)`
- ◯ `vg.group('Genre', np.mean)`
- ◯ `vg.group('Genre')`
- ◯ `vg.pivot('Global_Sales', 'Genre', np.mean)`
- ◯ None of the above

Option 1

(c) **(2 pt)** Which of the following is the best choice for a test statistic for our experiment?

- ◯ The average global sales for the shooter genre minus the average global sales for the racing genre
- ◯ The absolute value of the difference between average global sales for the shooter genre and the average global sales for the racing genre
- ◯ The TVD between the racing genre and the shooting genre
- ◯ None of the above

Option 2

(d) **(3 pt)** In order to continue, we need to view the distribution of the test statistic under the null hypothesis. Finish the lines of code below to create a table that simulates taking one sample under the assumption of the null hypothesis, but does not compute the test statistic.

```
shuffled = vg._____________________________________________________________________________

Table.with_column(_____________________________________________________________________________)
```

```
shuffled = vg.sample(with_replacement=False).column(1)

Table().with_column('Genre', vg.column(0), 'Shuffled GS', shuffled)
```

After simulating the test statistic under the null hypothesis and comparing it with our observed test statistic, we calculate a P-Value of .04. Assume we use a P-Value cutoff of .05 for the following questions.

Mark the following statements as **True** or **False**.

(e) **(2 pt)** There is a 4% chance that the null hypothesis is true.

- ◯ True
- ◯ False

False

(f) **(2 pt)** If the null hypothesis is true, we have a 5% chance of incorrectly stating that our data is more consistent with the alternative hypothesis using this method of testing.

- ◯ True
- ◯ False

True

(g) **(2 pt)** In general, we have a 5% chance of incorrectly stating the data is more consistent with the null hypothesis using this method of testing.

- ◯ True
- ◯ False

False

(h) **(2 pt)** Based on our analysis, we conclude that our observed data is more consistent with the alternative hypothesis.

- ◯ True
- ◯ False

True

(i) (2 pt) Based on our analysis, we can conclude that the genre of a video game causes a difference in average global sales in 2016.

- ⬭ True
- ⬭ False

False