DATA 8 Fall 2021

Sample Exam.

FINAL EXAM

INSTRUCTIONS

This is your exam. Complete it either at exam.cs61a.org or, if that doesn't work, by emailing course staff with your solutions before the exam deadline.

This exam is intended for the student with email address `<EMAILADDRESS>`. If this is not your email address, notify course staff immediately, as each exam is different. Do not distribute this exam PDF even after the exam ends, as some students may be taking the exam in a different time zone.

For questions with **circular bubbles**, you should select exactly *one* choice.

- ◯ You must choose either this option
- ◯ Or this one, but not both!

For questions with **square checkboxes**, you may select *multiple* choices.

- ☐ You could select this choice.
- ☐ You could select this one too!

You may start your exam now. Your exam is due at <DEADLINE> Pacific Time. Go to the next page to begin.

Preliminaries

The exam is worth 120 points. You have 170 minutes to complete it.

An official final exam reference is provided. You may not use any other paper, reference, source, or computational device or system apart from those permitted for the online exam.

Unless the question specifically asks you to explain your answer, you do not need to do so, and if you write an explanation it will not be graded.

For all Python code, you may assume that the statements `from datascience import *` and `import numpy as np` have been executed. Do not use features of the Python language that have not been described in this course.

In any part, you are free to use any tables, arrays, or functions that have been defined in previous parts of the same question, and you may assume they have been defined correctly.

In starter code we provide, `_____` can mean any code, including commas and periods.

You can complete and submit the following questions before the exam starts.

(a) What is your full name?

(b) What is your student ID number?

(c) What is your Calcentral email (`_@berkeley.edu`)?

(d) Who is your Lab GSI?

1. (3 points) Counting

(a) (3 pt) Define a function `count_elem` that takes two arguments: an array `a` and a value `x`. It should return the number of times that `x` appears in `a`.

For example, `count_elem(make_array('cat', 'cat', 'dog'), 'cat')` should return 2.

```
def count_elem(a, x):
    ______________  np._______________(_______________)
```

```
def count_elem(a, x):
    return np.count_nonzero(a == x)


or


def count_elem(a, x):
    return np.sum(a == x)
```

2. (15 points) National Parks

After helping students during office hours with the Old Faithful lab, Raymond was interested in learning more about the famous geyser. After some quick research, he discovered that Old Faithful was in Yellowstone National Park. His curiosity was piqued, so he decided to do more research on other national parks. He found the following table `parks` and wanted to answer some questions, but he needs your help! For each of the following questions, write a line of code that will answer his question. The `parks` table is shown below:

Name	Location	Date established	Area	Entrance fee	Visitors
Acadia	Maine	1919	49000	25	3000000
Yosemite	California	1890	761000	35	2268000
Arches	Utah	1971	76000	30	1600000

. . . (28 more rows)

(a) (3 pt) In which year was the first national park appearing in the `parks` table established?

```
parks.sort('Date established').column('Date established').item(0)
```

We realized after the exam that "first" is ambiguous and might be interpreted as either the earliest-established park or the first row of the table. Accordingly, we accepted multiple answers for full credit:

- `parks.sort('Date established').column('Date established').item(0)`
- `parks.column('Date Established').item(0)`
- `min(parks.column('Date Established'))`

Common mistakes: - Sorting the table incorrectly by including `descending=False` in the call to sort - Finding the name of the park not the year it was established

(b) (3 pt) Assume each visitor to a park pays the corresponding entrance fee. Create a new table called `with_revenue` that contains all columns from the original `parks` table, plus a new column called 'Revenue' that shows how much money each park collected in entrance fees for that year (number of visitors times the entrance fee).

`with_revenue = _____`

```
with_revenue = parks.with_column('Revenue', parks.column('Entrance fee') *
parks.column('Visitors'))
```

Common mistakes: - Forgot to call table methods on the table (e.g., `with_column` and `column`) - Used `make_array` after array multiplication (the result of multiplying two arrays is already an array; `make_array` is not needed, and will give you the wrong result – an array of arrays, rather than an array of numbers) - Directly multiplying 'Entrance fee' and 'Visitors' (you can't multiply two strings; you need to extract the data from that column of the table)

Park

(c) **(3 pt)** Some of the national parks in the US are also designated as UNESCO World Heritage Sites, which are sites of importance to cultural or natural heritage. The table `unesco`, shown below, provides a list of national parks that are also UNESCO World Heritage Sites. How many national parks located in California are also designated as UNESCO World Heritage Sites?

Park
Yosemite
Glacier
Olympic
Everglades

... (54 more rows)

```
parks.join('Name', unesco, 'Park').where('Location', 'California').num_rows
```

Another valid solution:

```
parks.where('Location', 'California').where('Name', are_contained_in(unesco.column('Parks')).num_r
```

Common mistakes: - Not filtering to only include California parks - Using `np.count_nonzero` to directly compare elements in 'Name' to elements in 'Parks' - Returning a table instead of number of rows

We did not take off points for writing more than one line of code.

(d) **(3 pt)** After looking at the `parks` table again, Raymond realized that it may be easier to interpret the geographical size of each park by assigning it one of the labels "Small", "Medium", or "Large".

Using the skeleton code below, write a function that takes in a numeric area as input and returns a string corresponding to the geographical sizing group it belongs to. Use the following table for reference:

Category	Area Range
Small	[0, 100000)
Medium	[100000, 500000)
Large	[500000, infinity)

```
def park_size(area):
    if ______________:
        ______________
    elif _____________:
        ______________
    else:
         _____________
```

```
def park_size(area):
    if area < 100000:
        return 'Small'
    elif area < 500000:
        return 'Medium'
    else:
        return 'Large'
```

Common mistakes: - Using print statement instead of return - Did not return a string - Including end points in comparison statements (e.g., `<=` instead of `<`) - Using , which doesn't work in Python - Using `=<` instead of `<=` - Use other variable such as `x` or `Area Range` instead of the function parameter, `area`

(e) **(3 pt)** Now, using the `park_size` function you defined in the previous part, create a two-column table called `parks_with_sizes` that has one column containing the names of all parks as they appear in the `parks` table and another column containing the size label of each park as a string. The two columns should be named "Name" and "Size", respectively.

