YaleNUSCollege

YSC2239 Lecture 5

Recap

- Understanding Data
- Useful **table** functions: **select**, **sort**, **where**, **drop**, **show**, **take**, **with_column**, **plot**, **group, bar/barh**

https://github.com/data-8/datascience/blob/8d3fb1e0791b9e072c741b6d9c8efc98d554159e/datascience/tables.py

Today’s class

- Binning and Histogram
- Functions
- Apply
- Group
- Join

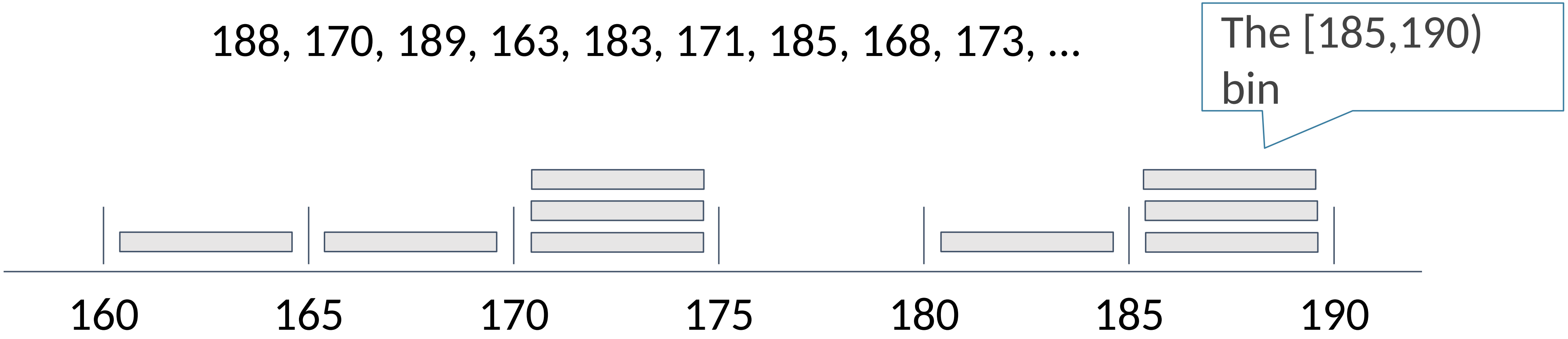
- Reading: Chapter 8

Binning Numerical Values

Binning is counting the number of numerical values that lie within ranges, called bins.

- Bins are defined by their lower bounds (inclusive)
- The upper bound is the lower bound of the next bin

188, 170, 189, 163, 183, 171, 185, 168, 173, ...



Histogram

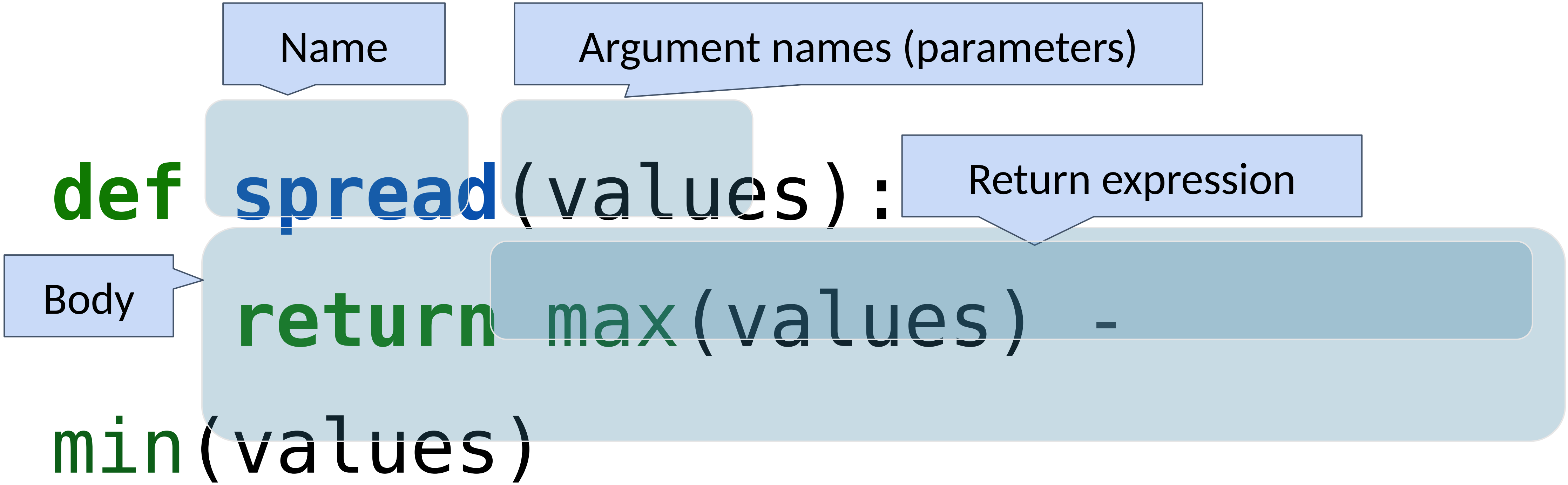
- Chart that displays the distribution of a numerical variable
- Uses bins; there is one bar corresponding to each bin
- Uses the area principle:
    - The ***area*** of each bar is the percent of individuals in the corresponding bin

(Demo)

Defining Functions

Def Statements

User-defined functions give names to blocks of code



Name

Argument names (parameters)

Return expression

Body

```
def spread(values):
    return max(values) -
min(values)
```

(Demo)

Apply with Multiple Columns

Apply

The `apply` method creates an array by calling a function on every element in one or more input columns

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

```
table_name.apply(one_arg_function, 'column_label')

table_name.apply(two_arg_function,
                 'column_label_for_first_arg',
                 'column_label_for_second_arg')
```

`apply` called with only a function applies it to each row

(Demo)

Grouping by One Attribute

Grouping by One Column

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

(Demo)

Lists

Lists are Generic Sequences

A list is a sequence of values (just like an array),
but the values can 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
- Built into python (you don’t need numpy.)

