DATA 8 Summer 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 108 points.

If you encounter any logistical problems during the exam, please contact us at data8berkeley@gmail.com. We will not be answering any questions related to the contents of the exam.

good_luck

(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." Do you agree to follow the honor code on this exam?

◯ Yes

◯ No

1. (15 points) Fun Report

Tyler hosts a weekly podcast called "The Fun Report". At the beginning of each episode, Tyler tells listeners if he is having fun or not. Coley thinks that episodes of the podcast are longer when Tyler is not having fun.

Coley listened to the most recent 30 episodes of the podcast and recorded whether Tyler was having fun or not, and the length of each episode, in minutes, in the table `fun`.

Episode	Having Fun	Length
54	Yes	118
53	No	145
52	No	168

{27 rows omitted}

There are an equal number of episodes where Tyler had fun as where he didn't have fun.

(a) (2 pt) Which of the following lines of code could Coley use to explore his hypothesis in the sample he collected? (Select all that apply)

- ■ `fun.group("Having Fun", np.average).barh("Having Fun", "Length average")`
- ■ `fun.hist("Length", group="Having Fun")`
- ☐ `fun.plot("Episode", "Length")`
- ☐ `fun.scatter("Episode", "Length")`

For the following questions, your test statistic may not involve taking the average (or mean) of any data.

Your answers must be exactly one sentence long. If your response is longer than one sentence, it will receive 0 credit.

(b) i. (2 pt) Write a null hypothesis Coley can use to test his hypothesis.

Episodes where Tyler is having fun have lengths that are drawn from the same underlying distribution as episodes where Tyler is not having fun.

ii. (2 pt) Write an alternative hypothesis Coley can use to test his hypothesis.

The lengths of episodes where Tyler is not having fun are drawn from a distribution with a larger (parameter, depends on your statistic, an example could be median) than the distribution of lengths episodes where Tyler is having fun are drawn from.

iii. (2 pt) Write a test statistic Coley can use to test his hypothesis. As a reminder, your test statistic may not involve taking the average (or mean) of any data.

Your answer needs to compare both the A and B group, and should have extreme values point to the alternative hypothesis. Some examples include:

- **The median length of episodes where Tyler is having fun, minus the median length of episodes where Tyler is not having fun**
- **Differences in percentiles (including max or min)**
- **Difference in sum of the same number of episodes for each group**
- **Anything involving ranking values**

(c) **(3 pt)** Can Coley conclude that Tyler not having fun causes him to record longer podcasts? (Select all that apply)

- [x] No because this was not a randomized control experiment
- [x] No because we don't know what the empirical p value of our hypothesis test is
- [x] No because this was an observational study
- [x] No because our sample was a convenience sample
- [x] No because we don't know our p value cutoff
- [ ] Yes because we shuffled the labels for having fun or not
- [ ] Yes because this was a randomized control experiment
- [ ] Yes because our results are statistically significant

(d) **(4 pt)** Which of the following are valid ways of shuffling labels for this hypothesis test?

Note, there are no intentional typos, all of this code is intended to execute with no error

- [ ]
```
tbl = fun.sample()
```
- [ ]
```
tbl = fun.sample(with_replacement=False)
```
- [ ]
```
lbls = fun.sample().column(‘‘Having Fun’’)
tbl = fun.with_column(‘‘Having Fun’’, fun.column(‘‘Having Fun’’))
```
- [x]
```
lbls = fun.sample(with_replacement=False).column(‘‘Having Fun’’)
tbl = fun.with_column(‘‘Having Fun’’, lbls)
```
- [x]
```
rows = np.random.choice(np.arange(fun.num_rows), fun.num_rows, replace=False)
lbls = fun.take(rows).column(‘‘Having Fun’’)
tbl = fun.with_column(‘‘Having Fun’’, lbls)
```
- [ ]
```
lbls = make_array()

for i in np.arange(fun.num_rows):
    lbl = np.random.choice(make_array("Yes", "No"))
    lbls = np.append(lbls, lbl)
tbl = fun.with_column("Having Fun", lbls)
```
- [ ]
```
lbls = make_array()

for i in np.arange(fun.num_rows):
    if i % 2 == 0:
        lbls = np.append(lbls, "Yes")
    else:
        lbls = np.append(lbls, "No")
tbl = fun.with_column("Having Fun", lbls)
```
- [x]
```
lbls = make_array()

for i in np.arange(fun.num_rows):
    if i % 2 == 0:
        lbls = np.append(lbls, "Yes")
    else:
        lbls = np.append(lbls, "No")
tbl = fun.sample(with_replacement=False).with_column("Having Fun", lbls)
```