You may assume that the `park_size` function has been implemented correctly, even if you did not complete the previous part.

```
parks_with_sizes = ______________________
```

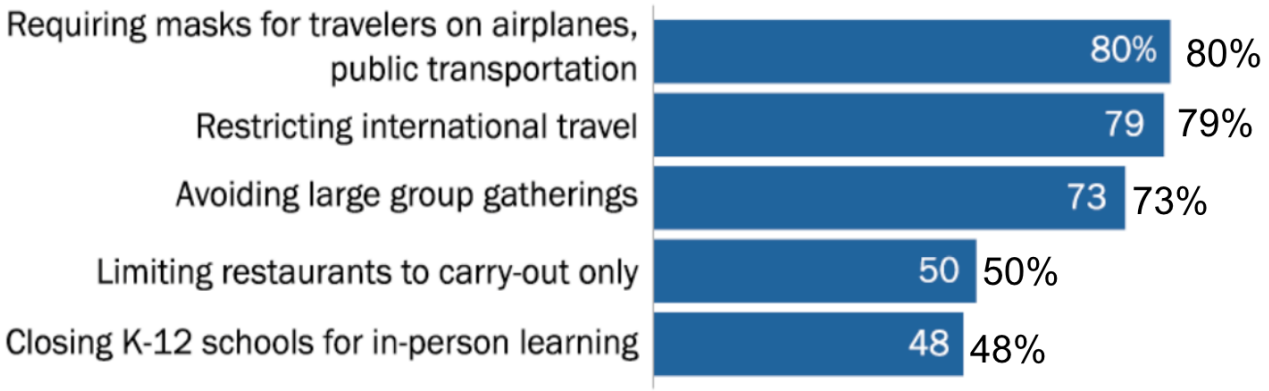
```
parks_with_sizes = parks.select('Name').with_column('Size',
parks.apply(park_size, 'Area'))
```

Common mistakes: - Calling `park_size` directly on the 'Area' column instead of using `.apply()` - Applying `park_size` onto column 'Name' or 0 instead of 'Area' or 3 - Calling table functions such as `.drop()`, `.select()`, and/or `.relabeled()` after `.apply()` - Returning an array instead of a two-column table

3. (3 points) Fighting Covid

The bar chart below is from a recent PEW Research Center survey. Each bar represents the percent of U.S. adults who view the corresponding policy as necessary to address the coronavirus outbreak. The percent in each bar is provided next to the bar for ease of reading.

(a) (3 pt) Does this bar chart display a categorical distribution? Pick the best answer from the options below.



COVID Bar Chart

- ◯ Yes, because "Requiring masks," "Limiting restaurants," etc are categories.
- ◯ No, this is a numerical distribution because percents are numbers.
- ● No, this is not a distribution of any kind.
- ◯ Maybe, maybe not. There is not enough information to decide.

The percents don't add up to 100. The data show that each respondent can give more than one answer.

4. (6 points) Mysterious Figure

A Data 8 student has defined a function called `repeat_it`. The function takes two arguments.

- The first argument is a function that takes no arguments and returns a numerical value.
- The second argument is a positive whole number.

The expression `repeat_it(function, n)` evaluates to an array of the results of `n` repetitions of calling `function`.

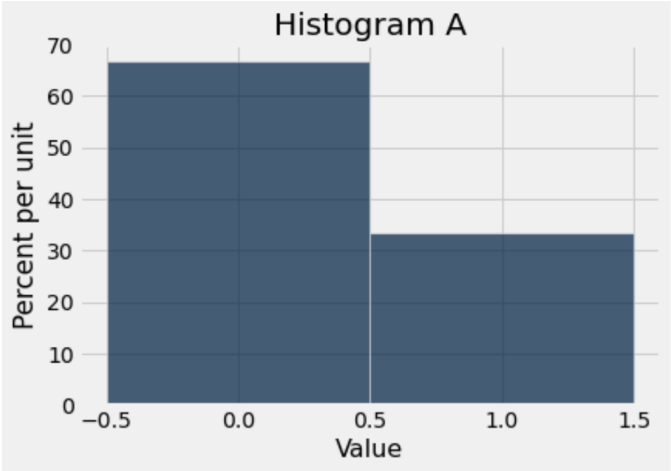
Here is an example.

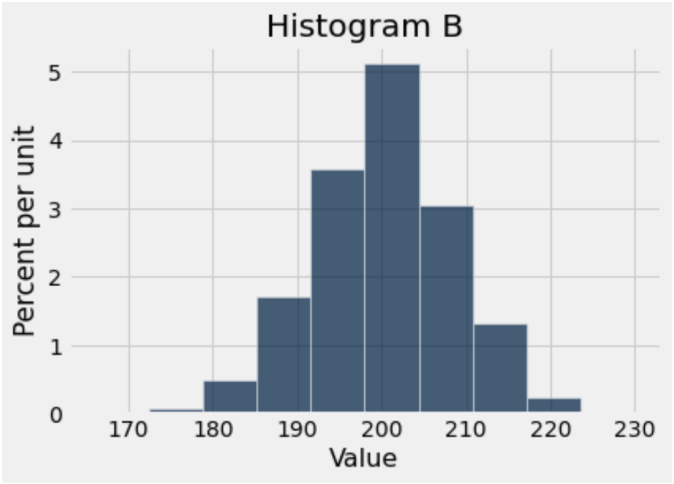
```
def example_function():
    return 10

repeat_it(example_function, 3)
```

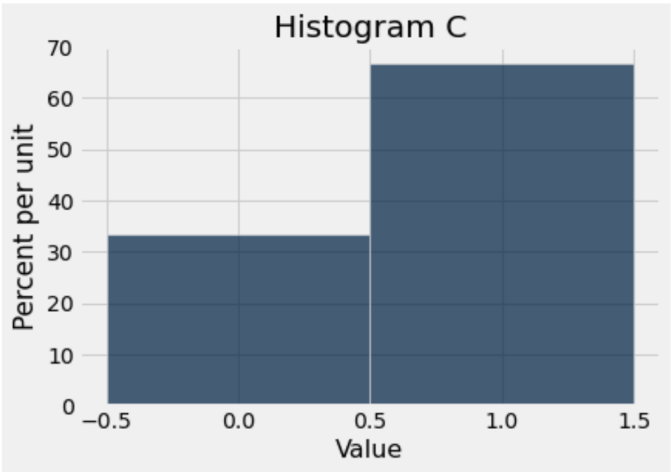
Running the example code yields the same array as you get from the call `make_array(10, 10, 10)`.

