Honors Class06 Activity: Practice with Tables – Breakfast Cereals

Team Members: _____

Learning Objectives

- Read data into tables
- Understanding metadata
- Manipulating data tables

You will be using data tables for just about everything the rest of the semester, so let warm-up today with exercise designed to review many of the basic table operations. And what could be better on a table than some breakfast cereal? We'll start by loading a data from a file.

To save you a lot of typing, there is a starter Jupyter notebook under "class activities." Open the one for class_06. At the start you are provided with metadata.

Problem 1: Suppose you are collecting water samples to access stream quality. Give some examples of metadata might you record.

Recall that an array has to contain data of all the same type. Most of the data in this table is numeric.

Problem 2: Which columns contain an arrays of strings?

Enter the code below to extract the rating column from the table as an array.

```
ratings = cereal.column("rating")
ratings
```

Problem 3: What is are the average, maximum and minimum ratings for the cereals in this data set?

To sort from highest to lowest rating, you can use the sort function with the descending=True option.

Problem 4: Which cereal has the highest rating?

After you've answered the questions above, continue exploring the data table. You might try to find the cereal with the most sugar, or the one with the highest fiber content.

Problem 5: What other interesting facts can you find about the cereals in this data set?

Filtering based on a condition

This comes up a lot. You want to pull the rows from a table based on some condition. For this, you need the where() method of data tables.

For example, suppose we want just the cereals with more than 8 grams of fiber per serving.

```
high_fiber = cereal.where("fiber", are.above(8))
high_fiber
```

Ah, bran cereals! No surprise they are the ones highest in fiber. Create a table with just the hot cereals.

Problem 6: Which cereals are hot cereals?

Applying multiple conditions

To filter on multiple conditions, simply apply them one at a time.

Let's look for the cereals high in sodium, and suger – the ones you probably shouldn't eat. First, let's find the range of values for each. The code for this is provided for you in the notebook.

Negative Sugar!? If there is such a cereal it would be extremely popular as a diet food! I'm guessing this is an error. Data sets often have mistakes. *Remember this when you do your own data collection and analysis.*

Enter the code below to find any cereals with negative sugar content.

```
cereal.where("sugar", are.below(0))
```

Oatmeal has both negative sugar and negative carbs. Who knew?

Problem 7: Ignoring this obvious flaw in the data. Let's see if you can find a cereal(s) with more than 230 milligrams of sodium, and more than 10 grams of sugars.

(The ones low in fiber too are truly without redeeming value.)

Discussion Questions

The activity emphasizes that tables are comprised of columns, and columns are arrays of data. The data in a coliumn are all of the same type.

Problem 8: At what point does data analysis stop being about computation and start being about judgment?

Point to a specific moment in this activity where that shift occurs.

Problem 9: The dataset includes negative values for sugar and carbohydrates.

- `What are some realistic ways such errors could arise?`
- `Should these rows be removed, corrected, or flagged? Why?`

Problem 10: Suppose you were to design a data table to track your daily activities (e.g., studying, exercising, socializing, etc.). What columns would you include in this table? Justify your choices.