Data 8 Spring 2017 - Midterm 1 Review Worksheet II

I. Tables

You have two tables, animals and types. Use the two tables to answer the questions.

animals

Class	Famous?	Name	Specific Type
Asteroidea	True	Patrick Star	Starfish
Mammalia	True	Simba	Lion
Mammalia	False	McGruff	Bloodhound
Reptilia	True	Geico Gecko	Day Gecko

(87 more rows)

types

Specific Type	Weight (lbs)	Color
Starfish	0.125	Pink
Lion	500	Gold
Orca	2000	Multicolored

(102 more rows)

Write the code that you would use to show the following:

```
joined = animals.join("Specific Type", types)
```

1. What is the name of the heaviest animal in the animal table?

```
joined.sort("Weight (lbs)",
descending=True).column("Name").item(0)
```

2. Create a new column for types with the weight in kilograms (2.2 pounds = 1 kilogram).

```
types.with_column("Weight (kg)", types.column("Weight (lbs)") *
1 / 2.2)
```

3. Figure out the average weight for each class in the animals table.

```
joined.group("Class", np.mean)
```

4. Create an array of the types of animals that are multicolored.

```
types.where("Color", "Multicolored").column("Specific Type")
```

5. True or False: Patrick Star is heavier than the Geico Gecko.

```
joined.where("Name", "Patrick Star").column("Weight (lbs)").item(0) > joined.where("Name", "Geico Gecko").column("Weight (lbs)").item(0)
```

6. How many animals in the animals table are in the reptilia class?

```
animals.group('Class').where("Class", "Reptilia").column("Count").item(0)
```

OR

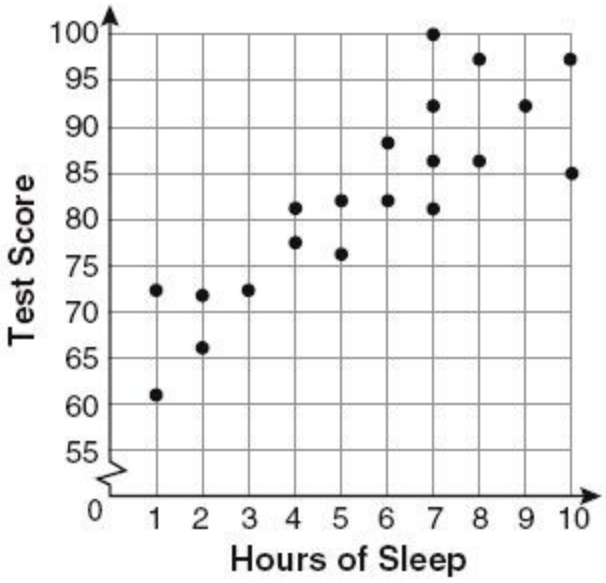
```
animals.where("Class", "Reptilia").num_rows
```

7. Create a table that has the counts for how many famous and not famous animals there are in each class for animals.

```
animals.group(["Class", "Famous"])
```

II. Plot

Look at the following scatter plot. What can you say about the relationship between the amount of sleep that a student gets the night before a test and their score?