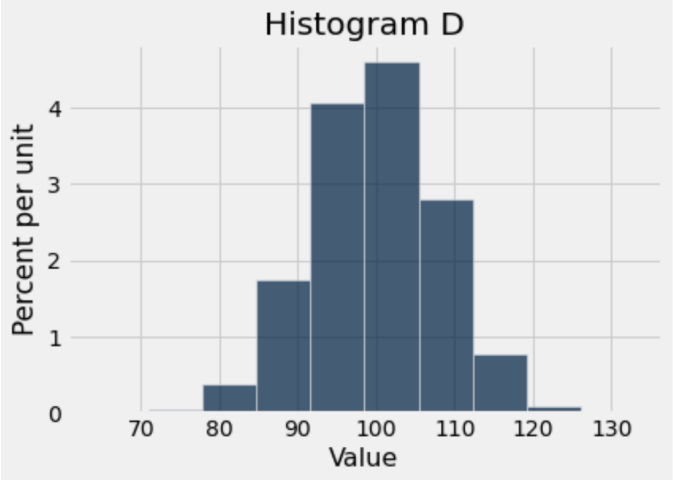
(a) (3 pt) The code below produces one of the histograms (A)-(E).

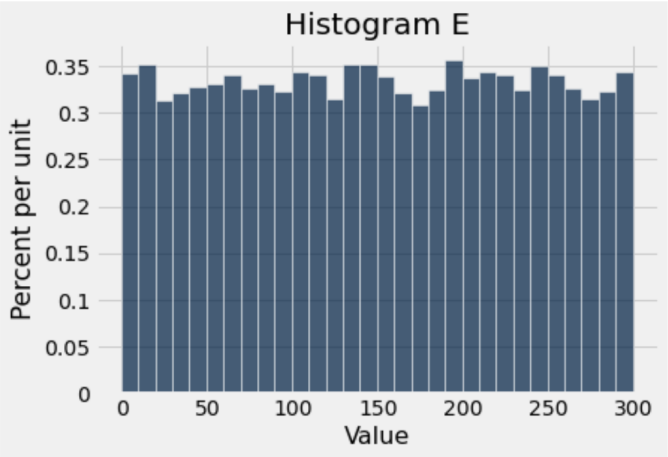
```
def my_function():
    t = Table().with_column('Value', make_array(0, 1, 1))
    return sum(t.sample(300).column('Value'))

Table().with_columns(
    'Value', repeat_it(my_function, 10000)
).hist()
```











Which histogram is produced by the code?

- ○ A
- ● B
- ○ C
- ○ D
- ○ E

(b) (3 pt) Briefly explain your choice of histogram.

We are adding the results of sampling 300 times at random with replacement from 0, 1, 1. That's the sum of a large independent sample, so its distribution is roughly normal by the CLT. About two-thirds of the time we'll get 1, so the sum will be around 200.

Common mistakes: - Only mentioning CLT, or only mentioning the mean (not both)

5. (6 points) Song Durations

Christina is interested in learning more about the duration of songs on Spotify. She collects a random sample of 400 songs listed on the platform and stores the data in the table `songs`, which has one column labelled "Duration". The average song duration in the sample is 185 seconds and the standard deviation is 25 seconds. Christina wants to use this sample of songs to make some estimates about the population of songs and their durations.

(a) (3 pt) Define a function `song_ci` that constructs a 95% confidence interval for the population mean as follows and returns it as an array. The function takes in the argument `reps`, the number of bootstrap repetitions wanted.

```
def song_ci(reps):
    stats = ________
    for ______:
        resample = _______
        new_mean = _______
        stats = _________
    left_end = __________
    right_end = ________

    return ______________
```

```
def song_ci(reps):
    stats = make_array()
    for i in np.arange(reps):
        resample = songs.sample()
        new_mean = np.mean(resample.column(0))
        stats = stats.append(stats, new_mean)
    left_end = percentile(2.5, stats)
    right_end = percentile(97.5, stats)

    return make_array(left_end, right_end)
```

(b) (3 pt) Christina creates an interval by using `song_ci(10000)`. To get a more accurate estimate at the same level of confidence, Christina would like to create a new 95% confidence interval that is half as wide as this one. Which one of the following do you think is the best advice for her?

- ○ She should use song_ci(20000)
- ○ She should use a sample of size 800
- ○ She should use song_ci(40000)
- ● She should use a sample of size 1600

The width of the confidence interval is about 4 times the standard deviation of the sample mean. By the CLT, the SD of the sample mean is the population SD divided by the square root of the sample size. So, to reduce the confidence interval width by 2x, we need to reduce the SD of the sample mean by 2x, which requires us to increase the sample size by 4x.

Another way to approach this is to notice (as above) that in calculating the width of the interval, the sample size appears only in the denominator, with a square root. The original sample size of 400 has a square root of 20, and a sample size of 1600 has a square root of 40. If you divide by 40 you'll get half of what you got from dividing by 20.

By the Law of Averages, increasing the number of bootstrap repetitions will not change the shape of the histogram of `stats` much, so it won't change the confidence interval that is produced much.

6. (21 points) Birth Days

(a) (3 pt) Many simulations involve carrying out the same chance-based process repeatedly and generating an array of simulated values. Define a function `repeat_it` that takes two arguments:

- The first argument is a function that takes no arguments and returns a numerical value. If `f` is such a function, remember that to call it you have to use `f()`.
- The second argument is a positive whole number.

The expression `repeat_it(f, n)` evaluates to an array of the results of `n` repetitions of calling `f`.

```
def repeat_it(f, n):
    results = make_array()
    for i in np.arange(n):
        results = np.append(results, f())
    return results
```

Common mistakes: - Not redefining the array when appending - Using `for i in n` instead of `for i in np.arange(n)`

(b) (3 pt) Now for the data. A doctor studying births in a large hospital system asks you to determine whether or not births in the system are distributed evenly over the week.

To help you make your decision, the doctor has gathered data on 1000 randomly sampled births in the system. Here is the distribution of days of the week for the 1000 births in his sample.