2. (11 points) Why-thon

(a) UEFA, the organizing body behind the European Football Championship, operates complex software to manage the teams, spectators, and news-networks. You have been tasked with recreating some features of this software using Python!

i. One of the software developers, Oscar, wrote the following functions, but forgot to document them! What existing functions are these functions equivalent to?

A. (2 pt)

```
def mystery_1(arg):
    return_value = make_array()
    for i in np.arange(len(arg)):
        if i == 0:
            val = arg.item(0)
        else:
            val = arg.item(i) + return_value.item(i-1)
        return_value = np.append(return_value, val)
    return return_value
```

np.cumsum

B. (2 pt)

```
def mystery_2(arg):
    return_value = make_array()
    for j in np.arange(1, len(arg) ):
        return_value = np.append(return_value, arg.item(j) - arg.item(j-1))
    return return_value
```

np.diff

ii. (3 pt) Oscar wrote the following code to announce winners of games. Assume there aren't any ties.

```
def announce(team_1_name, team_2_name, team_1_score, team_2_score):
    winner = “”
if team_1_score > team_2_score:
        winner = print(team_1_name)
    elif team_2_score >  team_1_score:
        winner = print(team_2_name)
    print(winner + “ won by ” + (team_1_score - team_2_score) + “ goals”)
```

The code has 3 errors. List them.

Note: you should be describing three different kinds of errors. For example, if the code has a syntax error involving missing commas between arguments, that counts as a single error, even if it occurs multiple times.

- **print(...) returns none so winner will be None**
- **we need to take the absolute value of the difference of ‘(team_1_score - team_2_score)“**
- **we need to cast that number to a string**

(b) (2 pt) Write a line of code that evaluates to the first ARRAYS_TWO multiples of ARRAYS_ONE starting at ARRAYS_ONE.

For example, the first 3 multiples of 4 are 4, 8 and 12.

```
np.arange(1, ARRAYS_TWO + 1) * ARRAYS_ONE
```

(c) (2 pt) Write a line of code that evaluates to a 3 item array such that the result of calling `sum(arr)` and `sum(np.diff(arr))` are equal. (`arr` will be replaced by the array you are writing).

