# DATA 8 Spring 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.

**For fill-in-the-blank coding questions, you can put anything inside the blanks, including commas, parentheses, and periods.**

The exam is worth 160 points.

In alphabetical order, the sections are as follows (it might not be this order on your exam):

Banana Stand - 24 points

Dogecoin - 19 points

Multiverse of McDonald's - 11 points

Paint Ball - 14 points

Roses - 35 points

TikTok - 27 points

Yanay-Bot - 30 points

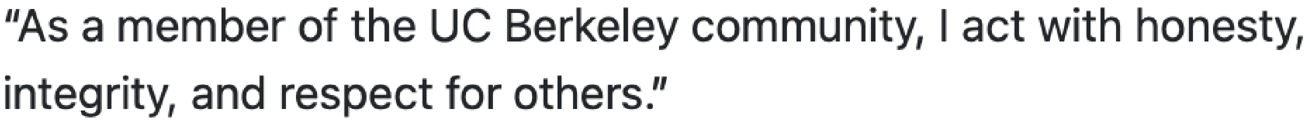
There is also a Just For Fun section, worth 0 points, and a Last Words section, where you can state any assumptions you made on the exam, also worth 0 points.

If you encounter any logistical problems during the exam, please contact us at data8berkeley@gmail.com.

**(a)** Your name:

**(b)** Your @berkeley.edu email address:

**(c)** The Berkeley Honor Code states:



"As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others."

If you agree to follow the honor code on this exam, type the statement above in the field below.

*Note*: You do not need to include the quotation marks.

**1. (27.0 points) TikTok**

Dunder Mifflin is a paper company with multiple branch locations in the Northeastern United States. Its employees at the Scranton branch are starting their own TikTok channel, but they don't know what content to post.

They're inspired by trending challenges (i.e., hashtags, expressions, sounds) so they collect data about videos posted by other Dunder Mifflin branches (e.g., Utica, Stamford) as part of a company-wide challenge.

The data are put into a table called `TIKTOK_TBL`.

Here are the first few rows:

| Branch | CHALLENGE | Day | Views | Likes | Shares | Hashtags |
|---|---|---|---|---|---|---|
| Stamford | inverted | 21 | 10324 | 4921 | 8731 | #invert #symmetrical |
| Utica | bury a friend | 23 | 4021 | 189 | 2761 | #billie #TAG #foryou |
| New York | deja vu | 27 | 32384 | 1029 | 591 | #nyc #doubletake |
| Stamford | inverted | 35 | 9349 | 2492 | 3429 | #invert #take2 |
| Utica | inverted | 24 | 8747 | 7803 | 5812 | #invert #classic |

... (NUM_ROWS rows omitted)

The table has the following columns:

- *Branch*: (string) the branch that posted the video
- *CHALLENGE*: (string) the name of the trending challenge the video was created for
- *Day*: (int) the day of the post (1 = January 1, 32 = February 1, etc.)
- *Views*: (int) the number of unique users who viewed the post
- *Likes*: (int) the number of likes the post received
- *Shares*: (int) the number of times the post was shared by other users
- *Hashtags*: (string) the hashtags included in the post

For each question below, write Python code to answer the question using what we have taught you in this class. If we ran your Python code, it should evaluate to the answer to the question.

**(a) (3.0 pt)** Write a Python expression that will return the three branches that earned the largest total number of likes.

*Hint*: Your answer should return an array.

*Recall*: The `TIKTOK_TBL` table has the following columns `['Branch', 'CHALLENGE', 'Day', 'Views', 'Likes', 'Shares', 'Hashtags']`.

```
TIKTOK_TBL.select("Branch", "Likes").group("Branch", np.sum).sort("Likes
sum", descending = True).take(np.arange(3)).column("Branch")
```

**(b) (4.0 pt)** Write a Python expression that will return the name of the most popular trending challenge.

*Note*: A trending challenge's popularity is defined as the number of unique branches that participated in it.

*Recall*: The `TIKTOK_TBL` table has the following columns `['Branch', 'CHALLENGE', 'Day', 'Views', 'Likes', 'Shares', 'Hashtags']`.

```
TIKTOK_TBL.group(make_array("CHALLENGE", "Branch")).group("CHALLENGE").sort("count",
descending = True).column("CHALLENGE").item(0)
```

**(c)** **(3.0 pt)** Write a Python expression that will visualize the branch distribution of videos whose hashtags include "#TAG".

*Recall*: The `TIKTOK_TBL` table has the following columns `['Branch', 'CHALLENGE', 'Day', 'Views', 'Likes', 'Shares', 'Hashtags']`.

```
TIKTOK_TBL.where("Hashtag", are.containing("TAG")).group('Branch').barh('Branch')
```

**(d)** **(3.0 pt)** Write a Python expression that will return a table containing the average number of views for each branch-challenge combination of all posts that got strictly at least MIN_LIKES likes.

*Hint*: A branch may participate in a challenge more than once if they like.

*Note*: Your code should return a table that has one column for each unique branch.

*Recall*: The `TIKTOK_TBL` table has the following columns `['Branch', 'CHALLENGE', 'Day', 'Views', 'Likes', 'Shares', 'Hashtags']`.

```
TIKTOK_TBL.where("Likes",are.above(MIN_LIKES)).pivot("Branch", "CHALLENGE",
"Views", np.mean)
```

**(e)** **(3.0 points)**

Dwight, the assistant to the regional manager at the Scranton branch, thinks metrics like views and shares are too simplistic. He creates his own metric, called *engagement rate*, which is defined as the sum of a video's likes and shares divided by its views.

For example, if a video has 10 likes, 20 shares, and 5 views, its *engagement rate* is $(10 + 20)/5 = 6$.

Suppose Dwight writes the following partially completed code, which returns the name of the trending challenge for the video that has the highest *engagement rate*:

```
engagement = ______(1)______
TIKTOK_TBL.with_column("Engagement", engagement).______(2)______
```

*Recall*: The `TIKTOK_TBL` table has the following columns `['Branch', 'CHALLENGE', 'Day', 'Views', 'Likes', 'Shares', 'Hashtags']`.

**i.** **(1.0 pt)** Write a Python expression that should go in blank (1).

```
(TIKTOK_TBL.column("Likes") + TIKTOK_TBL.column("Shares")) /
TIKTOK_TBL.column("Views")
```

**ii.** **(2.0 pt)** Write a Python expression that should go in blank (2).

```
sort("Engagement", descending = True).column("CHALLENGE").item(0)
```

**(f) (8.0 points)**

Rather than focusing on the trending challenges, Dwight's colleague, Jim, thinks a better strategy is to understand which branch has grown the most since its first post.

He defines a branch's *growth* as the difference between the number of views on their first video and the number of views on their latest video.