	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Proportion of births	0.11	0.14	0.15	0.15	0.17	0.15	0.13

In case you need it later, we have put the proportions in an array.

```
data = make_array(0.11, 0.14, 0.15, 0.15, 0.17, 0.15, 0.13)
```

From the following options, select all that are correct statements of the null hypothesis.

☐ The distribution of days in the sample is the same as the distribution of days in the population. Any difference is due to chance.

■ The 1000 days in the sample are drawn uniformly at random with replacement from the seven days of the week.

☐ The 1000 days in the sample are drawn uniformly at random with replacement from the distribution given in the data table above.

☐ Each of the 1000 babies has an 13% chance of being born on Sunday, regardless of all the other babies.

☐ Each of the 1000 babies has a 1/7 chance of being born on Sunday, regardless of all the other babies.

■ Each of the 1000 babies has an equal chance of being born on any day of the week, regardless of all the other babies.

The null hypothesis comes from the doctor's question about whether the births "are distributed evenly over the week." - The first option on the list is wrong because the distribution in a random sample is highly unlikely to be the same as in the population (there's always some chance error), and the statement doesn't involve an even distribution over the week. - The third option is wrong because it specifies a non-uniform distribution. - The fourth and fifth options are wrong because they focus on a specific day whereas the doctor is asking about the whole week. Also, in the fourth option the specified proportion is 13% which is not 1/7.

(c) **(3 pt)** Select which one of the following is a correct alternative hypothesis.

- ○ Babies are more likely to be born on Thursday than on any other day of the week.
- ● The model in the null hypothesis is incorrect.
- ○ The model in the null hypothesis overestimates the proportion of births on Sundays.
- ○ The distribution of the days in the sample is different from the distribution of days in the population.

The alternative comes from the doctor's question about whether "or not" the births have a uniform distribution over the week. - The first and third options are wrong because each one specifies a direction whereas the doctor doesn't. Additional technical problem: They are based on the extreme values in the sampled data, which biases the test. - The last option is not a hypothesis. It's a consequence of random sampling.

(d) **(3 pt)** Choose an appropriate test statistic to conduct this hypothesis test.

TVD

Total variation distance (TVD) is a good way to measure the similarity of two distributions, namely, the distribution of the sample and the uniform distribution hypothesized by the null hypothesis.

Common mistakes: - Chose wrong test statistics based on the alternative hypothesis in 6c - If you chose option 1 (Babies more likely on Thu) or 3 (overestimates births on Sun) in 6c, you cannot use TVD as your test stat as TVD measures distances between 2 categorical distributions - Chose a directional alternative hypothesis in 6c but the test statistic uses some sort of absolute difference - Chose a non-directional alternative hypothesis in 6c but the test statistic didn't use absolute difference

(e) **(3 pt)** Write a Python expression that evaluates to the observed test statistic.

```
sum(abs(data - (1/7)*np.ones(7))) / 2
```

Common mistakes: - Measuring the calculated distance between the sample proportions and the *population* proportions (should be calculating against the null proportions, i.e., the uniform distribution) - Mistakes in the TVD formula, e.g., not dividing by 2, taking the average of the differences

(f) (3 pt) To carry out the hypothesis test, you must simulate your test statistic. Define a function `simulate_one` that takes no argument and returns one value of your test statistic simulated under appropriate assumptions.

```
def simulate_one():
    sim_data = ______________________
    return ______________________
```

```
def simulate_one():
    sim_data = sample_proportions(1000, (1/7)*np.ones(7))
    return sum(abs(sim_data - (1/7)*np.ones(7))) / 2


or


def simulate_one():
    days = make_array('Su','Mo','Tu','We','Th','Fr','Sa')
    sim_data = Table().with_column('Day', days).sample(1000).group('Day').column(1)
    return sum(abs(sim_data - (1/7)*np.ones(7))) / 2
```

(g) (3 pt) Suppose you decide to simulate the test statistic 5000 times and use 1% as the cutoff for the p-value of the test. Fill in the blank in the code below.

For the test to reject the null hypothesis, the observed test statistic has to be bigger than the value of the following expression:

```
percentile(________, repeat_it(simulate_one, 5000)).
```

99

For the test to reject the null hypothesis, the p-value has to be less than 1%. Large values of the TVD favor the alternative, so the p-value is the right-hand tail area starting at the observed statistic. 1% is the area of the tail to the right of the 99th percentile of the simulated statistics. If the observed statistic is bigger than that, the area of the right-hand tail starting there will be less than 1%.