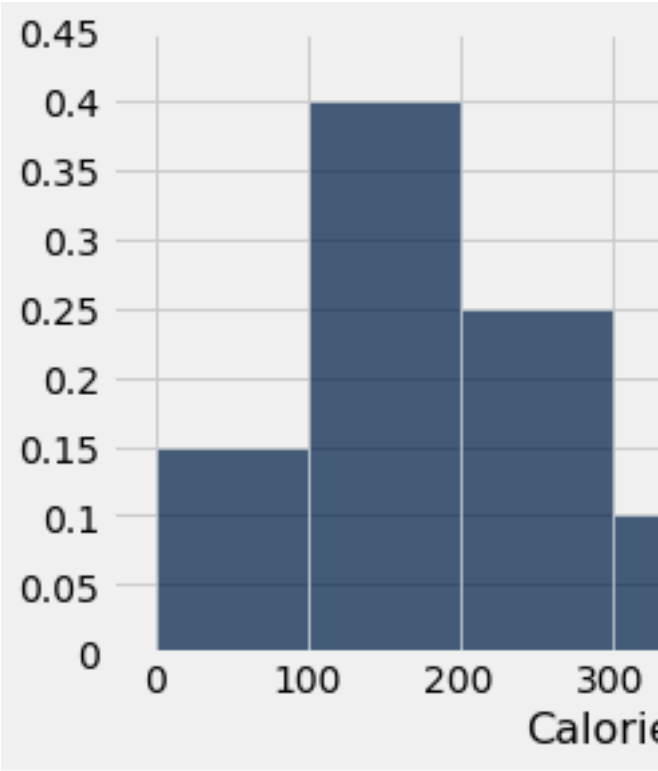
```
make_array(0,0,0)
```

3. (12 points) Holy Frap!

While some of her co-workers like other coffee shops like Peet's Coffee, Wendy is a Starbucks fanatic, but sometimes finds it difficult to decide on a drink when there are so many options. She decides to get data from the complete Starbucks drink menu to analyze the caloric content in each drink. The data is stored in the `starbucks` table. The first few rows from the table are shown below. There are 200 drinks in this table.

Beverage	Calories
Coffee Brewed Coffee	5
Caffè Latte	100
Caffè Mocha	280
Vanilla Latte	110

{196 rows omitted}



Wendy generates a histogram of the Calories column, shown below.

(a) (2 pt) What percentage of drinks are between 200 (inclusive) and 400 (exclusive) Calories? If there's not enough information, please write "not enough information".

35%

(b) (2 pt) What is the absolute difference between the number of drinks in the [0, 100) bin and the number of drinks in the [100, 200) bin? If there's not enough information, please write "not enough information".

50

(c) **(2 pt)** True or False. There are more drinks that have exactly 100 Calories than drinks that have exactly 200 Calories.

- ◯ TRUE
- ◯ FALSE
- ● Not enough information

(d) **(2 pt)** Which statement, regarding the heights of the bins, is correct?

- ● The height of the [0, 100) bin is 0.15% per 1 Calorie.
- ◯ The height of the [0, 100) bin is 0.15% per 100 Calories.
- ◯ The height of the [0, 100) bin is 0.15%.
- ◯ Not enough information
- ◯ None of the these options

(e) **(4 pt)** Wendy does not believe frappuccinos count as real coffee drinks and wants to remove all frappuccinos from the drink menu. Wendy counts that there are 20 frappuccino drinks within [200, 300) Calories. In total, there are 50 total frappuccino drinks on the menu. Wendy removes all frappuccinos from the `starbucks` table and regenerates the histogram of the Calories column. What is the new height of the [200, 300) bin in the histogram? If there's not enough information, please write "not enough information".

.20% per Calorie

4. (22 points) milk and honey

If you order a coffee drink at Peet's Coffee, you can choose between 4 different types of milk to add to your coffee: Whole Milk, Oat Milk, Soy Milk, and Almond Milk.

You can also not add milk to your coffee.

Peet's Coffee reports that of all the people that order coffee, 50% get Whole Milk, 12% get Oat Milk, 10% get Soy Milk, 8% get Almond Milk, and 20% get no milk. Assume that each type of milk ordered is drawn independently from the distribution. The table `milk` shows the distribution below.

Milk	Proportion
Whole Milk	.50
Oat Milk	.12
Soy Milk	.10
Almond Milk	.08
No Milk	.20

Olivia goes to Peet's Coffee's first location (on Vine St in Berkeley) one morning to do her data science homework and sits at a table by the barista. As she overhears some of the coffee orders, she suspects that the distribution of types of milk in coffee orders is actually different from Peet's Coffee's report. She gets ahold of the proportions of each type of milk for all 350 coffee orders in the Vine St. location. The distribution of types of milk is given in the `vine` table below.

