Data 8 Spring 2024

Functions, Table Methods

Discussion 05

February 2024

Welcome to the Project 1 Discussion Worksheet! This week we will cover functions and table methods like `.group` and `.pivot`. As a reminder, make use of the Python Reference and the Table Functions Visualizer on the Data 8 website.

1. Fun(ctions)

a. After learning about them in Data 8, Conan wants to write a function that can calculate the hypotenuse of any right triangle. He wants to use his function to assign `C` to the hypotenuse of a triangle with side lengths `A`, `B`, and `C`. However, he's made a few mistakes. Which ones can you identify?

Hint: There are 5 unique issues. Assume that `numpy` has been imported as `np`.

```
def hypotenuse(a, b)

    """Returns the length of the hypotenuse of a right triangle, the square root of a

    squared + b squared."""

    squares = make_array(side1, side2) * 2

    sum = sum(squares)

    squareroot = np.sqrt(sum)

    print(squareroot)
```

b. Write a function that takes in the following arguments:

- `tbl`: a table
- `col`: a string, name of a column in `tbl`
- `n`: an int

The function should return a table that contains the rows that have the `n` *largest* values for the specified column.

```
def top_n(tbl, col, n):

    sorted_tbl = ____________________________________________

    top_n_rows = ____________________________________________

    return ______________________
```

2. Table Matchmaking

Shown below are the `chocolates` and `nutrition` tables.

Color	Shape	Amount	Price ($)
Dark	Round	4	1.30
Milk	Rectangular	6	1.20
White	Rectangular	12	2.00
Dark	Round	7	1.75
Milk	Rectangular	9	1.40
Milk	Round	2	1.00

Type	Calories
Dark	120
Milk	130
White	115
Ruby	120

Match the following table method calls to the resulting descriptions of tables.

Letter	Function Call
A	`chocolates.group("Shape")`
B	`chocolates.group("Shape", max)`
C	`chocolates.group(["Shape", "Color"], max)`
D	`chocolates.pivot("Color", "Shape", "Price($)", max)`
E	`chocolates.join("Color", nutrition, "Type")`
F	`chocolates.group(["Shape", "Color"])`

Number	Columns	# of Rows
1	Shape, Color max, Amount max, Price ($) max	2
2	Shape, Dark, Milk, White	2
3	Shape, Color, Amount max, Price ($) max	4
4	Color, Shape, Amount, Price ($), Calories	6
5	Shape, count	2
6	Shape, Color, count	4

A: _________
B: _________
C: _________
D: _________
E: _________
F: _________

3. Squirrel!

The table `squirrel` below contains some information on reported squirrel sightings across the UC Berkeley campus:

- **Squirrel ID** (int): unique identification number for each unique squirrel
- **Location** (string): common name of the nearest campus landmark where the squirrel was spotted
- **Month** (int): numerical representation of the month when the squirrel was spotted
- **Day** (int): day of the month when the squirrel was spotted
- **Year** (int): year when the squirrel was spotted

Squirrel ID	**Location**	**Day**	**Month**	**Year**
2937	Wheeler Hall	17	3	2022
8421	East Asian Library	28	9	2021
472	Etcheverry Hall	7	1	2022
239	Campbell Hall	4	10	2021
2937	Moffitt Library	7	6	2022

a. Identify the categorical variables in the table.
Hint: Try to imagine taking the average of a variable and see if that quantity makes sense. If it doesn't make sense, the variable is categorical.

b. Jessica wants to find the best location where she is most likely to find a squirrel. Write a line of code that evaluates to the location with the most squirrel sightings.

c. Jessica is interested in how many squirrels were sighted at every location during every month. Create a table called `sightings` where each cell contains the number of squirrel sightings that occurred in 2022 at each location during each month. Note: Each row should be a different location.

```
squirrels_2022 = ______________________________________________

sightings = ______________________________________________
```