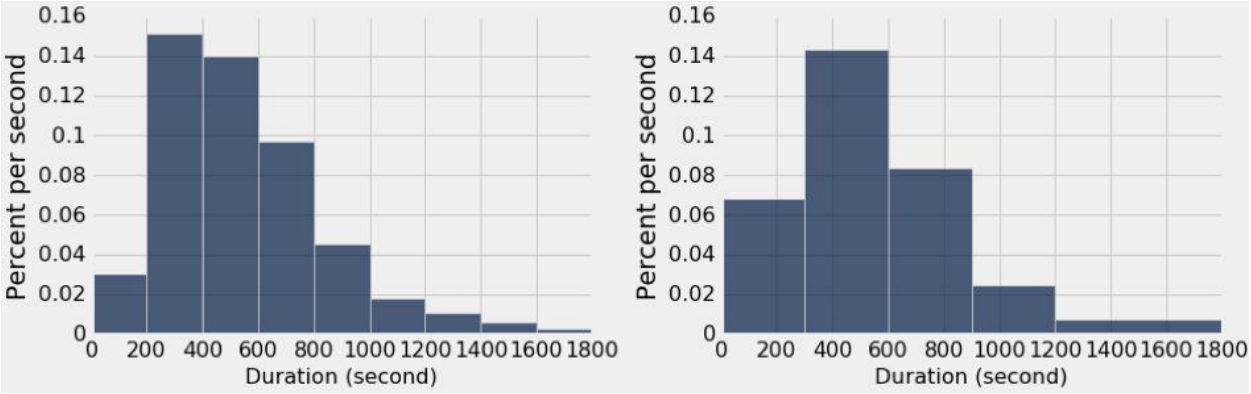


We can say that there is a positive association between someone's test score and the hours of sleep that they got. Based purely on this graph, with no information about how the data came to be, we can not say if there is a causal relationship.

III. Histograms

(From Spring 2016 Midterm)

The two histograms of bike trip durations below were both generated by trip.hist(...) using different bins (same data, different bins).



Write the proportion of trips that fall into each range of durations below. Show your work. If it is not possible to tell from the histograms, instead write NOT ENOUGH INFO.

1. Between 200 (inclusive) and 400 (exclusive) seconds

   `0.0015 * 200 == 0.30 or 30%`

2. Between 300 (inclusive) and 900 (exclusive) seconds
   `0.0014 * 300 + 0.0008 * 300 == 0.66 or 66%`

3. Between 400 (inclusive) and 900 (exclusive) seconds
   `0.0014 * 200 + 0.0008 * 300 == 0.52 or 52%`

4. Between 200 (inclusive) and 300 (exclusive) seconds
   `0.0015 * 200 + 0.0014 * 200 - 0.0014 * 300 == 0.16 or 16%`
   `Alternatively: 0.0007 * 300 - 0.0003 * 200 == 0.15 or 15%`

IV. Functions

1. Define a function that when given an array of numbers, prints all of the even numbers in the array.

```
def print_evens(some_array):
    for element in some_array:
        if element % 2 == 0:
            print(element)
```

2. Define a function that when given a table and a column name, returns an array of all the different possible values in that column.

```
def unique_column_values(column_name, table):
    return table.group(column_name).column(column_name)
```

3. Using a for loop, simulate sampling randomly 1000 times a number from 0 to 10 and output the percentage of number 7.

```
num_simulations = 5000
def sample():
    return np.random.choice(np.arange(11))
def simulate():
    total = 0
    seven = 0
    for i in np.arange(num_simulations):
        if sample() == 7:
            seven = seven + 1
        total = total + 1
    return seven / total
```

4. Fill in the blanks so that the following code outputs 2017.

```
big_data = make_array(987654321, 924645644, 54500654, 8506006)
```

```
# Note that 987654321 - 924645644 - 54500654 - 8506006 = 2017

difference = 0
for index in np.arange( len(big_data) ):
    difference = abs(big_data.item( index) - difference)
print(difference)
```

V. Probabilities

(From Fall 2016 Probability Review)

We draw 2 tickets at random, **with replacement**, from a box with the tickets described below.

Number of Tickets	**Red**	**Blue**	**Green**
Smooth	20	15	15
Jagged	30	15	5

1. What is the probability that both tickets are green?
   $(\frac{20}{100})^2$
2. What is the probability that both tickets are the same color?
   $(\frac{20}{100})^2 + (\frac{30}{100})^2 + (\frac{50}{100})^2$
   we're adding together P(both green), P(both blue), and P(both red)
3. What is the probability that both tickets are blue and jagged?
   $(\frac{15}{100})^2$

(From Spring 2016 Midterm)
A study followed 369 people with cardiovascular disease, randomly selected from hospital patients. A year later, those who owned a dog were four times more likely to be alive than those who didn't.

1. Circle True or False: This study is a randomized controlled experiment.
   Answer: False; the experimenters did not control who owned a dog.
2. Circle True or False: This study shows that dog owners live longer than cat owners on average.
   Answer: False; the experiment compares those who owned a dog to those who didn't (no mention of cats)
3. Circle True or False: This study shows that for someone with cardiovascular disease, adopting a dog will probably cause them to live longer.

Answer: False; an observational study does not show causation.