Suppose Jim writes the following partially completed code, which returns the name of the branch that has experienced the most *growth*:

```
def growth(branch):
    latest_views = ______(1)______
    first_views = ______(2)______
    return latest_views - first_views

branches = TIKTOK_TBL.group("Branch").select("Branch")
growths = ______(3)______
branches.with_column("Growth", growths).______(4)______
```

*Recall*: The TIKTOK_TBL table has the following columns ['Branch', 'CHALLENGE', 'Day', 'Views', 'Likes', 'Shares', 'Hashtags'].

i. **(2.0 pt)** Write a Python expression that should go in blank (1).

```
TIKTOK_TBL.where("Branch", branch).sort("Day", descending =
True).column("Views").item(0)
```

ii. **(2.0 pt)** Write a Python expression that should go in blank (2).

```
TIKTOK_TBL.where("Branch", branch).sort("Day").column("Views").item(0)
```

iii. **(2.0 pt)** Write a Python expression that should go in blank (3).

```
branches.apply(growth, "Branch")
```

iv. **(2.0 pt)** Write a Python expression that should go in blank (4).

```
sort("Growth", descending = True).column("Branch").item(0)
```

**(g) (3.0 pt)** Creed, another employee at the Scranton branch, believes that a branch's zip code could have an impact on its number of views. He creates a new table called `zips` which has two columns:

- *Office*: (string) the name of the Dunder Mifflin branch
- *Code*: (int) the branch's 5-digit zip code

Write a Python expression that returns the maximum number of views received by a video among videos posted by branches with a zip code that is strictly less than MAX_ZIP.

*Note*: Assume that none of the zip codes start with `0` as their first digit.

*Recall*: The `TIKTOK_TBL` table has the following columns `['Branch', 'CHALLENGE', 'Day', 'Views', 'Likes', 'Shares', 'Hashtags']`.

```
np.max(TIKTOK_TBL.join("Branch", zips, "Office").where('Code',
are.below(MAX_ZIP)).column("Views"))
```

**2. (14.0 points) Paint Ball**

Greendale College is hosting its annual paint ball competition. Jeff, Britta, and Annie, 3 students at the college who will not be competing, want to take their shot at betting on who will win.

They know that there are 500 people in the competition, each with an equal chance of winning.

Among the competitors, 70 are faculty, 400 are students, and 30 are staff.

**(a) (3.0 pt)** If Jeff, Britta, and Annie each randomly picks a competitor (without telling each other their picks), what is the chance that all three of them pick a student?

○ $1 - \left(\frac{100}{500} \times \frac{100}{500} \times \frac{100}{500}\right)$

○ $1 - \left(\frac{400}{500} \times \frac{400}{500} \times \frac{400}{500}\right)$

○ $3 \times \frac{400}{500} \times \frac{400}{500} \times \frac{400}{500}$

○ $\frac{400}{500}$

● $\frac{400}{500} \times \frac{400}{500} \times \frac{400}{500}$

○ There is not enough information to answer.

**(b) (3.0 pt)** If Jeff, Britta, and Annie each randomly picks a competitor (without telling each other their picks), what is the chance that at least one of them picks the actual winner of the paint ball competition?

● $1 - \left(\frac{499}{500} \times \frac{499}{500} \times \frac{499}{500}\right)$

○ $1 - \left(\frac{1}{500} \times \frac{1}{500} \times \frac{1}{500}\right)$

○ $3 \times \frac{1}{500} \times \frac{1}{500} \times \frac{1}{500}$

○ $\frac{1}{500}$

○ $\frac{1}{500} \times \frac{1}{500} \times \frac{1}{500}$

○ There is not enough information to answer.

**(c) (4.0 pt)** If Jeff, Britta, and Annie each randomly picks a competitor (without telling each other their picks), what is the chance that only one of them picks the actual winner of the paint ball competition?

○ $1-\left(\frac{1}{500}\times\frac{499}{500}\times\frac{499}{500}\right)$

○ $1-\left(\frac{499}{500}\times\frac{1}{500}\times\frac{1}{500}\right)$

● $3\times\frac{1}{500}\times\frac{499}{500}\times\frac{499}{500}$

○ $\frac{1}{500}$

○ $\frac{1}{500}\times\frac{499}{500}\times\frac{499}{500}$

○ There is not enough information to answer.

**(d) (4.0 pt)** Suppose Jeff knows that a faculty member will win the competition because Sehang (one of the faculty members) has removed the paint ammunition from all of the student and staff's paint ball guns ahead of the competition. Assume Britta and Annie don't have access to this extra information.

If Jeff, Britta, and Annie each randomly picks a competitor (without telling each other their picks), and Jeff knows to specifically pick from among the faculty members, what is the chance that all three of them pick the correct winner?

*Recall*: Among the 500 competitors, 70 are faculty, 400 are students, and 30 are staff.

○ $1-\left(\frac{1}{70}\times\frac{1}{500}\times\frac{1}{500}\right)$

○ $1-\left(\frac{1}{70}\times\frac{70}{500}\times\frac{70}{500}\right)$

○ $3\times\frac{1}{70}\times\frac{1}{500}\times\frac{1}{500}$

○ $\frac{1}{70}\times\frac{70}{500}\times\frac{70}{500}$

● $\frac{1}{70}\times\frac{1}{500}\times\frac{1}{500}$

○ There is not enough information to answer.

**3. (11.0 points) Multiverse of McDonald's**

Rick and Morty spend most of their time going on interdimensional adventures across parallel universes (called dimensions).

Every dimension contains a parallel version of our reality, with slight variations (for example, in Dimension C-137, the #1 public university is called UC Blockeley). There are a small percentage of dimensions (exactly *2%*), in which McDonald's offers Sriracha Mac sauce.

Morty wants to try this sauce, but he doesn't have time to explore every dimension. He builds a test (the Sauce Test) to estimate whether a dimension has a McDonald's that offers Sriracha Mac sauce.

If the dimension has the sauce, the Sauce Test returns a positive result 95% of the time. If the dimension doesn't have the sauce, the Sauce Test returns a positive result 4% of the time.

**(a)** **(2.0 pt)** If Morty randomly selects a dimension to travel to, what is the probability that the McDonald's in that dimension does **not** have Sriracha Mac sauce?

○ $\frac{0.02 \times 0.95}{0.02 \times 0.95 + 0.98 \times 0.04}$

○ $\frac{0.98 \times 0.95}{0.98 \times 0.95 + 0.02 \times 0.04}$

○ 0.02

● 0.98

○ $0.98 \times 0.04$

○ $0.02 \times 0.04$

**(b)** **(2.0 pt)** If Morty randomly selects a dimension to travel to, what is the probability that it does **not** have Sriracha Mac sauce **and** gets a positive result in his Sauce Test?

○ $\frac{0.02 \times 0.95}{0.02 \times 0.95 + 0.98 \times 0.04}$

○ $\frac{0.98 \times 0.95}{0.98 \times 0.95 + 0.02 \times 0.04}$

○ 0.02

○ 0.98

● $0.98 \times 0.04$

○ $0.02 \times 0.04$

**(c)** **(3.0 pt)** On their next interdimensional adventure, Rick and Morty visit the Tusk Dimension, in which all humans have elephant-like tusks (including Elon Tusk, CEO of Tuskla).

Morty runs the Sauce Test on that dimension and gets a positive result.

Given this information, what is the probability that the Tusk Dimension has Sriracha Mac sauce?

