S&DS 123 YData: An Introduction to Data Science
Spring 2020 MIDTERM

**INSTRUCTIONS**

- You have 50 minutes to complete the exam.
- The exam is closed book, closed notes, closed computer, closed phone, and closed calculator, except one hand-written 8.5" × 11" crib sheet of your own creation (both sides) and the official study guide provided with the exam.
- Mark your answers **on the exam itself**. We will *not* grade answers written on scratch paper.

| | |
|---|---|
| First name | |
| Last name (Surname) | |
| NetID | |

| Problem | Score | Out of |
|---|---|---|
| 1 | | 2 |
| 2 | | 2 |
| 3 | | 3 |
| 4 | | 8 |
| 5 | | 4 |
| 6 | | 9 |
| 7 | | 2 |
| Total | | 30 |

1. **(2 points) Sampling**

   Which of the following is an example of a random sample of students from our YData course? Circle *all* that apply.

   (a) All students who attended lectures in the first week of class

   (b) The first 20 students in the lecture hall on a random day of class

   (c) All juniors in the class

   (d) 50 students randomly selected from the course roster Answer

   (e) None of the above

2. **(2 points) Variables**

   A study is conducted on students taking a data science class. Several variables are recorded in the survey. Circle *all* the categorical variables.

   (a) Type of shoes the student wears. Answer

   (b) Current GPA of the student.

   (c) The time the student waited in line at the bookstore to pay for their textbooks.

   (d) Home state or country of the student. Answer

   (e) The hair color of the student. Answer

   (f) Whether or not the student took AP statistics in high school. Answer

   (g) None of the above.

3. **(3 points) Probability**

   Fred plays a dart throwing game at the state fair. The chance that he hits the bullseye on a given attempt is 0.65, and the attempts do not impact each other. Suppose he makes six attempts. What is the probability that he succeeds in hitting the bullseye in at least one but no more than five of the six attempts?

   Give your answer using Python expressions. You do not need to give the number value of the probability.
   1 - 0.65**6 - (1-0.65)**6

4. **(8 points) Tables**

(a) This semester a survey was administered to collect data on students in our YData class. A subset of the results are contained in a table named `survey`, with five rows displayed below.

Five rows of the `survey` table:

| year | sleep | tv | studying | hand | season |
|---|---|---|---|---|---|
| First Year | 7.5 | 3 | 20 | right | Spring |
| Graduate Student | 7 | 2 | 15 | left | Spring |
| Sophomore | 5 | 6 | 35 | right | Spring |
| nan | 6.5 | 20 | 30 | right | Spring |
| First Year | 7 | 3 | 20 | right | Fall |

The table has the following columns:
`year`: year in school
`sleep`: average number of hours of sleep per night
`tv`: number of hours spent watching TV during a typical week
`studying`: number of hours spent studying during a typical week
`hand`: the preferred hand
`season`: the favorite season

**Provide the Python expressions to compute the values described below.** You can assume the statements `from datascience import *` and `import numpy as np` have been executed. Be sure to report the value that is requested, with no additional information returned.

- (1 pt) The number of rows in the `survey` table. survey.num_rows

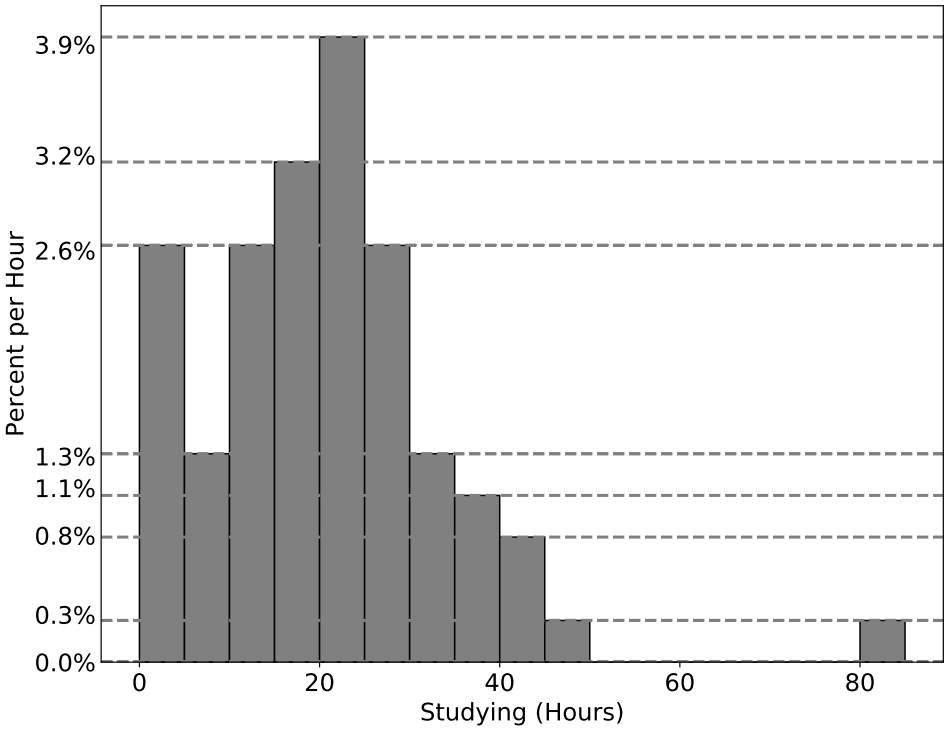
- (2 pts) The overall average of the average number of hours graduate students (`Graduate Student`) sleep each night.
np.average(survey.where('year','Graduate Student').column('sleep'))

