Lecture 7 – Functions
Announcements

- **HW02** - *Table Manipulation & Visualization:*
  - Due Monday (02/14)

- **Lab 03** - *Functions and Visualizations*
  - Due Monday (02/14)

- **HW03** - *Functions, Histograms, and Groups*
  - Due Monday (02/21)

- **Checkpoint/Project 1:**
  - Paired assignment that covers the previous section of the course material
  - Released tonight or tomorrow and due 2 weeks (Friday 02/25)
Anatomy of a Function

- Name
- Parameters / Argument Names
- Body
- Return Expression
def sread(values):
    spread_val = max(values) - min(values)
    return spread_val
def sread(values):
    spread_val = max(values) - min(values)
    return spread_val
def spread(values):
    spread_val = max(values) - min(values)
    return spread_val
Example Function - Body

def spread(values):
    spread_val = max(values) - min(values)
    return spread_val
def sread(values):
    spread_val = max(values) - min(values)
    return spread_val
def f(s):
    return np.round(s / sum(s) * 100, 2)

- What kind of input does it take?
- What output will it give?
- What's a reasonable name?
Applying Functions to Columns

The `apply` method creates an array by calling a function on every element in input column(s)

- First argument: Function to apply
- Other arguments: The input column(s)

```
table_name.apply(function_name, 'column_label')
```
The `group` method aggregates all rows with the same value for a column into a single row in the resulting table.

- First argument: Which column to group by
- Second argument: (Optional) How to combine values
  - `len` — number of grouped values (default)
  - `list` — list of all grouped values
  - `sum` — total of all grouped values
A list is a sequence of values (just like an array), but the values can all have different types

\[[2+3, 'four', Table().with_column('K', [3, 4])]\]

- Lists can be used to create table rows.
- If you create a table column from a list, it will be converted to an array automatically
The `group` method can also aggregate all rows that share the combination of values in multiple columns.

- **First argument**: A list of which columns to group by.
- **Second argument**: (Optional) How to combine values.
Pivot Tables

- Cross-classifies according to two categorical variables
- Produces a grid of counts or aggregated values
- Two required arguments:
  - First: variable that forms column labels of grid
  - Second: variable that forms row labels of grid
- Two optional arguments (include both or neither)
  - values='column_label_to_aggregate'
  - collect=function_to_aggregate_with
<table>
<thead>
<tr>
<th>Pivot</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>One combo of grouping</td>
<td>One combo of grouping</td>
</tr>
<tr>
<td>variables per entry</td>
<td>variables per row</td>
</tr>
<tr>
<td>Two grouping variables:</td>
<td>Any number of grouping</td>
</tr>
<tr>
<td>columns and rows</td>
<td>variables</td>
</tr>
<tr>
<td>Aggregate values of</td>
<td>Aggregate values of</td>
</tr>
<tr>
<td>values column</td>
<td>all other columns in table</td>
</tr>
<tr>
<td>Missing combos = 0</td>
<td>Missing combos absent</td>
</tr>
<tr>
<td>(or empty string)</td>
<td></td>
</tr>
</tbody>
</table>
Joining Two Tables

tblA.join(colA, tblB, colB)

tblA.join(colA, tblB)
Tomorrow’s Reading

- Chapter 9.1 – 9.3
- Conditionals & Randomness