● $\dfrac{0.02 \times 0.95}{0.02 \times 0.95 + 0.98 \times 0.04}$

○ $\dfrac{0.98 \times 0.95}{0.98 \times 0.95 + 0.02 \times 0.04}$

○ 0.02

○ 0.98

○ $0.98 \times 0.04$

○ $0.02 \times 0.04$

**(d)** **(4.0 pt)** Morty is a big fan of the Dog Dimension, in which dogs run the world and humans are their pets. Morty believes that dogs are nicer than humans, so there's a good chance their McDonald's would have Sriracha Mac sauce.

Prior to visiting the Dog Dimension, Morty believes there is a 45% probability that it has Sriracha Mac sauce.

Suppose Morty later runs the Sauce Test on the Dog Dimension and gets a positive result.

Given this new information, what is the probability that the Dog Dimension has Sriracha Mac sauce?

● $\dfrac{0.45 \times 0.95}{0.45 \times 0.95 + 0.55 \times 0.04}$

○ $\dfrac{0.55 \times 0.95}{0.55 \times 0.95 + 0.45 \times 0.04}$

○ 0.45

○ 0.55

○ $0.45 \times 0.04$

○ $0.55 \times 0.04$

**4. (35.0 points) Roses**

Moira Roise and her daughter, Alexis, are trying to buy some plants for the front yard of their new home. They decide to visit the only nursery in town to pick some out.

They want to buy plants that are tall and healthy but not too old. However, Jocelyn, the owner of the nursery, doesn't know the age of any of her plants.

Moira and Alexis realize that they have access to a data set stored in a table called `PLANTS_TBL`, which contains a variety of attributes about the plants they used to own in their prior home. Here are the first few rows:

| Name | Age | Mass | Height | Flower |
|---|---|---|---|---|
| Iris cristata | 4 | 6.1 | 16.3 | 0 |
| Scaevola aemula | 6 | 15.7 | 49.2 | 1 |
| Ficus lyrata | 5 | 52.9 | 61.7 | 1 |

... (158 rows omitted)

The table contains the following columns:

- *Name*: (string) the scientific name of the plant
- *Age*: (int) the plant's age, in years
- *Mass*: (float) the plant's mass, in Ounces
- *Height*: (float) the plant's height, in Centimeters
- *Flower*: (int) `1` if the plant has flowers, `0` if it does not

Along with the data set, Moira finds the following scatterplot that was generated from it:

Unfortunately, the scatterplot's x-axis is not labeled.

**(a) (2.0 pt)** Which of the following could be a variable whose values are shown along the x-axis?

*Select all that apply.*

□ *Mass* in standard units

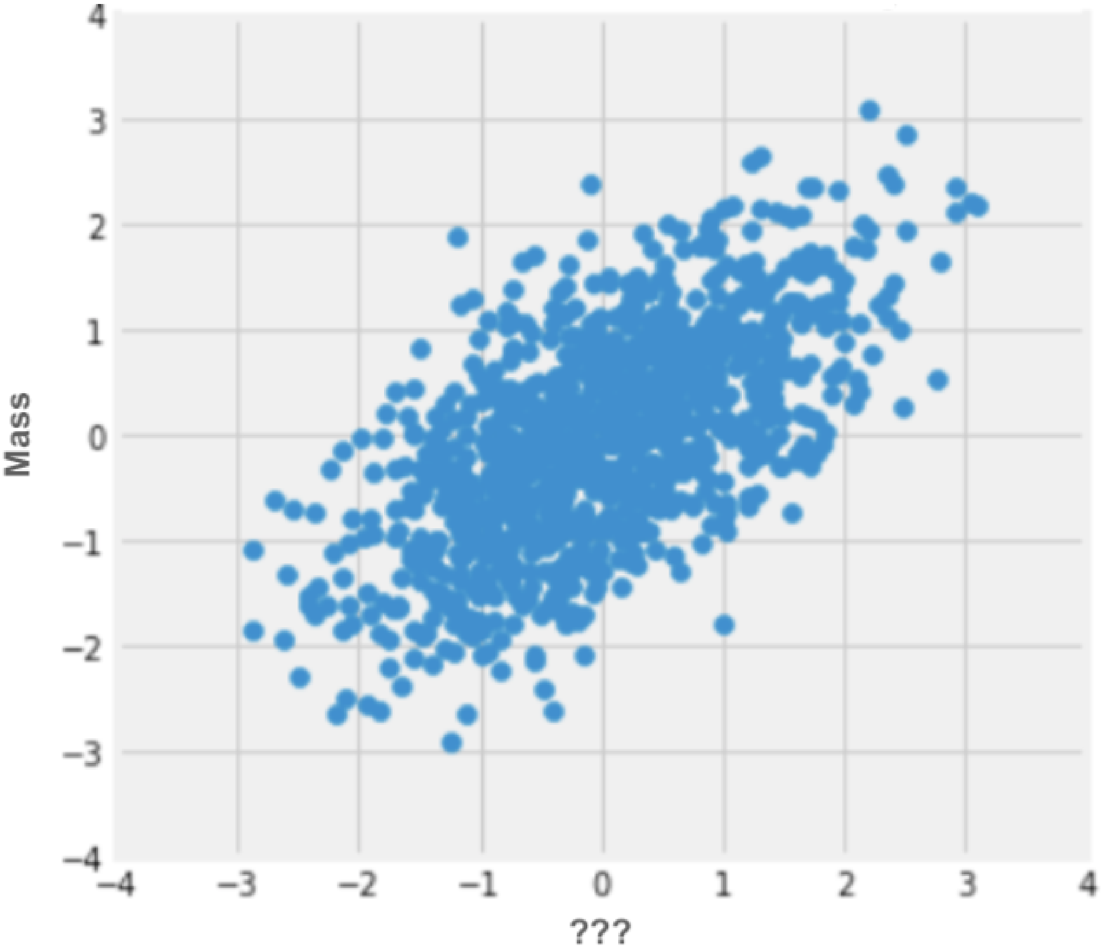
■ *Height* in standard units

□ *Age*

□ *Mass*

□ *Flower*

□ None of the above


Mass
???
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4

**(b)** **(2.0 pt)** From the scatterplot above, what can Moira conclude about the data set?

*Select all that apply.*

□ The x-axis variable is in original (non-standardized) units.

■ There is a positive association between plant mass and the x-axis variable.

■ There is a positive correlation between plant mass in standard units and the x-axis variable.

□ There is a negative association between plant mass and the x-axis variable.

□ There is no correlation between plant mass and the x-axis variable.

**(c)** **(11.0 points)**

For the next five questions, assume you know the following:

- the *Height* column has a mean of 50 and a standard deviation of 10
- the *Age* column has a mean of 8 and a standard deviation of 2
- the correlation between the *Height* and *Age* columns is 0.5

**i.** **(2.0 pt)** Suppose Moira wants to predict *Age* from *Height* and decides to fit a regression line.

What is the **intercept** of this regression line?

● 3

○ 2

○ 11

○ 0

○ -1

○ 30

○ 70

**ii.** **(2.0 pt)** If there's a plant with a height of 70 centimeters, what would the regression line (from the question above) predict as the plant's age?

