Lecture 8
Groups, Joins, and Maps
Announcements

- Project 1 out this Saturday AM.
  - Can work on it with a partner from your same lab section (or by yourself if you prefer).
  - Note: Only work on one copy of the notebook at a time!
- Prelim 1 is Thursday, Feb. 27. More info early next week.
What we’ll do: Citibike visualization

Learn enough computing to do our own visualizations and observations to identify patterns in big data sets.
Grouping Rows
The **group** method aggregates all rows with the same value for a column into a single row in the result

- First argument: Which column to group by
- Second argument: (Optional) How to combine values
  - `len` — number of grouped values (default)
  - `sum` — total of all grouped values
  - `list` — list of all grouped values

(Demo)
Grouping By Two Columns

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

(Demo)
Challenge Question

Which NBA teams spent the most on their starters in 2016?

- Each team has one *starter* per position
- Assume the starter for a team & position is the player with the highest salary on that team in that position

<table>
<thead>
<tr>
<th>PLAYER</th>
<th>POSITION</th>
<th>TEAM</th>
<th>SALARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paul Millsap</td>
<td>PF</td>
<td>Atlanta Hawks</td>
<td>18.6717</td>
</tr>
<tr>
<td>Al Horford</td>
<td>C</td>
<td>Atlanta Hawks</td>
<td>12</td>
</tr>
<tr>
<td>Tiago Splitter</td>
<td>C</td>
<td>Atlanta Hawks</td>
<td>9.75625</td>
</tr>
</tbody>
</table>
Pivot Tables
Pivot

- 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_with_which_to_aggregate`

(Demo)
## Take-Home Question

Generate a table of the names of the starters for each team

<table>
<thead>
<tr>
<th>TEAM</th>
<th>C</th>
<th>PF</th>
<th>PG</th>
<th>SF</th>
<th>SG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Atlanta Hawks</td>
<td>Al Horford</td>
<td>Paul Millsap</td>
<td>Jeff Teague</td>
<td>Thabo Sefolosha</td>
<td>Kyle Korver</td>
</tr>
<tr>
<td>Boston Celtics</td>
<td>Tyler Zeller</td>
<td>Jonas Jerebko</td>
<td>Avery Bradley</td>
<td>Jae Crowder</td>
<td>Evan Turner</td>
</tr>
<tr>
<td>Brooklyn Nets</td>
<td>Andrea Bargnani</td>
<td>Thaddeus Young</td>
<td>Jarrett Jack</td>
<td>Joe Johnson</td>
<td>Bojan Bogdanovic</td>
</tr>
<tr>
<td>Charlotte Hornets</td>
<td>Al Jefferson</td>
<td>Marvin Williams</td>
<td>Kemba Walker</td>
<td>Michael Kidd-Gilchrist</td>
<td>Nicolas Batum</td>
</tr>
<tr>
<td>Chicago Bulls</td>
<td>Joakim Noah</td>
<td>Nikola Mirotic</td>
<td>Derrick Rose</td>
<td>Doug McDermott</td>
<td>Jimmy Butler</td>
</tr>
<tr>
<td>Cleveland Cavaliers</td>
<td>Tristan Thompson</td>
<td>Kevin Love</td>
<td>Kyrie Irving</td>
<td>LeBron James</td>
<td>Iman Shumpert</td>
</tr>
<tr>
<td>Dallas Mavericks</td>
<td>Zaza Pachulia</td>
<td>David Lee</td>
<td>Deron Williams</td>
<td>Chandler Parsons</td>
<td>Justin Anderson</td>
</tr>
<tr>
<td>Denver Nuggets</td>
<td>JJ Hickson</td>
<td>Kenneth Faried</td>
<td>Jameer Nelson</td>
<td>Danilo Gallinari</td>
<td>Gary Harris</td>
</tr>
<tr>
<td>Detroit Pistons</td>
<td>Aron Baynes</td>
<td>Reggie Jackson</td>
<td>Stanley Johnson</td>
<td>Jodie Meeks</td>
<td></td>
</tr>
<tr>
<td>Golden State Warriors</td>
<td>Andrew Bogut</td>
<td>Draymond Green</td>
<td>Stephen Curry</td>
<td>Andre Iguodala</td>
<td>Klay Thompson</td>
</tr>
</tbody>
</table>
Joins
## Joining Two Tables

Keep all rows in the table that have a match...

```python
drinks.join('Cafe', discounts, 'Location')
```

... for the value in this column...

... somewhere in this other table's...

... column that contains matching values.

<table>
<thead>
<tr>
<th>drinks</th>
<th>discounts</th>
<th>Cafe</th>
<th>Drink</th>
<th>Price</th>
<th>Coupon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Milk tea</td>
<td></td>
<td>Panda Tea</td>
<td>Espresso</td>
<td>4</td>
<td>25%</td>
</tr>
<tr>
<td>Espresso</td>
<td></td>
<td>Gimme</td>
<td>Espresso</td>
<td>2</td>
<td>50%</td>
</tr>
<tr>
<td>Latte</td>
<td></td>
<td>Gimme</td>
<td>Latte</td>
<td>3</td>
<td>5%</td>
</tr>
<tr>
<td>Espresso</td>
<td></td>
<td>Cafe Gola</td>
<td>Milk Tea</td>
<td>2</td>
<td>25%</td>
</tr>
</tbody>
</table>

The joined column is sorted automatically.

(Demo)
Bikes
Maps
Maps

A table containing columns of latitude and longitude values can be used to generate a map of markers

.map_table(table, ...)