Data Displays and Lookups

Students continue to practice making different kinds of data displays, this time focusing less on programming and more on using displays to answer questions. They also learn how to extract individual rows from a table, and columns from a row.

Prerequisites	Applying Functions
Relevant Standards OK K12CS CSTA NGSS CC-Math	*Select one or more standards from the menu on the left (⌘-click on Mac, Ctrl-click elsewhere).*
Lesson Goals	Students will be able to... • Given a human-language request for a data display involving the entire Animals Dataset, break it down into parts and generate the display. • Given a Table, use the `row-n` method to extract any Row from that table • Given a Row, use the column lookups to extract the value of any column in the Row
Student-facing Lesson Goals	• Let's practice making graphs and retrieving information from tables.
Materials	• Lesson Slides (Google Slides) • Computer for each student (or pair), with access to the internet • Student workbook, and something to write with • All students should log into CPO and open the "Animals Starter File" they saved from the prior lesson. If they don't have the file, they can open a new one
Preparation	• Make sure all materials have been gathered • Decide how students will be grouped in pairs
Supplemental Resources	
Language Table	

Types	Functions	Values
Number	`num-sqrt, num-sqr`	`4, -1.2, 2/3`
String	`string-repeat, string-contains`	`"hello", "91"`
Boolean	`==, <, <=, >=, string-equal`	`true, false`
Image	`triangle, circle, star, rectangle, ellipse, square, text, overlay, bar-chart, pie-chart, bar-chart-summarized, pie-chart-summarized`	
Table	`count`	

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

method :: a function that is only associated with an instance of a datatype, which consumes inputs and produces an output based on that instance

quantitative data :: number values for which arithmetic makes sense

Displaying Data

20 minutes

Overview

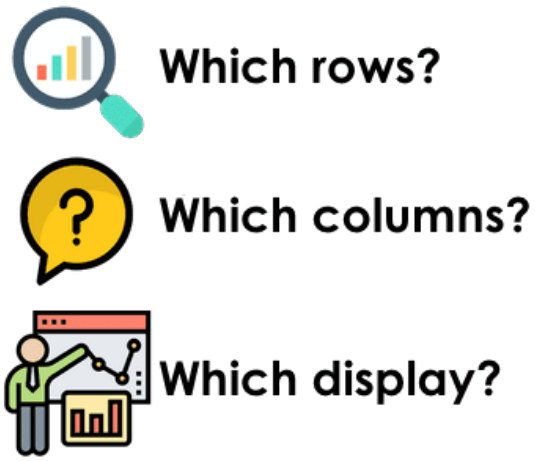
Students get some more practice applying the plotting functions and working with Contracts, and begin to shift the focus from *programming* to *data visualization* . This activity stresses a hard programming skill (reading Contracts) with formal reading comprehension (identifying key portions of the sentence).

Launch

The Contracts page in the back of students' workbooks contains ***contracts*** for many plotting functions.
Suppose we wanted to generate a display showing the ratio of fixed to un-fixed animals from the shelter? How do we go from a simple sentence to working code that makes a data display?

> To make a data display, we ask "Which Rows?", "Which Column(s)?", and "What Display?"

1. We start by asking **which rows** we're talking about. In this case, it's all the animals from the shelter.
2. We also need to know **which column(s)** - or "which variable(s)" - we are displaying. In this case, it's the `fixed` column.
3. Finally, we need to know **which display** we are using. Is it a histogram? Bar chart? Scatter plots are essential for displaying relationships *between* columns, but the other displays only deal with one column. Some displays work for ***categorical data***, and others are for ***quantitative data***.



Once we can answer these questions, all we need to do is find the Contract for that display and fill in the Domain!
To display the categorical data, we can choose between pie and bar charts. Which one of these two is best, and why?

Investigate

Do you know what kind of data is used for each display?

> Turn to What Display Goes with Which Data? (Page 18), and see if you identify what kind of data each display needs!

Let's get some practice going from questions to code, making visualizations.

> Turn to Data Displays (Page 19), and see if you can fill in these three parts for a number of data display requests. When you're finished, try to make the display in Pyret using the appropriate function.

Synthesize

Debrief the activity with students.
Optional: As an extension, have students break into teams and come up with additional Data Display challenges, then race to see which team can complete the other team's challenges first!

Row and Column Lookups

30 minutes

Overview

Students learn how to access individual rows from a table in Pyret, and how to access a particular column from those rows.

Launch

Have students open their saved Animals Starter File (or make a new copy), and click "Run".
Tables have special functions associated with them, called ***Methods***, which allow us to do all sorts of things with those tables. For example, we can get the first data row in a table by using the `row-n` method: `animals-table.row-n(0)`

tables. For example, we can get the first data row in a table by using the `.row-n` method: `animals-table.row-n(0)`

> Don't forget: data rows start at index *zero*!

For practice, in the Interactions Area, use the `row-n` method to get the second and third data rows.

What is the Domain of .row-n? What is the Range? Find the contract for this method in your contracts table. A table ***method*** is a special kind of function which always operates on a specific table. In our example, we always use .row-n with the animals table, so the number we pass in is always used to grab a particular row from animals-table.

Pyret also has a way for us to get at individual columns of a Row, by using a Row Accessor. Row accessors start with a Row value, followed by square brackets and the name of the column where the value can be found. Here are three examples that use row accessors to get at different columns from the first row in the animals-table:

```
animals-table.row-n(0)["name"]
animals-table.row-n(0)["age"]
animals-table.row-n(0)["fixed"]
```

Investigate

- How would you get the `weeks` column out of the *second* row? The third?
- Complete the exercises on Lookup Questions (Page 20).

We can use the `row-n` method to define entire *animal rows* as values. Type the following lines of code into the Definitions Area and click "Run":

```
animalA = animals-table.row-n(1)
animalB = animals-table.row-n(16)
```

Flip back to page 2 of your workbook and look at The Animals Dataset. Which row is animalA? Label it in the margin next to the dataset. Which row is animalB? Label it in the margin next to the dataset.

Now turn back to your screen. What happens when you evaluate `animalA` in the Interactions Area?

- Define *at least* two additional values to be animals from the `animals-table`, called `animalC` and `animalD`.

Synthesize

Have students share their answers, and see if there are any common questions that arise.

Additional Exercises:

- More Practice with Lookups