- (2 pts) The typical number of hours of tv watched by the Sophomore student who studies the most and is right-handed (assuming no ties).
survey.where('year','Sophomore').where('hand','right').sort('studying',descending=True).column('tv').item(0)

- (3 pts) Create a table that displays the maximum number of hours spent studying in a typical week by year in school. Not all survey responders reported their year in school, so be sure to remove the 'nan' values (indicating missing values) from the `year` column.
survey.select('year','studying').where('year',are.not_equal_to('nan')).group('year',max)

5. **(4 points) Distributions**

Using the `survey` table from an earlier question on this exam, suppose we decide to look at the number of hours spent studying during a typical week (i.e., the `studying` column) in more detail. A histogram of the `studying` column is displayed below using the bins defined by `bins = np.arange(0, 90,5)`.




Calculate the quantities specified below using values displayed on the histogram, or write "Unknown" if there is not enough information to express the quantity as a single number (not a range). Be sure to show your work.

- (2 pts) What percentage of students study fewer than 40 hours?
  100 - 5*(.3+.3+.8) = 100-7% = 93%

- (2 pts) If a new bin was defined as $[60, 85)$, what would the height of that bar be?
  5*.3/(85-60) = 0.06

6. **(9 points) Hypothesis testing.**

Four different versions of a midterm exam (Exam 0, Exam 1, Exam 2, Exam 3) were randomly given to students in a data science class with 200 students. Each version of the exam had 50 students take it. The professor claimed that the exams all had the same level of difficulty. However, the average score for the 50 students who took Exam 0 was lower than those who took the other versions of the exam. The students think that Exam 0 must have been more difficult.

The first few rows of the table with the exam scores, `scores`, is displayed below along with the average scores for each of the four exam versions.

First 6 (of 200) rows of table `scores`:

| Exam | Scores |
|---|---|
| 0 | 66 |
| 1 | 94 |
| 0 | 86 |
| 1 | 83 |
| 3 | 84 |
| 2 | 90 |

`scores.group('Exam', np.average)`

| Exam | Scores average |
|---|---|
| 0 | 69 |
| 1 | 81 |
| 2 | 81 |
| 3 | 78 |

See the Demo from lecture on 2/24 (Lecture 18) for similar problem.

- (2 pt) Which of the following are appropriate null and alternative hypotheses to test the claim of the students?

  (a) Null: Exam 0 was more difficult than the other exams
  Alternative: Exam 0 was the same level of difficulty as the other exams

  (b) Null: Exam 0 was the same level of difficulty as the other exams Answer
  Alternative: Exam 0 was more difficult than the other exams

  (c) Null: Exam 0 was the same level of difficulty as the other exams
  Alternative: Exam 0 was easier than the other exams

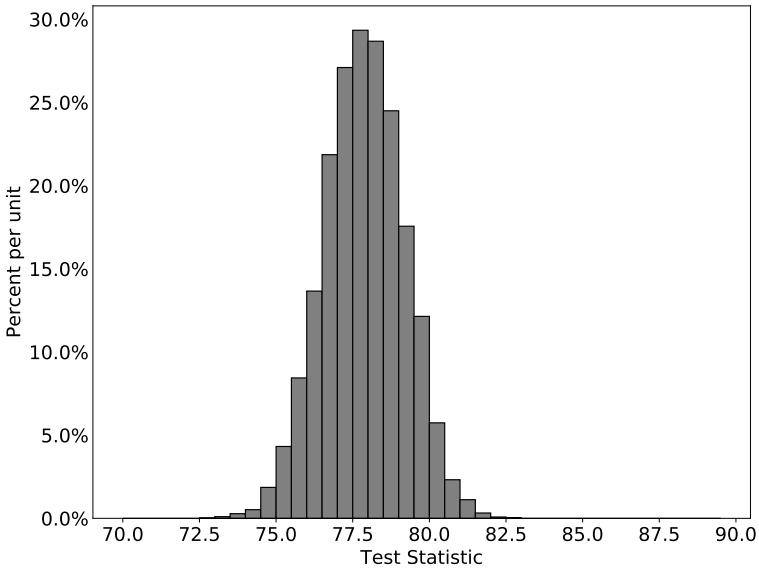
  (d) None of the above.

- (5 pts) A simulation is carried out by generating 10,000 values of the test statistic in the usual way to test the null hypothesis. Fill in the blanks (a) - (e) below with Python code for generating a sample of the test statistics under the assumption that the null hypothesis is true.

```
statistics = (a)_____

for i in (b)_____:

  random_sample = scores.sample((c)_____, with_replacement = False)

  new_statistic = np.average(random_sample.column((d)_____))

  statistics = np.append(statistics, (e)_____)
```

(a) `make_array()`, (b) `np.arange(10000)`, (c) `50`, (d) `'Scores'`, (e) `new_statistic`

- (2 pts) Suppose the simulation of the test statistic under the null hypothesis resulted in the following histogram. Which of the following is the correct conclusion using a p-value cutoff of 5%? Circle your response.