○ 6

○ 8

○ 9

● 10

○ 12

**iii.** **(3.0 pt)** With regards to the regression in the question above, which of the following statements are guaranteed to be **true**?

*Select all that apply.*

□ The standard deviation of the residuals is 5.

■ The average of the residuals is 0.

■ The residuals and *Height* have a correlation of 0.

■ The sum of the positive residuals equals the absolute value of the sum of the negative residuals.

□ There are an equal number positive residuals as there are negative residuals.

**iv.** **(2.0 pt)** What are the units for the residuals in the above question?

● Years

○ Centimeters

○ Ounces

○ Shrute Bucks

○ Dogecoin

○ None of the above

**v.** **(2.0 pt)** Suppose Moira now wants to fit a regression line to predict *Height* in standard units from *Age* in standard units.

*Recall*:

- the *Height* column has a mean of 50 and a standard deviation of 10
- the *Age* column has a mean of 8 and a standard deviation of 2
- the correlation between the *Height* and *Age* columns is 0.5

What is the **slope** of this regression line?

● 0.5

○ 1/0.5

○ 10/2

○ 2/10

○ 0.5 * (10/2)

○ 0.5 * (2/10)

○ There is not enough information to answer.

**(d) (8.0 pt)** To get a baseline, Alexis wants to understand what the age of the plants tend to be when they haven't yet sprouted and have a height of 0 cm.

She decides to construct a PERCENT_CONF% confidence interval for the true intercept of the regression line between *Age* and *Height* by bootstrapping the regression line 5,000 times.

To assist with this, she first creates a function called `get_intercept`, which takes in a table and two column names as input and returns the intercept of the least squares regression line.

She then writes the following partially completed code to create the confidence interval for the intercept (as an array):

```
intercepts = make_array()
for i in _______:
    BOOT_TBLNAME = PLANTS_TBL._______(_______)
    boot_intercept = get_intercept(_______, _______, 'Height')
    _______ = np.append(_______, _______)
left = _______(_______, intercepts)
right = _______(_______, intercepts)
make_array(_______, _______)
```

Copy/paste the code above and fill in the blanks.

```
intercepts = make_array()
for i in np.arange(5000):
    BOOT_TBLNAME = plants.sample(with_replacement=True)
    boot_intercept = get_intercept(BOOT_TBLNAME, 'Age', 'Height')
    intercepts = np.append(intercepts, boot_intercept)
left = percentile((100-PERCENT_CONF)/2, intercepts)
right = percentile((100+PERCENT_CONF)/2, intercepts)
make_array(left, right)
```

**(e) (2.0 pt)** Suppose that the value of `left` in part (4) above is `-0.03`. Which of the following are conclusions that Alexis can make?

*Select all that apply.*

□ The data are consistent with the hypothesis that the true intercept is 0.

□ The data are consistent with the hypothesis that the true intercept is not 0.

□ The data are consistent with the hypothesis that the true intercept is less than 0.

□ The data are consistent with the hypothesis that the true intercept is greater than 0.

■ There is not enough information to answer.

**(f) (2.0 points)**

Suppose Moira now wants to predict *Age* from *Mass*. She finds that there is a correlation of 0.4 and asserts that there is a relationship between the two variables.

Alexis, however, claims that the observed association in the sample is only due to chance.

**i. (1.0 pt)** Provide a null hypothesis Alexis could use to assess her claims.

**The true correlation between *Age* and *Height* is 0.**

**ii. (1.0 pt)** Provide an alternative hypothesis Alexis could use to assess her claims.

**The true correlation between *Age* and *Height* is not 0. (above 0 is also okay)**

**(g)** **(8.0 pt)** Suppose Moira now makes a scatterplot of *Age* against *Mass*. She notices that instead of a linear trend, there is a quadratic trend (i.e., a parabola that looks like the letter "U").

She decides to try to predict *Age* from *Mass* using a quadratic regression, whose prediction equation looks like the following:

$$age\_predicted = intercept + slope\_1 \times mass + slope\_2 \times mass^2$$

To find the optimal values of $intercept$, $slope\_1$ and $slope\_2$, Moira writes the following partially completed function, which returns the root mean squared error of the quadratic regression for any given values of the intercept and slopes:

```
def rmse(intercept, slope_1, slope_2):
    mass = ____________________
    age = ____________________
    age_predicted = ____________________
    return (____________________) ** 0.5
```

Copy/paste the code above and fill in the blanks.

*Note*: You may reference the PLANTS_TBL table in your solution.

```
def rmse(intercept, slope_1, slope_2):
    mass = PLANTS_TBL.column('Mass')
    age = PLANTS_TBL.column('Age')
    age_predicted = intercept + slope_1 * mass + slope_2 * (mass ** 2)
    return (np.mean((age-age_predicted) ** 2)) ** 0.5
```

**5. (19.0 points) Dogecoin**

To mine (earn) one Dogecoin, a computer must solve a specific math problem, which takes a typical computer an **average** of 10 minutes to solve.

Gilfoyle, an engineer in Palo Alto, has built a gigantic warehouse (called "Anton") that contains hundreds of thousands of computers whose sole purpose is to mine Dogecoin.

To get a sense of how long this mining time varies across his entire warehouse (i.e., the population), Gilfoyle plans to randomly sample some computers and record the amount of time each takes to mine one Dogecoin.

**(a) (2.0 pt)** Suppose that Gilfoyle wants to randomly sample 100 computers from the warehouse without replacement to create a confidence interval for the **population 90th percentile** of Dogecoin mining time.

Which of the following techniques could be applied to help him create this confidence interval?

*Select all that apply.*

□ Central Limit Theorem

■ Bootstrapping

□ Nearest Neighbors

□ Linear Regression

□ Classification

□ None of the above

**(b) (2.0 pt)** Suppose that Gilfoyle wants to randomly sample computers from the warehouse without replacement to create a **95%** confidence interval for the **population mean** of Dogecoin mining time and he knows that the population SD is 40 seconds.

What is the minimum sample size he needs to create a confidence interval that has a width of 8 seconds?

○ 1800

○ 900

○ 800

● 400

○ 225

○ 100

○ There is not enough information to answer

**(c) (2.0 pt)** Suppose that Gilfoyle wants to randomly sample computers from the warehouse without replacement to create a **99.7%** confidence interval for the **population median** of mining time per Dogecoin and he knows that the population SD is 40 seconds.

What is the minimum sample size he needs to create a confidence interval that has a width of 8 seconds?

○ 1800

○ 900

○ 800

○ 400

○ 225

○ 100

● There is not enough information to answer

**(d) (5.0 points)**

Suppose that Gilfoyle randomly samples 500 computers from the warehouse without replacement and uses his sample to create a **95%** confidence interval for the **population mean** Dogecoin mining time.

For the following two questions, assume that the confidence interval he constructs is (550, 590) seconds.

**i. (3.0 pt)** Which of the following can be concluded from the confidence interval above?

■ If Gilfoyle repeats this process 1,000 times, he can expect that roughly 95% of the intervals he creates will contain the true population mean.

□ If Gilfoyle randomly samples 1,000 computers without replacement from the warehouse, he can expect roughly 95% of the computers to take between 550 and 590 seconds to mine one Dogecoin.