(Demo)

Cross-Classification

Grouping By Multiple 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)

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_to_aggregate_with'

(Demo)

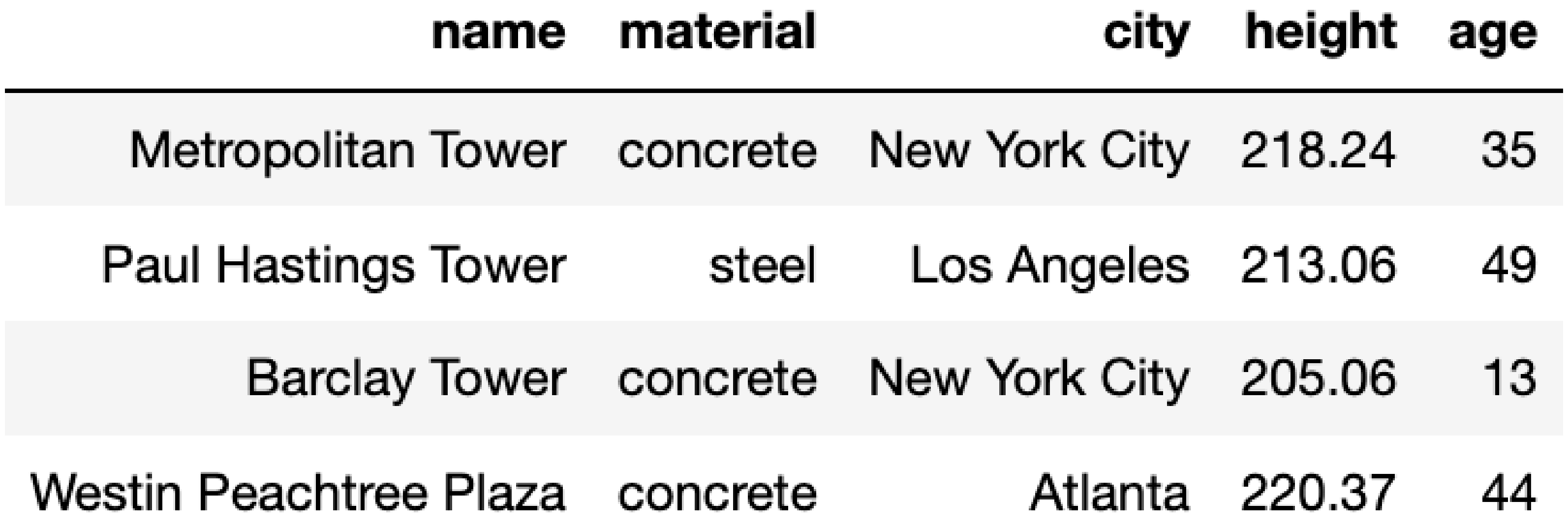
Group or Pivot?

- Distribution of one categorical variable → .group()
- Cross-classification of two or more categorical variables:
  - One row per combination → .group()
  - One variable vertically, one horizontally → .pivot()

Challenge Question

1. For each city, what’s the height of the tallest building for each material?
2. For each city, what’s the height difference between the tallest steel building and the tallest concrete building?

`sky`



name	material	city	height	age
Metropolitan Tower	concrete	New York City	218.24	35
Paul Hastings Tower	steel	Los Angeles	213.06	49
Barclay Tower	concrete	New York City	205.06	13
Westin Peachtree Plaza	concrete	Atlanta	220.37	44

(Demo)

Joins

Joining Two Tables

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

Match rows in this table ...

… using values in this column ...

… with rows in that table ...

… using values in that column.

Columns from both tables

`drinks`

Drink	Cafe	Price
Milk Tea	Asha	5.5
Espresso	Strada	1.75
Latte	Strada	3.25
Espresso	FSM	2

`discounts`

Coupon	Location
10%	Asha
25%	Strada
5%	Asha

The joined column is sorted automatically

Cafe	Drink	Price	Coupon
Asha	Milk Tea	5.5	10%
Asha	Milk Tea	5.5	5%
Strada	Espresso	1.75	25%
Strada	Latte	3.25	25%

Important Table Methods

```
t.select(column, …) or t.drop(column, …)
t.take([row, …]) or t.exclude([row, …])

t.sort(column, descending=False, distinct=False)
t.where(column, are.condition(...))

t.apply(function, column, …)
t.group(column) or t.group(column, function)
t.group([column, …]) or t.group([column, …], function)
t.pivot(cols, rows) or t.pivot(cols, rows, vals, function)

t.join(column, other_table, other_table_column)
```

http://data8.org/datascience/tables.html

Reminders

- Assignment 2