Milk	Proportion
Whole Milk	.43
Oat Milk	.14
Soy Milk	.07
Almond Milk	.12
No Milk	.24

(a) Provide a null and alternative hypothesis that Olivia can use to test if the Vine St. Peet's Coffee's distribution of types of milk in coffee orders is the same as all Peet's Coffees.

i. (2 pt) Null hypothesis:

The population distribution of types of milk in coffee orders at the Vine St. Peet's is the same as the population distribution given by Peet's Coffee's report. Any difference is due to random chance.

ii. (2 pt) Alternative hypothesis:

The population distribution of types of milk in coffee orders at the Vine St. Peet's is not the same as the population distribution given by Peet's Coffee's report. Any difference is not due to random chance.

(b) **(2 pt)** Which of the following are valid test statistics for this hypothesis test? Select all that apply. Assume `pop_distribution` is an array of the proportions for the distribution of types of milk in the population and `sample_distribution` is an array of proportions for the distribution of types of milk in a random sample.

- ■ sum(abs(pop_distribution - sample_distribution))
- ☐ abs(sum(pop_distribution - sample_distribution))
- ■ sum(abs(pop_distribution - sample_distribution)) / 4
- ☐ sum(pop_distribution - sample_distribution)
- ☐ abs(sample_distribution.item(0) - .50)
- ■ TVD
- ☐ None of these

`test_statistic` is a function that takes in two arguments and returns a test statistic for this hypothesis test. Both arguments represent distributions in the form of an array. For example, to calculate the observed test statistic, we could run the following code:

```
test_statistic(pop_distribution, obs_distribution)
```

where `pop_distribution` and `obs_distribution` are both arrays of proportions, that are the same length.

(c) **i.** **(3 pt)** Fill in the blank to write a function that simulates one value of the test statistic. Select all that apply. As a reminder, `milk.column(1)` outputs array([.50, .12, .10, .08, .20]) and `vine.column(1)` outputs array([.43, .14, .07, .12, .24]).

```
def one_test_stat():
    return test_statistic(pop_distribution, _________)
```

- ■ sample_proportions(350, make_array(.50, .12, .10, .08, .20))
- ☐ sample_proportions(350, make_array(.43, .14, .07, .12, .24))
- ☐ sample_proportions(350, make_array(0.5, 0.5))
- ☐ np.random.choice(make_array(.50, .12, .10, .08, .20), 350)
- ☐ np.random.choice(make_array(.43, .14, .07, .12, .24), 350)
- ☐ milk.sample(350).column(1)
- ☐ vine.sample(350).column(1)

(d) (6 pt) Copy and paste the following code below and fill in the blanks to generate an array called `simulated_stats` that contains NUM_MILK values of the simulated test statistic and to calculate a p-value. For this hypothesis test, larger values of the test statistic will support the alternative.

```
pop_distribution = _______.column(1)
obs_distribution = _______.column(1)

simulated_stats = ____________
for i in np.arange(NUM_MILK):
    test_stat = one_test_stat()
    simulated_stats = ______________________

OBS_STAT = test_statistic(pop_distribution, obs_distribution)
p_value = ______________________
```

