Summer 2025
Course Name: Data 6
Instructor: Tsang
Quiz

Print Your Name: _____

Print Your Student ID: _____

You have 110 minutes.

◯ For questions with **circular bubbles**, you may select only one choice.

☐ For questions with **square checkboxes**, you may select one or more choices.

Anything you write outside the answer boxes or anything you *cross out* will not be graded. If you write multiple answers, your answer is ambiguous, or the bubble/checkbox is not entirely filled in, we will grade the worst interpretation.

You are disallowed from using the following: for loops, while loops, if statements, boolean operators, ternary operators, lambdas.

You may only write one statement per line (i.e. you may not use semicolons).

Trick or Treat!

The Data 6 staff decide to go trick-or-treating! Below is a dataset showing the candy they received on their night out:

- Candy Name represents the name of the Candy Brand
- Sugar per Serving represents the number of grams of sugar per serving.
- Received represents the number of servings the Data 6 staff received.
- Type represents what category (between Gummy, Chocolate, and Other) that Candy falls into.
- Yumminess level is a rating (from Gross to "this is the best thing to ever have graced planet Earth) representing how good Maryam thinks the candy is.
- Zip code represents the zip code where the majority of that particular candy came from.

Variable Types

What is the variable type of the following columns? You may write either “Ordinal”, “Nominal”, “Discrete”, or “Continuous”.

“Candy name” - Nominal

“Received” - Discrete

“Yumminess level” - Ordinal

“Zip Code” - Nominal

Sampling Methods

Suppose we wanted to create this dataset to answer the question: "Within the City of Berkeley, which candy is the most popular to give out during Halloween?" We collected this data using different approaches. Match each approach to the corresponding sampling method. You may choose between a Convenience Sample, Snowball Sample, and a Simple Random Sample.

We decided to only go to the Zip Codes in District 7 (in Berkeley, there are a total of 8 districts) since the rest are too far from campus.
Convenience sample
We have a detailed map of all of the different houses and Zip Codes in Berkeley, and we use a random number generator to decide which ones to visit.
Simple Random Sample
After visiting the first house, we ask them if they know any other houses that are giving out candy. We use that to determine the next house to visit.
Snowball sample

Visualizations

For each visualization, note which is most appropriate between a: bar chart, histogram, scatter plot, line plot, or choropleth map.

I want to see the most popular candy in each Zip Code.
Choropleth map

I want to see the distribution of Sugar per Serving across all the different types of Candy.
Histogram
I want to see if there's a correlation between the Sugar per Serving of a given candy, and the amount of it I received.
Scatter Plot

Suppose we want to use an overlaid histogram to visualize the distribution of “Received”, one for each type of candy we received. We want the raw counts of each candy on the y-axis (rather than the density), with three bins from [0, 10), [10, 20), and [20, 30). Fill in the line of code below to create this plot:

```python
data.hist("Received", density = False, bins = np.arange(0, 40, 10),
group = "Type", overlay = True)
```

How many encodings would be present in the visualization that we just created above?

- [ ] 1
- [ ] 2
- [x] 3
- [ ] 4
- [ ] 7

Given these bin sizes, can we figure out how many values fall within the range…

… [0, 10)?

- [x] yes
- [ ] no

… [0, 20)?

- [x] yes
- [ ] no

…[5, 15)?

- [ ] yes
- [x] no

Briefly describe why it may not be appropriate to group based on some numerical column like "Sugar per Serving."

There may be too many unique values in this column, resulting in visual clutter.

Data Types

Let's verify that the values in the "Sugar per Serving" column are indeed Integers and not strings. Write a Python expression below that displays the data type of each element in the "Sugar per Serving" column.

```python
type(data.column('Sugar per Serving')[0])
```

Turns out, the values in our "Sugar per Serving" column are all strings! Replace the values in the "Sugar per Serving" column such that the values are all integers, not strings.

```python
data = data._____(blank a)_____(
    "Sugar per Serving",
    data._____(blank b)_____(
        _____(blank c)_____,
        "Sugar per Serving"
    )
)
```

1. What goes in _____(blank a)_____? (3 points)
   - [ ] column
   - [ ] relabeled
   - [ ] sort
   - [ ] apply
   - [ ] join
   - [ ] with_column
   - [ ] with_row
   - [ ] where

2. What goes in `____(blank b)____`? (3 points)
   - [ ] column
   - [ ] relabeled
   - [ ] sort
   - [ ] apply
   - [ ] join
   - [ ] with_column
   - [ ] with_row
   - [ ] where
3. What goes in `____(blank c)____`? (3 points)

```python
int
```

Nicknames

Let's augment our dataset with a new column, "Nickname", which contains the first word of each candy's name. For example, "Sour Patch Kids" should turn into "Sour" and "M&Ms" should stay as "M&Ms".

Begin by implementing name_to_nickname, which takes in a string representing the candy name as an input, and returns the nickname of that candy.

```python
def name_to_nickname(name):

    return name.split()[0]
```

Suppose we call name_to_nickname with the input "Jedi", then attempt to print out the name as follows:

```python
name_to_nickname("Jedi")
print(name)
```

What would Python display?

- [ ] Jedi
- [x] Nothing, an error would occur

In 10 words or less, justify your answer from above.
name only exists within the scope of the name_to_nickname function.

Finally, create and add the new “Nickname” column to our dataset, modifying data.

```python
data = data.with_column("Nickname", data.apply(name_to_nickname, "Candy name"))
```

Nom Nom Nom

We want to figure out which candy is the healthiest to binge eat ALL of in one sitting (defining healthiest as the least grams of sugar overall). Start by adding a new column to data called

Total Grams which represents the total amount of sugar consumed if we were to eat all of it. For example, eating Sour Patch Kids would have a total of 24 * 15 = 360 grams of sugar.

```python
data = data.with_column("Total Grams",data.column("Received") *
data.column("Sugar per Serving"))
```

Finally, select all valid expressions that would print out the name of the healthiest candy to consume all of. Select all that apply.

- [ ] data.sort("Total Grams").column("Candy Name")[0]
- [ ] data.sort("Total Grams", descending = True).take(data.num_rows).column("Candy Name")[0]
- [ ] data.sort("Total Grams", descending = True).take(data.num_rows - 1).column("Candy Name")[0]
- [ ] data.take(0).sort("Total Grams")[0]
- [ ] data.select("Candy Name").sort("Total Grams")[0]

Conceptual Joins

Suppose we decided to join a table tbl1 with another table, tbl2, to get more information. If tbl1 has n rows and m columns and tbl2 has x rows and y columns, what is the:

maximum number of rows in the resulting table? Write your answer in terms of n, m, x, and y.
n * x

minimum number of rows in the resulting table? Write your answer in terms of n, m, x, and y.
0

Chocolate or Gummy?

We wonder if we received more Chocolate candies or Gummy candies. Since we don't really care as much about the other types, let's first filter out any Candies that aren't "Chocolate" or "Gummy", and call it "filtered_candies".