Lecture 5

Building Tables
Announcements
Review: Arrays

An array contains a sequence of values

- All elements of an array should have the same type
- Arithmetic is applied to each element individually
- When two arrays are added, they must have the same size; corresponding elements are added in the result
- A column of a table is an array
Ranges
Ranges

A range is an array of consecutive numbers

- `np.arange(end)`: An array of increasing integers from 0 up to `end`
- `np.arange(start, end)`: An array of increasing integers from `start` up to `end`
- `np.arange(start, end, step)`: A range with `step` between consecutive values

The range always includes `start` but excludes `end`
Tables
Ways to create a table

- `Table.read_table(filename)` - reads a table from a spreadsheet
- `Table()` - an empty table
- and... `select, where, sort` and so on all create new tables
Example
Charles Joseph Minard, 1781-1870

- French civil engineer who created one of the greatest graphs of all time
- Visualized Napoleon's 1812 invasion of Russia, including
  - the number of soldiers
  - the direction of the march
  - the latitude and longitude of each city
  - the temperature on the return journey
  - Dates in November and December
Some of Minard’s Data

<table>
<thead>
<tr>
<th>Longitude</th>
<th>Latitude</th>
<th>City</th>
<th>Direction</th>
<th>Survivors</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>54.8</td>
<td>Smolensk</td>
<td>Advance</td>
<td>145000</td>
</tr>
<tr>
<td>33.2</td>
<td>54.9</td>
<td>Dorogobouge</td>
<td>Advance</td>
<td>140000</td>
</tr>
<tr>
<td>34.4</td>
<td>55.5</td>
<td>Chjat</td>
<td>Advance</td>
<td>127100</td>
</tr>
<tr>
<td>37.6</td>
<td>55.8</td>
<td>Moscou</td>
<td>Advance</td>
<td>100000</td>
</tr>
<tr>
<td>34.3</td>
<td>55.2</td>
<td>Wixma</td>
<td>Retreat</td>
<td>55000</td>
</tr>
<tr>
<td>32</td>
<td>54.6</td>
<td>Smolensk</td>
<td>Retreat</td>
<td>24000</td>
</tr>
<tr>
<td>30.4</td>
<td>54.4</td>
<td>Orscha</td>
<td>Retreat</td>
<td>20000</td>
</tr>
<tr>
<td>26.8</td>
<td>54.3</td>
<td>Moiodexno</td>
<td>Retreat</td>
<td>12000</td>
</tr>
</tbody>
</table>

(Demo)
Table Methods

- Creating and extending tables:
  - `Table().with_column` and `Table.read_table`
- Finding the size: `num_rows` and `num_columns`
- Referring to columns: labels, relabeling, and indices
  - `labels` and `relabelled`; column indices start at 0
- Accessing data in a column
  - `column` takes a label or index and returns an array
- Using array methods to work with data in columns
  - `item, sum, min, max`, and so on
- Creating new tables containing some of the original columns:
  - `select, drop`
Manipulating Rows

- `t.sort(column)` sorts the rows in increasing order
- `t.take(row_numbers)` keeps the numbered rows
  - Each row has an index, starting at 0
- `t.where(column, are.condition)` keeps all rows for which a column's value satisfies a condition
- `t.where(column, value)` keeps all rows containing a certain value in a column