(a) The observed value of the statistic is in the middle of the histogram so we can reject the null hypothesis.

(b) The observed value of the statistic is in the middle of the histogram so we cannot reject the null hypothesis.

(c) The observed value of the statistic is so far in the right tail of histogram that we can reject the null hypothesis.

(d) The observed value of the statistic is so far in the right tail of histogram that we cannot reject the null hypothesis.

(e) The observed value of the statistic is so far in the left tail of histogram that we can reject the null hypothesis. Answer

(f) The observed value of the statistic is so far in the left tail of histogram that we cannot reject the null hypothesis.

(g) None of the above.

7. **(2 points) Expressions**

Recall the `nba` table used in some of our lectures, with the first several rows displayed below. The columns list the names of the players from the 2015-2016 season, their position, team, and salary (in millions of dollars).

First five rows of `nba`:

| PLAYER | POSITION | TEAM | SALARY |
|---|---|---|---|
| Paul Millsap | PF | Atlanta Hawks | 18.6717 |
| Al Horford | C | Atlanta Hawks | 12 |
| Tiago Splitter | C | Atlanta Hawks | 9.75625 |
| Jeff Teague | PG | Atlanta Hawks | 8 |
| Kyle Korver | SG | Atlanta Hawks | 5.74648 |

As we did in lecture, assume the "starter" for a team and position is the player with the highest salary on that team in that position. The following table displays the starters for each team and position during the 2015-2016 season.

First five rows of desired table of starters for each team:

| TEAM | C | PF | PG | SF | SG |
|---|---|---|---|---|---|
| Atlanta Hawks | Al Horford | Paul Millsap | Jeff Teague | Thabo Sefolosha | Kyle Korver |
| Boston Celtics | Tyler Zeller | Jonas Jerebko | Avery Bradley | Jae Crowder | Evan Turner |
| Brooklyn Nets | Andrea Bargnani | Thaddeus Young | Jarrett Jack | Joe Johnson | Bojan Bogdanovic |
| Charlotte Hornets | Al Jefferson | Marvin Williams | Kemba Walker | Michael Kidd-Gilchrist | Nicolas Batum |
| Chicago Bulls | Joakim Noah | Nikola Mirotic | Derrick Rose | Doug McDermott | Jimmy Butler |

Which of the following lines of code would generate a table of the names of the starters for each team. Circle all correct answers. (Functions `function1`, `function2`, `function3` are defined below the answers.)

(a) `nba.pivot('POSITION', 'TEAM', values = 'SALARY', collect = function1)`

(b) `nba.pivot('POSITION', 'TEAM', values = 'SALARY', collect = function2)`

(c) `nba.pivot('POSITION', 'TEAM', values = 'SALARY', collect = function3)`

(d) `nba.pivot('POSITION', 'TEAM', values = 'PLAYER', collect = function1)`

(e) `nba.pivot('POSITION', 'TEAM', values = 'PLAYER', collect = function2)`

(f) `nba.pivot('POSITION', 'TEAM', values = 'PLAYER', collect = function3)`

(g) `nba.sort('SALARY',descending = True).pivot('POSITION','TEAM',values='PLAYER',collect=function1)`

(h) `nba.sort('SALARY',descending = True).pivot('POSITION','TEAM',values='PLAYER',collect=function2)`
Answer

(i) `nba.sort("SALARY",descending = True).pivot('POSITION','TEAM',values="PLAYER",collect=function3)`

(j) `nba.sort('SALARY',descending = False).pivot('POSITION','TEAM',values='PLAYER',collect=function1)`

(k) `nba.sort('SALARY',descending = False).pivot('POSITION','TEAM',values='PLAYER',collect=function2)`

(l) `nba.sort("SALARY",descending = False).pivot('POSITION','TEAM',values="PLAYER",collect=function3)`

(m) `nba.sort('PLAYER',descending = True).pivot('POSITION','TEAM',values='SALARY',collect=function1)`

(n) `nba.sort('PLAYER',descending = True).pivot('POSITION','TEAM',values='SALARY',collect=function2)`

(o) `nba.sort("PLAYER",descending = True).pivot('POSITION','TEAM',values="SALARY",collect=function3)`

(p) `nba.sort('PLAYER',descending = False).pivot('POSITION','TEAM',values='SALARY',collect=function1)`

(q) `nba.sort('PLAYER',descending = False).pivot('POSITION','TEAM',values='SALARY',collect=function2)`

(r) `nba.sort("PLAYER",descending = False).pivot('POSITION','TEAM',values="SALARY",collect=function3)`

(s) None of the above

The following functions appear in some of the answer options:

```
def function1(x):
return max(x)
```

```
def function2(x):
return x.item(0)
```

```
def function3(x):
return x
```