□ There is a 95% chance that population mean of Dogecoin mining time is between 550 and 590 seconds.

□ If you randomly sample 100 computers without replacement from the warehouse, you can expect roughly 95% of them to take between 550 and 590 seconds to mine one Dogecoin.

**ii. (2.0 pt)** Gilfoyle thinks his warehouse of computers performs better than the average 10 minutes (600 seconds) that it takes a typical computer in the world to mine one Dogecoin.

Based on the above **95%** confidence interval of (550, 590) seconds, if his p-value cutoff is **5%**, what should he conclude?

■ The data are consistent with the hypothesis that Anton computers mine Dogecoin faster than standard computers do.

□ The data are consistent with the hypothesis that the distirubtion of Dogecoin mining times is the same for both Anton computers and standard computers.

□ The data are consistent with the hypothesis that Anton computers mine Dogecoin slower than standard computers do.

□ There is not enough information to make a conclusion of any kind.

**(e)** **(2.0 pt)** Suppose that Gilfoyle randomly samples 100 computers from the warehouse without replacement. He observes a sample average of 570 seconds for Dogecoin mining time and he also knows that the population SD is 30 seconds.

What is his **68%** confidence interval for the true **population mean** of Dogecoin mining time (in seconds)?

○ (564, 576)

● (567, 573)

○ (569.7, 570.3)

○ (569.4, 570.6)

○ (540, 600)

○ (510, 630)

**(f)** **(2.0 pt)** Suppose that Gilfoyle randomly samples 100 computers from the warehouse without replacement. He observes a sample average of 570 seconds for Dogecoin mining time and he also knows that the population SD is 30 seconds.

Which of the following statements is **guaranteed** to be true?

*Select all that apply.*

☐ At least 68% percent of the computers in the population will have a Dogecoin mining time that is between 3 seconds below and 3 seconds above the population mean.

■ At least 8/9ths of the computers in the population will have a Dogecoin mining time that is between 90 seconds below and 90 seconds above the population mean.

■ At least 75% of the computers in the population will have a Dogecoin mining time that is between 60 seconds below and 60 seconds above the population mean.

☐ At least 75% percent of the computers in the population will have a Dogecoin mining time that is between 510 seconds and 630 seconds.

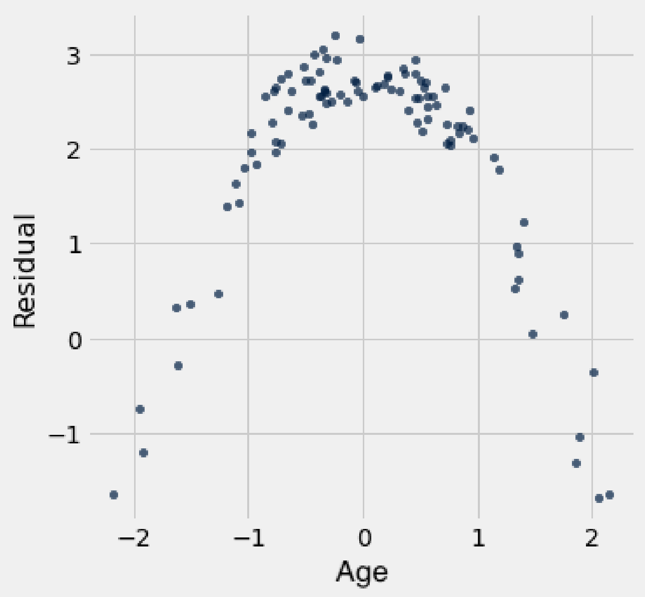
☐ At least 68% percent of the computers in Gilfoyle's sample have a Dogecoin mining time that is between 540 seconds and 600 seconds.

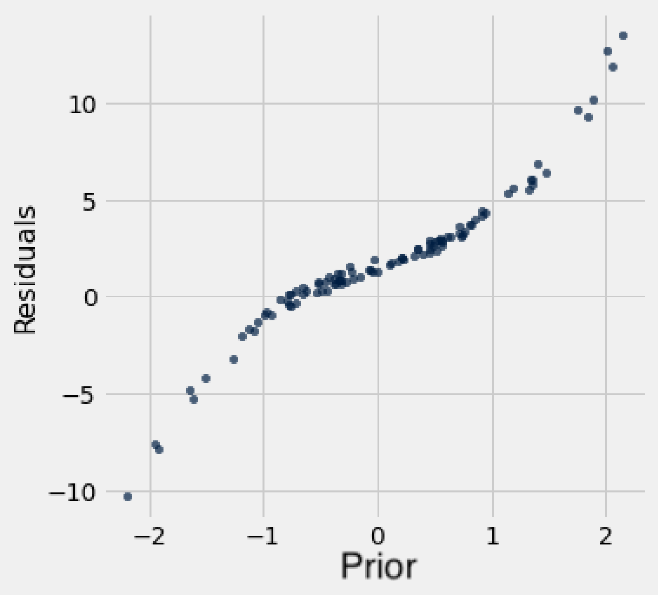
**(g)** **(4.0 points)**

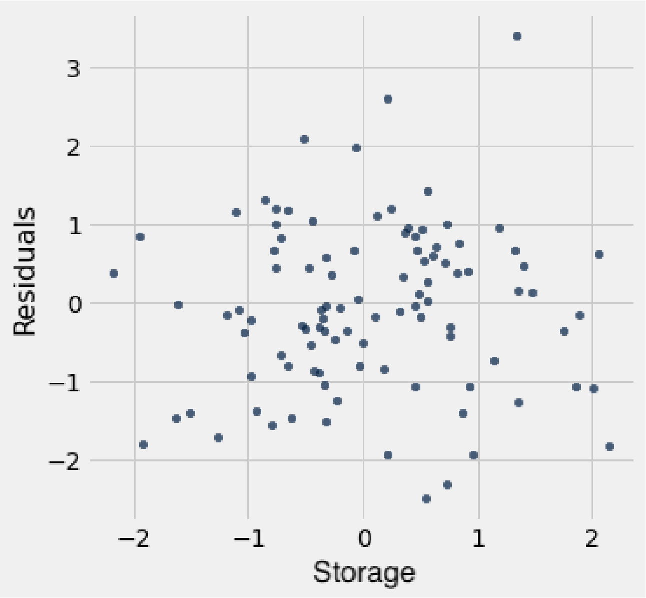
Suppose Gilfoyle tries to predict the Dogecoin mining time of his computers from each of the following four different variables:

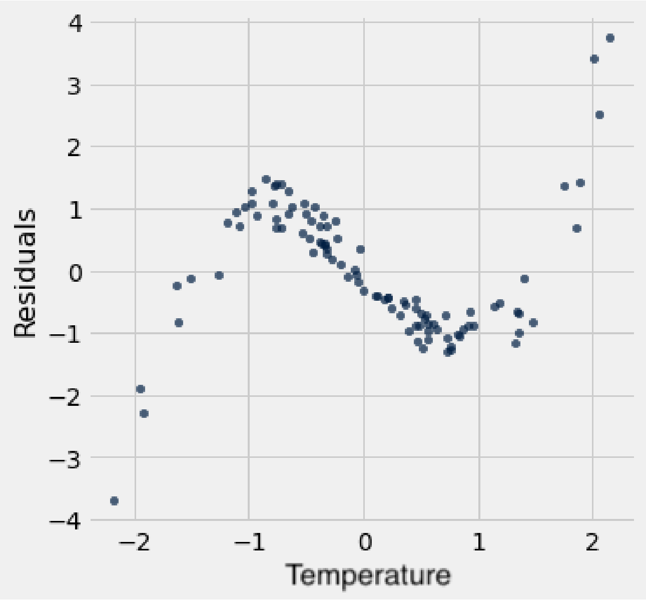
- *Age*: (int) the age of the computer, in days
- *Prior*: (float) the prior Dogecoin mining time of a computer on its most recent mine
- *Storage*: (int) the number of megabytes of storage space left on the computer
- *Temperature*: (float) the computer's temperature, in Fahrenheit

To assess his predictions from each variable, he creates the following plots:









**i. (2.0 pt)** Which of the plots above are impossible residual plots?

*Select all that apply.*

■ *Prior.*

□ *Temperature.*

□ *Storage.*

■ *Age.*

□ None of the above.

**ii. (2.0 pt)** Which of the plots above indicate that linear regression is a good fit?

*Select all that apply.*

□ *Prior.*

□ *Temperature.*

■ *Storage.*

□ *Age.*

□ None of the above.

**6. (24.0 points) Banana Stand**

George Michael Bluth has graduated from high school and is now living his lifelong dream: working at his family's frozen banana stand in Newport Beach.

He's noticed that the volume and makeup of orders during the school year is quite different than during the summer. His uncle, Buster Bluth (an archeological data scientist), decides to help analyze these differences in human behavior by sampling sales data throughout the year and putting this information into the `bananas` table.

Here are the first few rows:

| Date | Customers | Sales | Session | Topping |
|---|---|---|---|---|
| 04/16/20 | 46 | 243 | School | Fudge |
| 06/15/20 | 54 | 233 | Summer | Nuts |
| 09/01/20 | 65 | 254 | School | Pink Candy |
| 10/05/20 | 48 | 278 | School | Nuts |

... (112 rows omitted)

Each row corresponds to a day in 2020 in which the frozen banana stand was open. The table contains the following columns:

- *Date*: (string) the day the stand was open
- *Customers*: (int) the number of total paying customers received that day
- *Sales*: (int) the number of total bananas sold that day
- *Session*: (string) `'School'` if high school is in session on that day, `'Summer'` if it's not
- *Topping*: (string) the most popular topping sold that day

**(a) (15.0 points)**

George Michael believes that banana sales are lower during the school year. He cites that during the school year they sell, on average, 25 fewer bananas per day than during the summer.

However, Buster believes that this difference is only due to chance.

**i. (1.0 pt)** Select the null hypothesis that George Michael should use to assess his claims.

○ On average, the Bluth family sells 25 fewer bananas per day during the school year than during the summer.

● The distribution of bananas sold per day is the same during the school year as it is during the summer.

○ The Bluth family sells fewer bananas per day during the school year than during the summer, due to chance.

○ The Bluth family sells fewer bananas per day during the school year than during the summer.

○ The distribution of bananas sold per day is different during the school year compared to during the summer.

**ii.** **(1.0 pt)** Which of the following are alternative hypotheses that George Michael could use to assess his claims?

*Select all that apply.*

□ The Bluth family sells 25 fewer bananas per day during the school year than during the summer.

□ The distribution of bananas sold per day is the same during the school year as it is during the summer.

□ The Bluth family sells fewer bananas per day during the school year than during the summer, due to chance.

■ The Bluth family sells fewer bananas per day during the school year than during the summer.

■ The distribution of bananas sold per day is different during the school year compared to during the summer.

**iii.** **(9.0 pt)** George Michael decides to simulate 1,000 times from the null hypothesis and store the test statistics from each simulation in the array `test_stats`.

For his test statistic, he chooses the mean bananas sold during school year days minus mean bananas sold during summer days.

He writes the following partially completed code:

```
num_simulations = 1000
test_stats = make_array()

for ___________:
    shuffled_labels = bananas._______________________
    data_with_shuffled = bananas.with_column('Shuffled', shuffled_labels)
    first_mean = np.mean(_______________________)
    second_mean = np.mean(_______________________)
    simulated_stat = _______________________
    test_stats = _______________________
```

Copy/paste the code above and fill in the blanks.

*Recall*: The `bananas` table has the following columns `['Date', 'Customers', 'Sales', 'Session', 'Topping']`.

```
num_simulations = 1000
test_stats = make_array()

for i in np.arange(num_simulations):
    shuffled_labels = bananas.sample(with_replacement = False).column('Session')
    data_with_shuffled = bananas.with_column('Shuffled', shuffled_labels)
    first_mean = np.mean(data_with_shuffled.where('Shuffled', 'Yes').column('Sales'))
    second_mean = np.mean(data_with_shuffled.where('Shuffled', 'No').column('Sales'))
    simulated_stat = first_mean - second_mean
    test_stats = np.append(test_stats, simulated_stat)
```

**iv. (2.0 pt)** After running the code above, George Michael runs the following code:

```
make_array(percentile(0.5, test_stats), percentile(99.5, test_stats))
```

which returns `(-2.5, 2.7)`.

If his $p$-value cutoff is **5%**, which of the following can he conclude about the hypothesis test?

*Select all that apply.*

□ The data are consistent with the null hypothesis.

■ The data are consistent with the alternative hypothesis.

□ The null hypothesis is true.

□ The null hypothesis is false.

□ There is not enough information to make a conclusion of any kind.

**v. (2.0 pt)** In the code in part (3) above, why does George Michael shuffle the labels?

*Select all that apply.*

■ Under the null hypothesis, the value of *Session* has no effect on banana sales.

□ He needs to randomly assign *Session* to establish causality.

□ He needs to ensure that the days of operation are sampled randomly.

□ He needs to protect the privacy of the bananas that were sold on each day.

■ He needs to simulate two groups of days such that their expected banana sales are identical under the null hypothesis.

□ None of the above.

**(b) (9.0 points)**

George Michael suspects that certain toppings are more commonly sold during the summer days than during the school year, while other toppings are less commonly sold during the summer days than during the school year.

Buster thinks any differences in toppings in the sample are only due to chance.

**i. (2.0 pt)** Write a null hypothesis that George Michael could use to assess his claims.

**Bananas sold during the summer have the same *Topping* distribution as those sold during the school year.**

**ii. (2.0 pt)** Write an alternative hypothesis that George Michael could use to assess his claims.

**Bananas sold during the summer have a different *Topping* distribution than those sold during the school year.**

**iii. (1.0 pt)** Select the test statistic that George Michael should use to assess his claims.

● The total variation distance between the *Topping* distribution of summer days and the *Topping* distribution of school year days.

○ The total variation distance between the *Customers* distribution of summer days and the *Customer* distribution of school year days.

○ Mean customers during school year days minus mean customers during summer days.

○ Absolute difference of mean customers during school year days and mean customers during summer days.

