Contracts: Making Tables and Displays

Students learn about functions for sorting and counting data in tables, then are introduced to one-variable displays.

Lesson Goals	Students will be able to… • sort & count columns of data in the programming environment • make pie charts, bar charts, histograms and box plots in Pyret • understand how to read contracts for any Pyret function
Student-facing Lesson Goals	• Let's learn some functions for counting and sorting data. • Let's learn how to make data displays in Pyret!
Prerequisites	• Simple Data Types
Materials	• PDF of all Handouts and Page • Animals Starter File • Lesson Slides • Printable Lesson Plan (a PDF of this web page)
Supplemental Materials	• Additional Printable Pages for Scaffolding and Practice

Glossary

categorical data :: data whose values are qualities that are not subject to the laws of arithmetic
contract :: a statement of the name, domain, and range of a function
data type :: a way of classifying values, such as: Number, String, Image, Boolean, or any user-defined data structure
domain :: the type or set of inputs a function expects, i.e., the independent variable(s) that govern the output of the function
name :: how we refer to a function or value defined in a language (examples: +, *, star, circle)
quantitative data :: number values for which arithmetic makes sense
range :: the type or set of outputs that a function produces, i.e., the dependent variable(s)

Functions for Working with Tables *20 minutes*

Overview

Students will be introduced to functions for working with tables, including: `sort`, `count` and `first-n-rows`. In the process they will review simple ***data types*** and be introduced to ***contracts*** for describing the ***Domain*** and ***Range*** of any function.

If you have time and interest in digging deeper into contracts, we have a whole lesson focused on contracts for image-producing functions, which students find highly motivating!

Launch

- Open the Animals Starter File and click "Run".
- In the Interactions Window on the right type `animals-table` and hit "Enter" to see the the table.
- What do you Notice? (sample answers below)
  - *We only see the first 10 rows of the table and would have to click to see the remaining 22 rows.*
  - *The animals aren't listed in any particular order.*
- What do you Wonder? (sample answers below)
  - *Is it possible to alphabetize the list, or sort it in other ways?*

(If this is the first time your students are seeing a table in code.pyret.org (CPO), you may also want to acknowledge line 7 of the Definitions Area, where `animals-table` is defined along with the names of the 8 columns.)

Investigate

Turn to Functions for Tables and complete numbers 1-6.

- What did these `sort` expressions do?
  - *They took in the `animals-table`, the "Name" column, and true or false… producing a new table, sorted alphabetically (A-Z for true and Z-A for false)*
- Just below question 6, we see the Contract for `sort`
- What does it mean that the Domain of `sort` is Table, String, Boolean?

- *The functions needs a Table, a String (describing a column), and a Boolean (an order to sort the column by) in order to do its job.*

- What does it mean that the Range of `sort` is Table?
  - *The function produces a new table.*

As you can imagine, there are many other functions that work with tables, each with a different set of inputs. For each of these functions, we need to keep track of three things:

1. **Name** — the name of the function, which we type in whenever we want to use it
2. **Domain** — the type(s) of data we give to the function
3. **Range** — the type of data the function produces

The ***Name***, ***Domain*** and ***Range*** are used to write a ***Contract***.

Contracts are the *instruction manual* for functions

The Domain in the Contract for `sort` tells us exactly what kind of inputs it needs, and in what order. The Range tells us exactly what we'll get back.

Complete question 7-10 on Functions for Tables.

- How did `sort` work with quantitative columns?
  - *true sorted the values from least to greatest. false sorted the values from biggest to smallest.*
- What new questions did you think of that could be answered using `sort`?

Once you know how to read Contracts, you can easily use all the tools in our programming language. Let's explore the `count` and `first-n-rows` functions, by completing Functions for Tables (continued).

Common Misconceptions

Students are likely to think that `sort` *changes the table*, when instead it *produces a new, sorted table.* Encourage students to say out loud what they think they will get if they type `animals-table` after evaluating `sort(animals-table, "name", true)`. By testing their hypothesis, students who are surprised at the outcome are much more likely to remember the difference later on.

Synthesize

- How is the function of `count` different from `sort`?
  - `sort` *made a new table that was reordered*
  - `count` *made a new table that summarized the data from a column of the original table!*
- How does the function `first-n-rows` work?
  - *It makes a new smaller table with the number of rows you type into the expression.*
- Does anyone have any questions about the functions or expressions you've worked with today?
- Where have you seen tables summarizing counts in the real world?
- How else do journalists and newscasters display summaries of data besides in tables?
  - *Ideally someone will say bar charts!*

Composing with Circles of Evaluation *15 minutes*

Overview

Students learn to work with more than one function at once, by way of Circles of Evaluation, a visual representation of the underlying structure.

Launch

What if we wanted to see the ten youngest animals? How could the `first-n-rows` and `sort` functions work together? What order should we use the functions in?

Investigate

One way to organize our thoughts is to diagram what we want to do, using the Circles of Evaluation. The rules are simple:

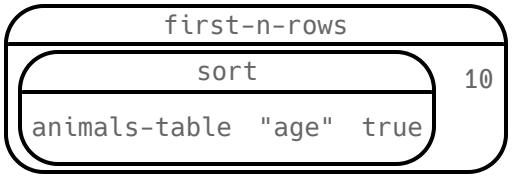
1) Every Circle of Evaluation must have one - and only one! - function, written at the top.

