Displaying Categorical Data

Students learn to apply functions to entire Tables, generating pie charts and bar charts. They then explore other plotting and display functions that are part of the Data Science library.

<table>
<tr><td><b>Prerequisites</b></td><td>Applying Functions</td></tr>
<tr><td><b>Relevant Standards</b>
OK
K12CS
CSTA</td><td><i>Select one or more standards from the menu on the left (⌘-click on Mac, Ctrl-click elsewhere).</i></td></tr>
<tr><td><b>Lesson Goals</b></td><td>Students will be able to:
• Read pie and bar charts
• Explain the difference between pie and bar charts
• Generate pie and bar charts (among others) from the Animals Dataset</td></tr>
<tr><td><b>Student-facing Lesson Goals</b></td><td>• Let’s use functions to create graphs from data.</td></tr>
<tr><td><b>Materials</b></td><td>• Lesson slides (Google Slides)
• Computer for each student (or pair), with access to the internet
• Student workbook, and something to write with</td></tr>
<tr><td><b>Preparation</b></td><td>• Make sure all materials have been gathered
• Decide how students will be grouped in pairs
• 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</td></tr>
<tr><td><b>Supplemental Resources</b></td><td></td></tr>
<tr><td><b>Language Table</b></td><td>
<table>
<tr><th>Types</th><th>Functions</th><th>Values</th></tr>
<tr><td>Number</td><td><code>num-sqrt, num-sqr</code></td><td><code>4, -1.2, 2/3</code></td></tr>
<tr><td>String</td><td><code>string-repeat, string-contains</code></td><td><code>"hello", "91"</code></td></tr>
<tr><td>Boolean</td><td><code>==, <, <=, >=, string-equal</code></td><td><code>true, false</code></td></tr>
<tr><td>Image</td><td><code>triangle, circle, star, rectangle, ellipse, square, text, overlay</code></td><td>◍◭◈</td></tr>
</table>
</td></tr>
</table>

Glossary

bar chart :: a display of categorical data that uses bars positioned over category values; each bar’s height reflects the count or percentage of data values in that category

contract :: a statement of the name, domain, and range of a function

domain :: the type or set of inputs that a function expects

pie chart :: a display that uses areas of a circular pie’s slices to show percentages in each category

Displaying Categorical Variables

10 minutes

Overview

Students extend their understanding of Contracts and function application, learning new functions that consume Tables and produce displays and plots.

Launch

Have students ever seen any *pictures* created from tables of data? Can they think of a situation when they'd want to consume a *Table*, and use that to produce an image? The library included at the top of the file includes some helper functions that are useful for Data Science, which we will use throughout this course. Here is the ***Contract*** for a function that makes ***pie charts***, and an example of using it:

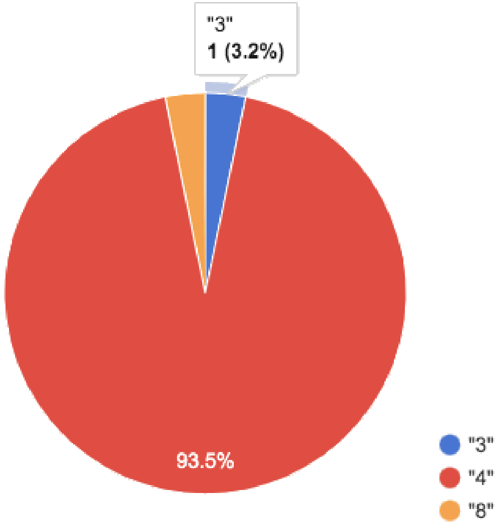
```
# pie-chart :: (t :: Table, col :: String) -> Image
pie-chart(animals-table, "legs")
```

- What is the Name of this function?
- How many inputs are in its ***Domain***?
- In the Interactions Area, type `pie-chart(animals-table, "legs")` and hit Enter. What happens?

Hovering over a pie slice reveals the label, as well as the count and the percentage of the whole. In this example we see that there is one three-legged animal, representing 3.2% of the population.