7. (12 points) Movie Reviews

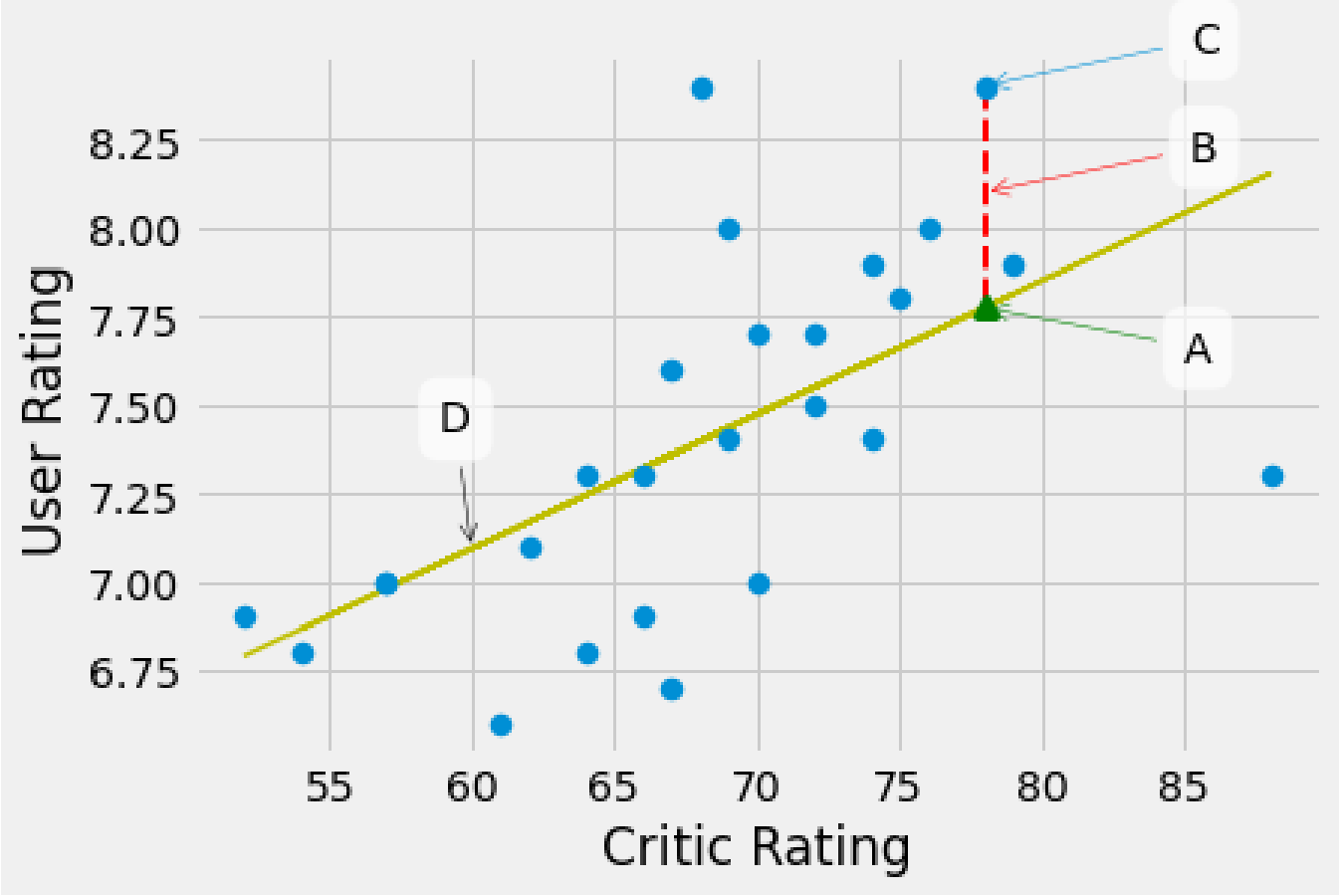
Sonya is interested in seeing how IMDb user reviews differ from critic's opinions on movies. She collects data on User Rating and Critic Rating for every movie released by Marvel Studios in the past 15 years.

(a) (3 pt) Since critics often see the movie before regular users, Sonya is interested in predicting User Rating from Critic Rating. Which of the following techniques would help her do so? Select all that are correct.

- ☐ Classification
- ☐ Bootstrapping
- ■ Method of Least Squares
- ■ Regression Equations
- ☐ Simulation
- ☐ None of the above

(b) (3 points)

After inspecting her data, Sonya reasons that a linear model would be a good fit, and creates one for her data. Below is her visualization with certain aspects labeled A, B, C, or D. Match each term below to its label on the graph, or "Not Pictured". Some letters may be used multiple times or not at all.



i. Predicted Value

- (●) A
- ( ) B
- ( ) C
- ( ) D
- ( ) Not Pictured

ii. Residual

- ( ) A
- (●) B
- ( ) C
- ( ) D
- ( ) Not Pictured

iii. Line of Best Fit

- ( ) A
- ( ) B
- ( ) C
- (●) D
- ( ) Not Pictured

iv. Intercept

- ( ) A
- ( ) B
- ( ) C
- ( ) D
- (●) Not Pictured

v. RMSE

- ( ) A
- ( ) B
- ( ) C
- ( ) D
- (●) Not Pictured

vi. Observed Value

- ○ A
- ○ B
- ● C
- ○ D
- ○ Not Pictured

vii. Fitted Value

- ● A
- ○ B
- ○ C
- ○ D
- ○ Not Pictured

(c) **(3 pt)** In the graph, Sonya finds the correlation between Critic Rating and User Rating to be 0.6. Suppose she now adds a new movie to her dataset with a Critic Rating of 60 and a User Rating of 7.8. Will the correlation increase, decrease, or stay the same?

- ○ Increase
- ● Decrease
- ○ Stay the same

(d) **(3 pt)** Explain your choice above.

(60, 7.8) is above the left side of the line, so it will "pull up" the left side of the best-fit line, making the best-fit line more horizontal, which corresponds to a decrease in the correlation.
In general, a single outlier can have a large effect on the best-fit line.

8. (6 points) Prediction Error

Students in a class take two tests called Midterm and Final. The professor has developed a model to predict Final scores based on Midterm scores, using the data from past semesters' offerings of the course.

The table `predictions` contains a column `Midterm` consisting of each possible Midterm score. The only possible Midterm scores are 0, 1, 2, ..., 100, and therefore the table has 101 rows. For each possible Midterm score, the second column `Predicted Final` contains the corresponding predicted Final score based on the professor's model.

Midterm	Predicted Final
0	23.17
1	23.34
2	23.52

... (98 rows omitted)

This semester's class has 200 students. After the students take both tests, the professor creates a table called `scores` that has one row for each of the 200 students. The column `Midterm` contains the student's score on Midterm. The column `Final` contains the student's score on Final.

Midterm	Final
69	39
63	47
63	45

... (197 rows omitted)

(a) (3 pt) Create a table `errors` that has one row for each of the 200 students, and four columns. In each row,

- The column `Midterm` should have the student's Midterm score.
- The column `Final` should have the student's Final score.
- The column `Predicted Final` should have the student's predicted Final score based on the professor's model.
- The column `Error` should have the difference between the student's Final score and predicted Final score.

For clarity, here is a randomly selected row of the table `errors` that you are going to create.