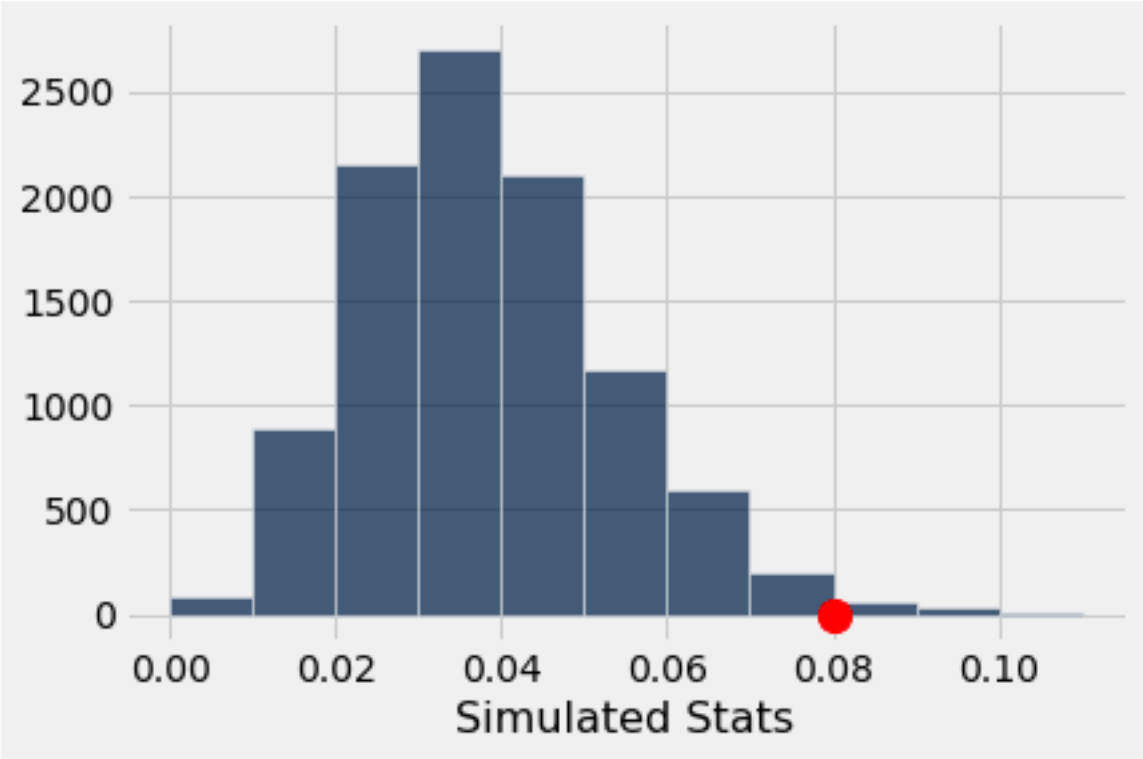
pop_distribution = milk.column(1) obs_distribution = vine.column(1) simulated_stats = make_array() for i in np.arange(NUM_MILK): test_stat = one_test_stat() simulated_stats = np.append(simulated_stats, test_stat) OBS_STAT = test_statistic(pop_distribution, obs_distribution) p_value = np.count_nonzero(simulated_stats >=OBS_STAT)/len(simulated_stats)

(e) (4 pt) Let's say you run the above code and get a p-value of 0.045. Which of the following statements could be true? Select all that apply.

- ■ You reject the null hypothesis with a p-value cutoff of 0.05.
- ☐ You fail to reject the null hypothesis with a p-value cutoff of 0.05.
- ☐ You conclude that the Vine St. location of Peet's Coffee causes customers to order from a different distribution than all the other Peet's Coffee locations.
- ■ You found that you accidentally made a mistake with the initial observation and had to recalculate the observed test statistic. Now the observed test statistic is **higher** than what you initially calculated, so you would reject the null hypothesis with a p-value cutoff of 0.05.
- ☐ You found that you accidentally made a mistake with the initial observation and had to recalculate the observed test statistic. Now the observed test statistic is **higher** than what you initially calculated, so you would fail to reject the null hypothesis with a p-value cutoff of 0.05.
- ☐ You found that you accidentally made a mistake with the initial observation and had to recalculate the observed test statistic. Now the observed test statistic is **lower** than what you initially calculated, so you would fail to reject the null hypothesis with a p-value cutoff of 0.05.

(f) (3 pt) Natalie wants to calculate the p-value and draw a conclusion herself, but she doesn't have access to a computer to run the above code.

We give her the following histogram of simulated statistics (where the `bins = np.arange(0, 0.12, 0.01))` and the observed test statistic of 0.08. Which of the following conclusions would be true? Select all that apply.