We can also resize the window by dragging its borders. This allows us to experiment with the data before closing the window and generating the final, non-interactive image.

The function `pie-chart` consumes a Table of data, along with the *name of a **categorical** column you want to display*. The computer goes through the column, counting the number of times that each value appears. Then it draws a pie slice for each value, with the size of the slice being the percentage of times it appears. In this example, we used our `animals-table` table as our dataset, and made a pie chart showing the distribution of `legs` across the shelter.



Investigate

Here is the ***Contract*** for another function, which makes ***bar charts***:

```
# bar-chart :: (t :: Table, col :: String) -> Image
```

- Which column of the animals table tells us how many legs an animal has?
- Use `bar-chart` to make a display showing how many animals have each number of legs.
- Experiment with pie and bar charts, passing in different column names. If you get an error message, *read it carefully!*
- What do you think are the rules for what kinds of columns can be used by *bar-chart* and *pie-chart*?
- When would you want to use one chart instead of another?

Possible Misconceptions

Pie charts and bar charts may show counts or percentages (in Pyret, pie charts show percentages and bar charts show counts). Bar charts look a lot like histograms, which are actually quite different because they display quantitative data, not categorical. Also, a pie chart can only display one categorical variable but a bar chart might be used to display two or more categorical variables.

Synthesize

Pie and Bar Charts display what portion of a sample that belongs to each category. If they are based on sample data from a larger population, we use them to *infer* the proportion of a whole population that might belong to each category.

larger population, we use them to *infer* the proportion of a whole population that might belong to each category.

> Pie charts and bar charts are mostly used to *display categorical columns* .

While bars in some bar charts should follow some logical order (alphabetical, small-medium-large, etc), the pie slices and bars can technically be placed in *any* order, without changing the meaning of the chart.

Exploring other Displays

30 minutes

Overview

Students freely explore the Data Science display library. In doing so, they experiment with new charts, practice reading *Contracts* and error messages, and develop better intuition for the programming constructs they've seen before.

Launch

There are *lots* of other functions, for all different kinds of charts and plots. Even if you don't know what these plots are for yet, see if you can use your knowledge of Contracts to figure out how to use them.

Investigate

Complete Exploring Displays (Page 14) and (More) Exploring Displays (Page 15).

Possible Misconceptions

There are *many* possible misconceptions about displays that students may encounter here. **But that's ok!** Understanding all those other plots is *not* a learning goal for this lesson. Rather, the goal is to have them develop some loose familiarity, and to get more practice reading Contracts.

Synthesize

Today you've added more functions to your toolbox. Functions like `pie-chart` and `bar-chart` can be used to visually display data, and even transform entire tables!
You will have many opportunities to use these concepts in this course, by writing programs to answer data science questions.

Extension Activity

Sometimes we want to summarize a categorical column in a Table, rather than a pie chart. For example, it might be handy to have a table that has a row for dogs, cats, lizards, and rabbits, and then the count of how many of each type there are. Pyret has a function that does exactly this! Try typing this code into the Interactions Area:

```
count(animals-table, "species")
```

What did we get back? `count` is a function that consumes a table and the name of a categorical column, and produces a *new table* with exactly the columns we want: the name of the category and the number of times that category occurs in the dataset. What are the names of the columns in this new table?

- Use the `count` function to make a table showing the number of animals that are `fixed` (or not) from the shelter.
- Use the `count` function to make a table showing the number of animals of each `sex` from the shelter.

Sometimes the dataset we have is *already* summarized in a table like this, and we want to make a chart from *that*. In this situation, we want to base our display on the summary table: the size of the pie slice or bar is taken directly from the count column, and the label is taken directly from the value column. When we want to use summarized data to produce a pie chart, we have another function:

```
# pie-chart-summarized :: (t :: Table, label :: String, data ::
```

```
String) -> Image
pie-chart-summarized(count(animals-table,"species"), "value",
"count")
```

Additional Exercises:

Practice Plotting