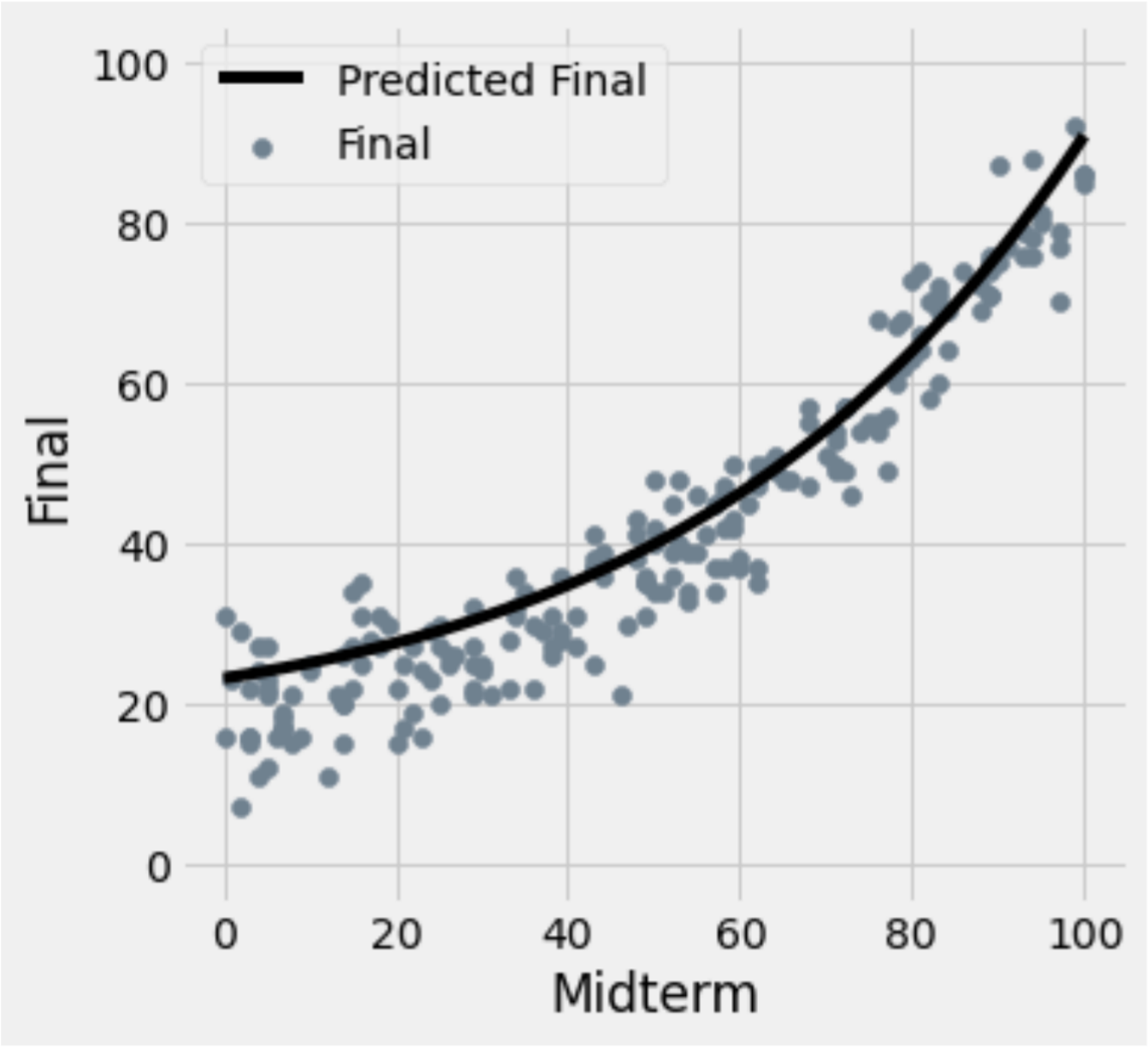
Midterm	Final	Predicted Final	Error
91	72	77.4	-5.4

```
t = ____________________
e = ____________________
errors = ____________________
```

```
t = scores.join('Midterm', predictions)
e = t.column('Final') - t.column('Predicted Final')
errors = t.with_column('Error', e)
```

Common mistakes: - Did not join scores and predictions correctly (e.g., on incorrect columns, or did not join at all) - Forgets to find difference on joined table, so rows do not match - Subtracts incorrectly i.e. Predicted Final - Final instead of Final - Predicted Final

(b) **(3 pt)** The graph below shows the actual and predicted scores. It is fine to assume that each dot represents exactly one student.



Prediction Error Scatter

Use the scatter plot to pick the statement that evaluates to `True`.

- (●) `np.average(errors.column('Error')) < 0`
- ( ) `np.average(errors.column('Error')) == 0`
- ( ) `np.average(errors.column('Error')) > 0`

Visually, we can see that the data points are more often below the prediction line than above it, so the error is more often negative than positive.

9. (15 points) Ages

A data scientist takes a random sample of 400 people in a large city. The ages of the sampled people have an average of 35 years and an SD (standard deviation) of 20 years.

The data scientist bootstraps the sample 10,000 times, calculates the mean age of each bootstrapped sample, and finds the interval that contains the middle 95% of the 10,000 bootstrapped means. The interval goes from 33 years to 37 years.

(a) (3 points)

The interval (33 years, 37 years) is an approximate 95% confidence interval for the _____ of the people in the _____.

Fill in the blanks above by selecting from the following options: ages, average age, average, sample, sample mean, city, city mean

i. Blank 1:

- ○ ages
- ● average age
- ○ average
- ○ sample
- ○ sample mean
- ○ city
- ○ city mean

ii. Blank 2:

- ○ ages
- ○ average age
- ○ average
- ○ sample
- ○ sample mean
- ● city
- ○ city mean

A confidence interval estimates a parameter, that is, a number in the population. Here, the population is the city, which is the second blank. The first blank is the specific number being estimated, which is the average age.

(b) **(3 pt)** The distribution of the ages of the sampled people (pick one option):

- ○ is approximately normal by the Central Limit Theorem.
- ○ is approximately normal, but not because of the Central Limit Theorem.
- ● is not normal, not even approximately.
- ○ may be approximately normal, or not; we need more information to decide.

The distribution of ages can't be normal, as it has a mean of 35 and a SD of 20. A normal distribution has 2.5% of its data that is below the mean minus 2 times the SD. So, if the distribution of ages were normally distributed, 2.5% of the ages would be below -5. But ages can never be negative—so the data distribution must not be normal.

We gave significant partial credit to answers that state "we need more information to decide".

CLT is not applicable: the question asks about the distribution of ages of sampled people, not the distribution of the sample mean.

(c) **(3 pt)** True or false: Approximately 95% of the people in the sample are between 33 and 37 years old.