- [x] She should reject the null hypothesis with a 7% p-value cutoff.
- [ ] She should fail to reject the null hypothesis with a 7% p-value cutoff.
- [ ] She does not have enough information to make a conclusion with a 7% p-value cutoff.
- [x] She should reject the null hypothesis with a 5% p-value cutoff.
- [ ] She should fail to reject the null hypothesis with a 5% p-value cutoff.
- [ ] She does not have enough information to make a conclusion with a 5% p-value cutoff.

5. (18 points) Multiverse of Mischief

Loki and Mobius are searching for Sylvie. They know Sylvie is physically on one of 6 planets, and temporally in one of 3 different time periods (they can time travel!).

The planets are Earth, Lamentis-1, Asgard, Titan, Mars and Vormir.

The time periods are ancient, modern and future.

Loki and Mobius will choose a random planet and time to look for Sylvie together. Each option has an equal likelihood of being chosen for an attempt to look for Sylvie. Loki and Mobius are guaranteed to find Sylvie if they search the planet she is on in the correct time period.

Leave your answers unsimplified. For example, if your answer is (1/7) + (1/4), leave it in that form.

(a) (1 pt) Given that one of the planets is Earth, what is the chance that Mobius and Loki choose Earth to search, in one attempt?

1/6

(b) (1 pt) What is the chance that Loki and Mobius visit the planet Lamentis-1 (one of the options) in the modern time period, in one attempt?

1/18

(c) (2 pt) What is the chance that Mobius and Loki visit either the planet Earth or the planet Asgard in their first attempt?

(Both planets are options)

1/6 + 1/6

(d) (3 pt) What is the chance that Mobius and Loki find Sylvie in a maximum of two searches, assuming she stays in the same place and time?

Note: Loki and Mobius won't pick the same planet and time combination twice.

1/18 + (17/18 * 1/17)

(e) For the following questions only, assume that Sylvie can choose a different planet and time after each guess Mobius and Loki make, and that Loki and Mobius know this, so they may revisit the same planet and time combination multiple times.

Note: we suggest you determine your answer before looking at the options

i. **(2 pt)** What is the probability that Loki and Mobius visit three different planets in three searches under this new scheme?

- (●) $6 * (1/6) * (5/6) * (4/6)$
- ( ) $(1/6) + (1/6) + (1/6)$
- ( ) $(5/6) + (4/6) - (3/6)$
- ( ) $5 * (4/5) * (3/6) * (2/6)$
- ( ) 1
- ( ) 0
- ( ) $(1/6) * (5/6) * (4/6) * (3/6)$

ii. **(2 pt)** What is the probability that Loki and Mobius visit more than one different time period in three searches?

- (●) $1 - 3 * (1/3)^3$
- ( ) $1 - (1/3)^3$
- ( ) $3 * (1/3)^3$
- ( ) 0
- ( ) $1 - 3 * (1/3)$
- ( ) $1 * 3 * (1/3)^3$
- ( ) 1
- ( ) $(1 - 1/3) + (3 * (1/3)^3)$

iii. **(3 pt)** Ravonna, Mobius's boss, thinks Loki will betray Mobius. Ravonna thinks that every time the two search a planet and time period, Loki has a 70% chance of betraying Mobius. Even if Loki betrays Mobius, they continue searching.

Whether or not Loki betrays Mobius has no effect on the outcome of a search or on the outcome of future searches or future betrayals.

What is the chance that at least one of these events happens in 8 searches?

- **Event 1:** Loki betrays Mobius in at least one of the searches
- **Event 2:** Loki and Mobius find Sylvie in at least one of the searches

- (●) $1 - (17/18 * 0.3)^8$
- ( ) 0
- ( ) $(17/18 * 0.3)$
- ( ) $1 - (17/18 * 0.3)$
- ( ) 1
- ( ) $(1 - (0.3)^8)$
- ( ) $(1 - (17/18)^8)$
- ( ) $(0.3)^8 + ((1 - (0.3)^8) * (17/18)^8)$
- ( ) $(1 - (0.3)^8) * ((17/18)^8)$

iv. (3 pt) What is the chance that at least one of these events happens in 8 searches?