○ None of the above.

**iv. (2.0 pt)** Buster simulates 1,000 values of the test statistic and stores these in an array called `test_stats`. Suppose the observed value of the test statistic is `0.27`.

Which of the following Python expressions will return the p-value for this hypothesis test?

*Select all that apply.*

■ np.count_nonzero(test_stats >= 0.27)/len(test_stats)

□ np.count_nonzero(test_stats <= 0.27)/len(test_stats)

□ np.count_nonzero(test_stats == 0.27)/len(test_stats)

■ np.sum(test_stats >= 0.27)/len(test_stats)

□ np.sum(test_stats <= 0.27)/len(test_stats)

□ np.sum(test_stats == 0.27)/len(test_stats)

**v. (2.0 pt)** Buster then takes the `test_stats` array above and runs the following code:

```
percentile(75, test_stats)
```

which returns a value of `0.25`.

If his $p$-value cutoff is **5%** and the observed test statistic is `0.27`, which of the following can he conclude?

*Select all that apply.*

□ The data are consistent with the null hypothesis.

□ The data are consistent with the alternative hypothesis.

□ The null hypothesis is true.

□ The null hypothesis is false.

■ There is not enough information to make a conclusion of any kind.

**7. (30.0 points) Yanay-Bot**

Data 8 staff are attempting to train a machine learning algorithm to replace Yanay's Piazza posting prowess after he graduates.

Staff have collected every Piazza post Yanay has responded to in the past 4 semesters and stored it in the table `piazza` shown here:

| Semester | Title | Body | Image | Response | Label | Helpfuls |
|---|---|---|---|---|---|---|
| fa19 | What time is lecture? | Thanks! | No | Lecture is 10-11am PDT | Resolved | 7 |
| fa19 | Nevermind | Sorry, I don't know how to delete this | No | Marking as resolved! | Resolved | 1 |
| sp20 | Cell is stuck | Been running for the past ten minutes... | Yes | Please try restarting your kernel! | Unresolved | 8 |

... (8764 rows omitted)

The table contains the following columns:

- *Semester*: (string) the corresponding semester
- *Title*: (string) the title of the thread created by a student
- *Body*: (string) the description in the student's original post
- *Image*: (string) `'Yes'` if the post had an image, `'No'` if it did not
- *Response*: (string) the answer Yanay responded with
- *Label*: (string) `'Resolved'` if Yanay marked the post as resolved, `'Unresolved'` if he did not
- *Helpfuls*: (int) the number of "Helpful!" votes Yanay's response received

**(a) (1.0 pt)** Name an ethical and or privacy concern related to the use of this algorithm and dataset. Please limit your answer to 1 sentence.

**Piazza posts can have identifying information that needs to remain private. Anything related to ML might have bias etc**

**(b) (2.0 pt)** Suppose the Data 8 staff would like to understand how *Helpfuls* varies between posts that were marked as resolved and those that were not.

Which of the following would be most appropriate to visualize the relationship between these variables?

○ Scatterplot

○ Pivot Table

○ Total Variation Distance

● Histogram

○ Line Graph

○ Bar Chart

**(c)** **(2.0 pt)** Suppose the Data 8 staff want to understand how the distribution of *Label* varies between posts that had an image and those that did not.

Which of the following could be used to help understand the relationship between these variables?

*Select all that apply.*

□ Scatterplot

■ Pivot Table

■ Total Variation Distance

□ Histogram

□ Line Graph

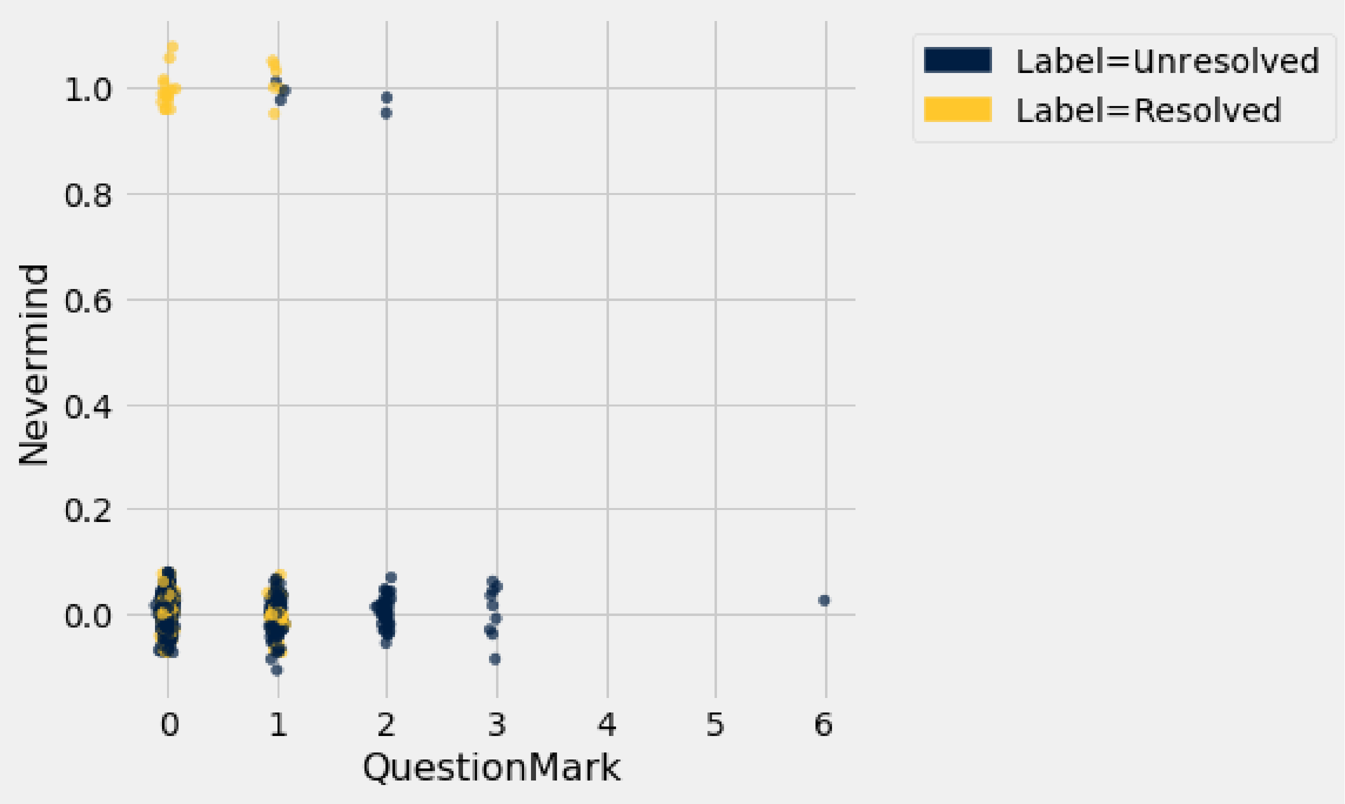
■ Bar Chart

**(d)** **(10.0 points)**

Because of privacy concerns, the Data 8 staff write some Python to add two new columns to the `piazza` table:

- *Neverminds*: (int) the number of occurrences of "nevermind" in the post's *Body*
- *QuestionMarks*: (int) the number of occurrences of "?" in the post's *Body*

The staff then creates the following scatterplot of *Neverminds* against *QuestionMarks*, with different colors based on the value of *Label*.