- ○ True
- ● False

The interval is too narrow to capture 95% of the ages in the sample. The SD of the sample is 20.

(d) **(3 pt)** True or false: Approximately 95% of the people in the city are between 33 and 37 years old.

- ○ True
- ● False

The interval is too narrow to capture 95% of the ages in the city. It only estimates the average age in the city. The SD of ages in the city is around 20.

(e) **(3 pt)** The city is in a country where the average age is 35.5 years. If possible, perform a statistical test of whether or not the average age in the city is 35.5 years, using 1% as the cutoff for the p-value. State your conclusion by picking one of the options below.

- ○ Since the p-value cutoff and the confidence level of the interval are inconsistent, we cannot perform this test.
- ● The test concludes that the data are consistent with the hypothesis that the average age in the city is 35.5 years.
- ○ The test concludes that the data are not consistent with the hypothesis that the average age in the city is 35.5 years.

The 95% confidence interval is [33,37]. We don't know what the 99% confidence interval would be, but it must be even wider and include all of [33,37] plus some more. So, we know that 35.5 is inside the 99% confidence interval. Therefore, the observed data are consistent with the null hypothesis, at the 99% confidence (1% cutoff) level.

To put it another way: at the 95% confidence level, we would not reject the null hypothesis, because 35.5 is in the 95% interval [33,37]. If we don't reject the null hypothesis at 95% confidence, we don't reject it at any higher confidence, either.

10. (9 points) Coffee Consumption

Meghan wants to estimate the difference between the coffee consumption of Data 8 students and Data 100 students. To help answer this question, she will take a random sample of 150 students from each of the two classes next term. You can assume that each class will have well over 1000 students and that no student will be enrolled in both classes.

Meghan will measure each sampled student's coffee consumption by the total amount (in ounces) of coffee that the student will drink during the Spring semester. She will put the results in a table `data` that has one row for each of the 300 sampled students, one column `Coffee` containing the student's coffee consumption, and one column `Class` containing the string `Data 8` or `Data 100` depending on which class the student is taking.

She will then create two new tables as follows:

```
data8 = data.where('Class', are.equal_to('Data 8'))
data100 = data.where('Class', are.equal_to('Data 100'))
```

Meghan would like to estimate the following parameter: the difference between the mean consumption of coffee in Data 8 and the mean consumption of coffee in Data 100. Define this difference as Data 8 mean - Data 100 mean.

(a) (1.5 pt) Consider the following process:

- Repeat the following 10,000 times:
  - Bootstrap the table `data8` and compute the mean of the `Coffee` column of the bootstrapped table.
  - Bootstrap the table `data100` and compute the mean of the `Coffee` column of the bootstrapped table.
  - Find the difference between the bootstrapped Data 8 mean and the bootstrapped Data 100 mean.
- Find the endpoints of the interval created by the middle 99% of the 10,000 differences.

Is that a correct way of creating an approximate 99% confidence interval for the parameter?

● Yes

○ No

(b) (1.5 pt) If you chose "No" above, why not? If you chose "Yes," you don't have to write anything.

"No" is incorrect - this should be blank.

(c) (1.5 pt) Is the following process a correct way of creating an approximate 99% confidence interval for the parameter?

- Repeat the following 10,000 times:
  - Shuffle the rows of `data8` and compute the mean of the `Coffee` column of the shuffled table.
  - Shuffle the rows of `data100` and compute the mean of the `Coffee` column of the shuffled table.
  - Find the difference between the `Coffee` mean of the shuffled `data8` table and the `Coffee` mean of the shuffled `data100` table.
- Find the endpoints of the interval created by the middle 99% of the 10,000 differences.

○ Yes

● No

(d) (1.5 pt) If you chose "No" above, why not? If you chose "Yes," you don't have to write anything.

There won't be an interval. All 10,000 differences will be the same. Shuffling an entire class table will generate the same class mean every time.

(e) (1.5 pt) Is the following process a correct way of creating an approximate 99% confidence interval for the parameter?

- Repeat the following 10,000 times:
  - Shuffle the `Class` column of `data` and create a new table by attaching the shuffled class labels to the original column `Coffee` of the `data` table.
  - Group this table by the shuffled labels and find the difference between the `Data 8` group mean and the `Data 100` group mean.
- Find the endpoints of the interval created by the middle 99% of the 10,000 differences.

○ Yes

● No

(f) (1.5 pt) If you chose "No" above, why not? If you chose "Yes," you don't have to write anything.

This process will incorrectly give an interval that is centered around 0. Each of the two samples will resemble the combined distribution of the two classes, instead of one of them resembling Data 8 and the other resembling Data 100.

Common mistakes: - Answer mentions A/B testing, but also states that we cannot get a confidence interval from A/B testing. This is not correct. We can get a CI from A/B test. The issue is that the problem asks us to estimate a parameter, not test a hypothesis, so A/B testing is not an appropriate approach to this problem. The method of generating 10,000 differences here would be a reasonable implementation of an A/B test (to test the null hypothesis that the Data 8 and Data 100 distributions are identical), but it is not a correct way to compute a 99% confidence interval for the population parameter. - Answer only mentions that we should sample with replacement and/or bootstrap instead of shuffling, without any other explanation.

11. (3 points) Positive Test

Doctors in a city have access to a medical test for a disease that affects 1% of the people in the city. The test has high accuracy:

- For a person who has the disease, the test returns a positive result with chance 98%.
- For a person who does not have the disease, the test returns a negative result with chance 99%.

(a) (3 pt) A person in the city has symptoms of the disease and visits their local doctor. The doctor examines the patient and recommends that the patient take the test. The test result comes back positive, and the doctor turns to you for advice, asking, "Now that we know the test result is positive, what is the chance that the person has the disease?" Which of the following is your answer?

- ○ 0.01
- ○ 0.98
- ○ 0.01×0.98
- ○ $0.01 \times 0.98/(0.01 \times 0.98 + 0.99 \times 0.01)$
- ○ $0.01 \times 0.98/(0.01 \times 0.98 + 0.99 \times 0.99)$
- ● I went through all the calculations above and none of them is valid.

Knowing that the patient had symptoms and the doctor recommended the test implies that we can't treat the patient as a random draw from the population. So using 0.01 and 0.99 as the prior probabilities in (d) is not valid, and the other options are wrong even for a randomly drawn patient.