- **Event 1:** Loki doesn't betray Mobius in any of the searches
- **Event 2:** Loki and Mobius don't find Sylvie in any of the searches

- ○ $1 - (17/18 * 0.3)^8$
- ○ 0
- ○ $(17/18 * 0.3)$
- ○ $1 - (17/18 * 0.3)$
- ○ 1
- ○ $(1 - (0.3)^8)$
- ○ $(1 - (17/18)^8)$
- ● $(0.3)^8 + ((1 - (0.3)^8) * (17/18)^8)$
- ○ $(1 - (0.3)^8) * ((17/18)^8)$

6. (26 points) Musical Chairs (and Tables)

Yanay has collected some information about musicians in popular orchestras in the United States.

The table `music` contains one row per musician, and has columns for the musician's `Name`, their `Rank` in the orchestra, the `Instrument` they play, the `Orchestra` they belong to, and their yearly `Salary` in dollars.

`music`:

	Name	Rank	Instrument	Orchestra	Salary
Nayvadius	First	Flute		Atlanta Symphony Orchestra	43670
Aidan	Second	Cello		Boston Philharmonic	28300
Jessie	First	Flute		San Francisco Symphony	41500

{ MUSIC_ROWS rows omitted}

Note: For fill-in-the-blank coding questions, there will be a template for you to follow. You should copy and paste the provided template, then fill in the `__________` to answer the question.

You can put anything inside the blanks, including commas, parentheses, and periods. Note that the length of the blank does not correspond to the length of the code you should write.

(a) (3 pt) Fill in the line of code so that it evaluates to the name of the instrument that is played by the highest paid musician in the table.

```
music._____(_____________)._____(___________)._____(_______)
```

```
music.sort("Salary", descending=True).column("Instrument").item(0)
```

(b) (2 pt) Fill in the following line of code so that it evaluates to the number of musicians who play the instrument `"FIRST_INST"`.

```
music._________(__________)._______
```

```
music.where("Instrument", "FIRST_INST").num_rows
```

(c) (3 pt) Fill in the following lines of code so that the last line evaluates to an array of the 3 orchestras with the fewest members listed in the table.

```
counts = ___________________._____________________(__________)
counts._______(_________)._______(________)._______(________)
```

```
counts = music.group("Orchestra")
counts.sort("count").take(np.arange(3)).column("Orchestra")
```

(d) Budgeting

i. **(2 pt)** Yanay collected some more data on each orchestra's budget. The table `budget` has a column for orchestra names called `Name`, and a column for the orchestra's budget in dollars called `Budget`.

Fill in the following line of code so that it assigns `music_and_budget` to a table with the same rows and columns as the `music` table, and a column called `Budget`.

```
music_and_budget = ________.__________(___________)
```

```
music_and_budget = music.join("Orchestra", budget, "Name")
```

ii. **(2 pt)** Add a column called `"Budget Proportion"` to the `music_and_budget` table which contains the proportion of their orchestra's budget each musician's salary takes.

```
proportions = ________________________________________
music_and_budget = ______.__________(________________)
```

```
proportions = music_and_budget.column("Salary") / music_and_budget.column("Budget")
music_and_budget = music_and_budget.with_column("Budget Proportion", proportions)
```

iii. **(2 pt)** Fill in the line of code so that it makes a visualization of the distribution of budget proportion values, broken down by musicians' rank.

```
________.______(_____________________)
```

```
music_and_budget.hist("Budget Proportion", group="Rank")
```

(e) (4 pt) Fill in the following lines of code to add a column called `"Position"` to the `music` table.