*Note:* The data are jittered (i.e. every point on the scatterplot has been moved by a small random amount so that the points don't all overlap).

**i. (2.0 pt)** Suppose the staff want to use *Neverminds* and *QuestionMarks* to create a Yanay-Bot that can automatically mark a post resolved (by replying with `'Resolved'`). They write the following partially completed code:

```
def classify(neverminds, question_marks):
    if question_marks > 0.5:
        return 'Unresolved'
    elif _________:
        return 'Unresolved'
    else:
        return 'Resolved'
```

Which of the following Python expressions, if used to fill in the blank above, would result in a classifier that never classifies a training point as `'Resolved'` when the true class is `'Unresolved'`.

*Select all that apply.*

□ `question_marks > 4`

□ `neverminds > 1.0`

■ `neverminds <= 0.5`

□ `neverminds >= 0.5`

■ `True`

□ None of the above.

**ii. (2.0 pt)** Using just *Neverminds* and *QuestionMarks* as their features, the staff can build a classifier that predicts whether a student's post should be marked as resolved with a 0% false negative rate.

*Note*: The false negative rate is the proportion of times a training example with a true class of `'Resolved'` is predicted as `'Unresolved'`.

● True

○ False

**iii. (2.0 pt)** Suppose the staff instead build a $k$-nearest-neighbor classifier with $k = 3$ to predict whether a post should be marked as resolved using *Neverminds* and *QuestionMarks* as its features.

A new Piazza post is made with the following *Body*:

`"Is nevermind the best nirvana album?? Jk, nevermind."`

What would this nearest neighbor classifier predict?

● `'Unresolved'`

○ `'Resolved'`

**iv. (2.0 pt)** What would a nearest neighbor with $k = 11$ predict for the *Body* in the question above?

○ `'Unresolved'`

● `'Resolved'`

**v. (2.0 pt)** The Data 8 staff are worried that this classifier is going to perform very well on the training set but won't do a good job of automatically marking posts as resolved in future semesters. Which of the following would **best** help them estimate the classifier's performance on future semesters?

○ Overfitting

○ Bayes Rule

● Test Set

○ A/B Testing

○ Bootstrapping

**(e)** **(2.0 pt)** Out of the 8,767 posts in the dataset, Yanay marked 2,207 posts as resolved. If the staff were to build a $k$-nearest-neighbors classifier to predict whether a post should be resolved, which values of $k$ will result in always predicting "Do not mark as resolved"?

*Select all that apply.*

■ 8,767

■ 6,941

■ 4,415

□ 3,563

□ 2,209

□ 2,207

□ 9

□ 1

**(f)** **(13.0 points)**

Suppose that the Data 8 staff now want to use $k$-nearest-neighbors to predict the number of *Helpfuls* Yanay's response to a future Piazza post would get based on its *Neverminds* and *QuestionMarks* (i.e., they are now doing a regression instead of classification)

**i.** **(7.0 pt)** To use the $k$-nearest-neighbors to perform this regression, the staff need to first find the $k$ nearest neighbors of the post with respect to *Neverminds* and *QuestionMarks*.

They write a `neighbors()` function, which takes in the following arguments:

- `train`: A three-column table in which the first column is labeled *Neverminds*, the second column is labeled *QuestionMarks*, and the third column is labeled *Helpfuls*. Each row of the table represents a post in the training set.
- `NEWPOST`: An array of length two containing the number of times "nevermind" and "?" appear in the new Piazza post. For example, `array([0, 1])` corresponds to a Piazza post with 0 occurrences of "nevermind" and 1 question mark.
- `k`: The value of $k$ to use for $k$-nearest-neighbors.

The function returns a table containing the $k$ neighbors in `train` that are closest to `NEWPOST`. It is shown, partially completed, here:

```
def neighbors(train, NEWPOST, k):
    questionmark_diffs = __________________
    nevermind_diffs = __________________
    distances = (__________________ + __________________) ** 0.5
    train_dist = train.with_column("Distance", distances)
    return train_dist.__________________
```

Copy/paste the code above and fill in the blanks.

```
def neighbors(train, NEWPOST, k):
    questionmark_diffs = train.column("QuestionMarks") - NEWPOST.item(1)
    nevermind_diffs = train.column("Neverminds") - NEWPOST.item(0)
    distances = (nevermind_diffs ** 2 + questionmark_diffs ** 2) ** 0.5
    train_dist = train.with_column("Distance", distances)
    return = train_dist.sort("Distance").take(np.arange(k))
```

**ii. (6.0 pt)** To generate a prediction for the new Piazza post, the Staff decide to compute the *harmonic weighted average* of the $k$-nearest-neighbors's *Helpfuls*, which is computed as follows:

A. For each of the $k$ nearest neighbors, determine their *weight* using the function `weight()`, which takes in a *Neverminds* value and a *QuestionMarks* value, respectively, as inputs.

B. Compute the *harmonic weighted average* of the neighbors' *Helpfuls*. This involves taking the sum of the weights and dividing this by the sum of the ratios of the weights to the *Helpfuls*. For example, if $k = 3$ and the 3 nearest neighbors have *Helpfuls* of 6, 7, & 8 with weights 0.9, 0.3, & 0.2, respectively, then the *harmonic weighted average* is:

$$\textit{prediction} = \frac{0.9 + 0.3 + 0.2}{\frac{0.9}{6} + \frac{0.3}{7} + \frac{0.2}{8}} = 6.42$$

The staff write a `prediction()` function that takes in the same arguments as the `neighbors()` function (i.e., `train`, `NEWPOST` and `k`) and returns the *harmonic weighted average* of *Helpfuls* among the new Piazza post's $k$ nearest neighbors. It is shown, partially completed, here:

```
def prediction(train, NEWPOST, k):
    NEIGHBORS = neighbors(________________)
    weights = train.apply(________________)
    harmonic_sum = np.sum(________________ / ________________)
    return ________________ / harmonic_sum
```

Copy/paste the code above and fill in the blanks.

```
def prediction(train, NEWPOST, k):
    NEIGHBORS = neighbors(train, NEWPOST, k)
    weights = train.apply(weight, "Neverminds", "QuestionMarks")
    harmonic_sum = np.sum(weights / NEIGHBORS.column("Helpfuls"))
    return np.sum(weights) / harmonic_sum
```

**8. (0.0 points) Just for Fun :)**

**(a)** Prof. Sahai hasn't seen a single episode of one of the following shows. Which is it?

● Firefly

○ Schitt's Creek

○ Community

○ Arrested Development

○ The Office

**9. (0.0 points) Last Words**

**(a)** If there was any question on the exam that you thought was ambiguous and required clarification to be answerable, please identify the question (including the title of the section, e.g., Experiments) and state your assumptions. Be warned: We only plan to consider this information if we agree that the question was erroneous or ambiguous and we consider your assumption reasonable.

**No more questions.**