Some students wrote in their Last Words that they were assuming that the patient is a random draw from the population. This is not true, and as such that answer received only partial credit. See Chapter 18.2 of the textbook for more details:

> Our assumption was that a randomly chosen person was tested and got a Positive result. But this doesn't happen in reality. People go in to get tested because they think they might have the disease, or because their doctor thinks they might have the disease. **People getting tested are not randomly chosen members of the population.**

12. (12 points) Movie Directors

In each part of this question, you are free to use tables and functions that have been defined earlier in the question, even if you couldn't define them correctly.

(a) (3 pt) The Python function `np.unique` takes an array as its argument and returns an array consisting of the distinct elements of the argument array. Here is an example of its use.

```
example_array = make_array('cat', 'cat', 'dog', 'bear', 'bear', 'dog', 'tiger', 'bear')
np.unique(example_array)
```

```
>>> array(['bear', 'cat', 'dog', 'tiger'], dtype='<U5')
```

The output is an array consisting of the four distinct elements in `example_array`.

Define a function `count_distinct` that takes an array as its argument and returns the number of distinct elements in the array.

```
def count_distinct(a):
    return len(np.unique(a))


or


def count_distinct(a):
    return Table().with_column('Uniqs', np.unique(a)).num_rows
```

Common mistakes: - Used `np.count_nonzero` rather than checking the length of the unique array - Returned an array rather than an integer - Tried to use `count_elem` which was prone to errors like comparing the result to the wrong number or not having an effective way to make an array of distinct elements

(b) (3 pt) Each row of the table `directors` corresponds to a movie released by a major Hollywood studio in the years 1980 through 2019. The table has five columns.

- `Movie` contains the name of the movie.
- `Studio` contains the name of the studio that released the movie.
- `Year` contains the year in which the movie was released.
- `Decade` contains the decade in which the movie was released. There are four decades: `1980` consists of the years 1980 through 1989, `1990` consists of the years 1990 through 1999, and so on.
- `Director` contains the name of the director of the movie. You can assume that only one director is listed for each movie.

The table has numerous rows. To show you what it looks like, here are just three of the rows in which the director is J.J. Abrams. He has directed many movies and appears in other rows as well.

Movie	Studio	Year	Decade	Director
Mission Impossible	Paramount	2006	2000	J.J. Abrams
Super 8	Paramount	2011	2010	J.J. Abrams
Star Wars: The Force Awakens	Disney	2015	2010	J.J. Abrams

Complete the code below so that the last line evaluates to a table consisting of two columns:

- The first column should be labeled `Studio` and contain all the distinct studios.
- The second column should contain the number of different directors whose movies were released by the studio in the years 1980 through 2019.

```
t1 = ________________
t1
```

```
t1 = directors.select('Studio', 'Director').group('Studio', count_distinct)
t1


or


t1 = directors.group(['Studio', 'Director']).group('Studio')
t1
```

(c) (3 pt) Complete the code below so that the last line evaluates to a table that has five columns:

- A column containing all the distinct studios
- A column for each of the four decades

For each studio, the values in each decade column should contain the number of different directors whose movies were released by the studio in that decade.

```
t2 = ________________
t2
```

```
t2 = directors.pivot('Decade', 'Studio', values='Director', collect=count_distinct)
t2
```

(d) (3 pt) Which of the following does the expression `t1.column(1) - t2.drop(0).apply(sum)` evaluate to?

(Technical note: You do not need to worry about numerical inaccuracy or roundoff: all numbers are `int`s, so that won't happen.)

◯ An array in which all the values are 0

● An array in which some of the values are not 0

◯ The expression generates an error message.

Summing the rows of the pivot table can lead to counting some directors in more than one decade, as in the case of Paramount and J.J. Abrams.

To put it another way: the entry of `t1.column(1)` for Paramount counts J.J. Abrams only once (even though he directed two movies for Paramount); but the entry of `t2.drop(0).apply(sum)` for Paramount counts J.J. Abrams multiple times (e.g., once for the decade 2000 (Mission Impossible) and once for the decade 2011 (Super 8)). This will cause the entry of `t1.column(1) - t2.drop(0).apply(sum)` to be negative.

13. (6 points) Animated Movies

(a) (3 pt) Professor Wagner wants to build a classifier to predict whether a movie is animated or not, based on the script. He hires an intern to download 64,000 movie scripts and identify which are animated. He randomly shuffles this dataset and then splits it into a training set with 32,000 movies and a test set with 32,000 movies, builds a 1-nearest neighbor classifier (with k=1) using the training set, and then measures its accuracy on the test set. His classifier gets 95% accuracy on the test set.

The next day, Prof. Wagner realizes that the intern took a shortcut. The intern built the dataset by downloading 16,000 different random movies and making 4 identical copies of each movie.

Which of the following do you think is most likely to be true?

- ● The accuracy on a randomly selected new movie will be significantly less than 95%.
- ○ The accuracy on a randomly selected new movie will be about 95%.
- ○ The accuracy on a randomly selected new movie will be significantly more than 95%.

(b) (3 pt) Briefly explain the reasoning behind your choice.

Only a small fraction of the test set will be movies that don't appear in the training set. Most of the movies will have at least one copy in the training set and at least one copy in the test set, so a nearest-neighbor classifier will classify every one of them correctly no matter how good or bad the classifier is on new movies. This artificially inflates the accuracy on the test set, making it an over-estimate of the actual accuracy on a new movie.

Common mistakes: - Gave reason as sample size decreased/is smaller (less data to train on) - Misread part a's question about the accuracy of a randomly selected movie, and instead described what would happen to training set's accuracy - Partial credit was granted if student mentioned having a smaller actual sample size, but not mentioning how the value of k=1 would affect training and test accuracy

If you like a challenging thought puzzle: calculate the accuracy of the classifier on a randomly selected new movie. There is enough information in the question to infer its actual accuracy.

14. Last Words (optional)

(a) (0 pt) If there was any question on the exam that you thought required clarification to be answerable, please identify the question and state the assumptions you made in your answer. Please note that we will only consider this information if we agree that the question required clarification and that your assumptions were reasonable.

We hope you did great on the exam! Thank you for a wonderful semester!

No more questions.