A musician's position represents their rank and the instrument they play. For a musician with the rank second, who plays the Cello, their position should be `"Second Cello"`.

```
def position(_________________________):
    ______________________________
positions = ______.______(____________)
music = ________.________(____________)
```

```
def position(rank, instrument):
    return rank + ' ' + instrument
positions = music.apply(position, "Rank", "Instrument")
music = music.with_column("Position", positions)
```

(f) (4 pt) Katherine wants to know how many musicians of each rank play woodwind instruments at each orchestra.

Fill in the following lines of code so the last line evaluates to a table with one row for each orchestra, a column called `Orchestra` which contains the names of each orchestra, and columns for each rank, which contain the counts of woodwind instrument players of that rank, at the corresponding orchestra. The names of woodwind instruments are given in the `wood_winds` array.

```
wood_winds = make_array("Piccolo", "WOOD_WIND", "Oboe")
wood_winds_tbl = music.____(_____________)
wood_winds_tbl.___________(____________)
```

```
wood_winds = make_array("Piccolo", "WOOD_WIND", "Oboe")
wood_winds_tbl = music.where("Instrument", are.contained_in(wood_winds))
wood_winds_tbl.pivot("Rank", "Orchestra")
```

(g) (4 pt) We want to identify which combinations of rank and instrument are paid the highest among the orchestras.

Create a table with the columns `First`, `Second` and `count`.

The `count` value for each row should correspond to the number of orchestras in which the highest paid musician of rank first played the instrument listed in the `First` column, and the highest paid musician of rank second played the instrument listed in the `Second` column.

Assume the `helpful` function has been defined for you. The `helpful` function returns the first item of the array that it takes as an argument.

Assume every orchestra has musicians of rank first and second listed in the table.

```
def helpful(arr):
    return arr.item(0)
by_salary = music._____________________(_________________)
first_and_second = by_salary._________(_________________)
first_and_second._________________(_____________________)
```

```
def helpful(arr):
    return arr.item(0)
by_salary = music.sort("Salary", descending = True)
first_and_second = by_salary.pivot("Rank", "Orchestra", "Instrument", helpful)
first_and_second.group(["First", "Second"])
```

7. (5 points) Avatar the Last Visualizer

Anna and Chenxi are obsessed with Avatar: The Last Airbender and ask a sample of Berkeley students who their favorite main character is. To get rid of possible spelling errors, each character is assigned a number on the survey (Aang: 1, Katara: 2, Sokka: 3, Toph: 4, Zuko: 5, Uncle Iroh: 6). The results of the survey are stored in the table `avatar`.

The first few rows of the `avatar` table are shown below:

Favorite Character	Student	Episodes Watched	Age When First Watched
4	Kyle	21	9
2	Raymond	32	11
6	Rita	54	15
3	Eddie	44	12
5	Yanay	61	6
6	Jessie	61	7
2	Grace	21	18

... (ROWS_ONE rows omitted)

(a) (1 pt) True or false, `Favorite Character` is a numerical variable

- ○ True
- ● False

(b) (1 pt) The best visualization to characterize the popularity of each character among the students is a/an (Choose only one.):

- ○ Scatter Plot
- ○ Line Plot
- ○ Histogram
- ● Bar Chart

(c) (1 pt) The best visualization to understand the association between `Episodes Watched` and `Age When First Watched` is a/an (Choose only one.):

- ● Scatter Plot
- ○ Overlaid Histogram
- ○ Bar Chart
- ○ Overlaid Bar Chart
- ○ Line Plot

(d) **(2 pt)** Which visualization(s) should we make before testing the hypothesis that students whose favorite character is Zuko (5) started watching the show at an earlier age than students whose favorite character is someone other than Zuko? (Select all that apply.)

- ☐ Scatter Plot
- ■ Overlaid Histogram
- ■ Bar Chart
- ☐ Overlaid Bar Chart
- ☐ Line Plot

8. (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., Why-thon) 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.