2) The arguments of the function are written left-to-right, in the middle of the Circle.

Values like Numbers, String, and Booleans are still written by themselves. It's only when we want to *use a function* that we need to draw a Circle, and write the values inside from left-to-right.

3) Circles can contain other Circles!

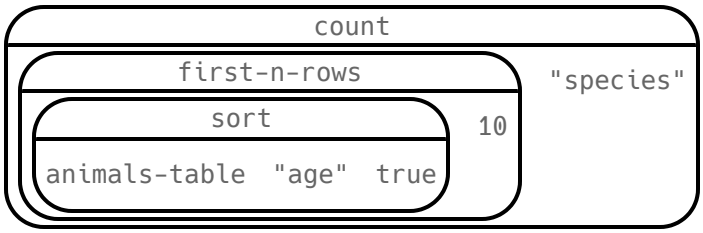
If we want to see the ten youngest animals, our diagram would look like this.



To convert a Circle of Evaluation into code, we start at the outside and work our way in. After each function we write a pair of parentheses, and then convert each argument inside the Circle. The code for this Circle of Evaluation would be:

```
first-n-rows(sort(animals-table, "age", true), 10)
```

If we wanted to get extra fancy and see the species count for the youngest ten animals, we could add another layer to our Circle of Evaluation.



That would translate to the following code:

```
count(first-n-rows(sort(animals-table, "age", true), 10), "species")
```

- Turn to Circles of Evaluation: Count, Sort, First-n-rows.
- Draw Circles of Evaluation and write code for each of the given scenarios.
- Then test your code out in Animals Starter File.

Synthesize

- What did you Notice?
- What did you wonder?

Functions for Making Displays *20 minutes*

Overview

Students will be introduced to functions for making one-variable displays in Pyret, including: `pie-chart`, `bar-chart`, `box-plot` and `histogram`.

The goal here is for students to become familiar with using Contracts to write expressions that will produce displays. Knowing how to *make* a histogram doesn't mean a student really *understands* histograms!

Once students know how to use Contracts to write expressions to make these displays, we have dedicated, in-depth lessons focused on understanding Bar and Pie Charts, Histograms, Visualizing the "Shape" of Data, Box Plots, Scatter Plots, Linear Regression, Advanced Displays, etc.

Launch

The `count` function summarized the data for a single variable in a new table. But the same information could be communicated as a picture! This is called data visualization, and Pyret has functions that can make displays for us!

Investigate

Turn to Exploring Displays. Let's look at the first function together.

- What is the name of the function?
  - *bar-chart*
- What is the Domain of the function?
  - *Table, String*
- What is the Range of the function?
  - *Image*
    - *Take a minute and see if you and your partner can write an expression that will generate a* `bar-chart`.
- Did `bar-chart` consume a categorical or quantitative column of data?
  - *categorical*
- What does the resulting display tell us?

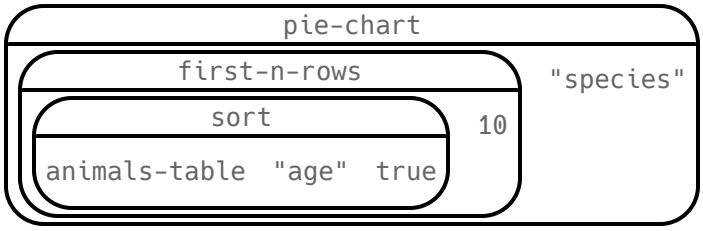
- Make a sketch of the display you just built in Pyret.

- Then work to complete Exploring Displays, generating each of the other 3 displays. Some of them may be new to you - you are not expected to be an expert in them yet, but you should be able to figure out how to use the Contract to get them building!

Optional:

If your students are already familiar with scatter plots, linear regression plots, and line graphs, you may also want to have them complete Exploring Displays (2).

Just as we can use Circles of Evaluation to help us combine `sort`, `count`, and `first-n-rows`, we can put Circles of Evaluation to work to help us write code to build more specific displays.



- What expression would this Circle of Evaluation generate?
  - `pie-chart(first-n-rows(sort(animals-table, "age", true), 10), "species")`
- What would be the resulting display?
  - *a pie chart showing the species of the 10 youngest animals*

Optional: If your students would benefit from seeing a few more examples before drawing their own Circles of Evaluation, have them Match Display Descriptions to Circles of Evaluation.

- Have students complete Circles of Evaluation: Composing Functions to Make Displays.
- Displays are often most interesting when compared with other displays. For example, we may want to see how the age range of the animals adopted quickly compares to the age range of all the animals or of the animals that were adopted slowly. **Consider what display it might be interesting to compare each of the displays on this page with.**

Optional: For more practice making tables and displays by composing functions, have students complete Circles of Evaluation: Composing Functions to Make Displays (2)

Synthesize

- Which displays worked with categorical data?
  - `pie-chart` *and* `bar-chart`
- Why might you choose a bar chart over a pie chart or vice versa?

- `pie-chart` *only makes sense when you have the full picture, since it's representing the proportion of the whole*
- `bar-chart` *shows the count*

- How are bar charts and histograms different?
  - `bar-chart` *summarizes* ***categorical*** *data. Each bar represents the count of a specific category.*
  - `histogram` *displays the distribution of* ***quantitative*** *data across the range.*

Additional Exercises

- Composing Functions: Match Display Descriptions to Circles of Evaluation