Lecture 4

Functions, Group, Join

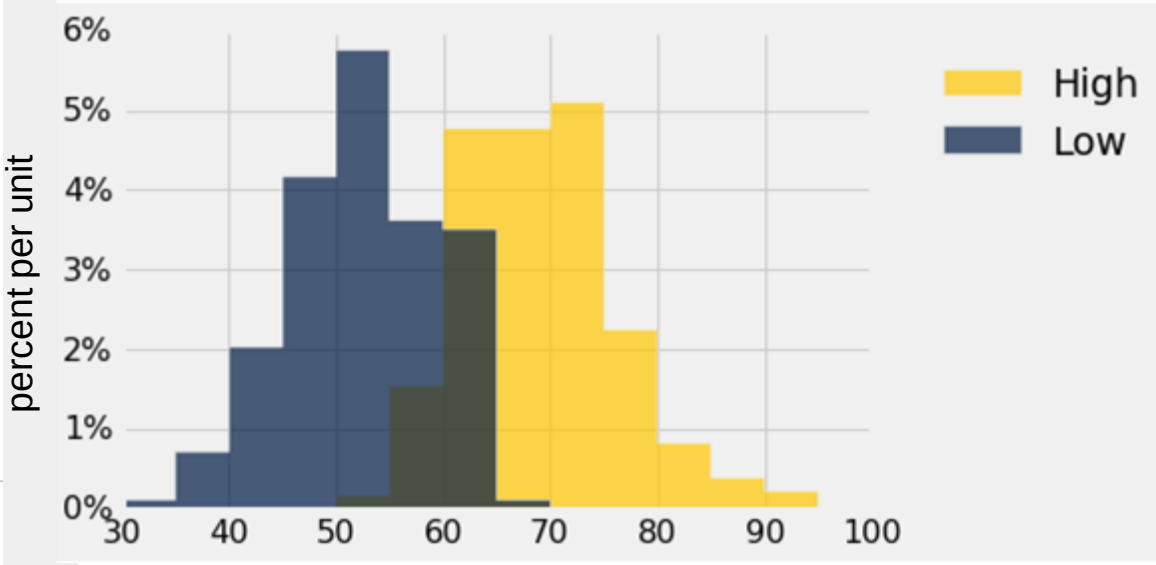
Slides apodpted from UC Bekeley, DATA 8 course

Discussion Question

This histogram describes a **year** of daily temperatures

Try to answer these questions:

- What proportion of days had a high temp in the range 60-69?
- What proportion had a low of 45 or more?
- How many days had a difference of more than 20 degrees between their high & low temperatures?



Reminder: Height Measures Density

$$\text{Height} = \frac{\%\text{ in bin}}{\text{width of bin}}$$

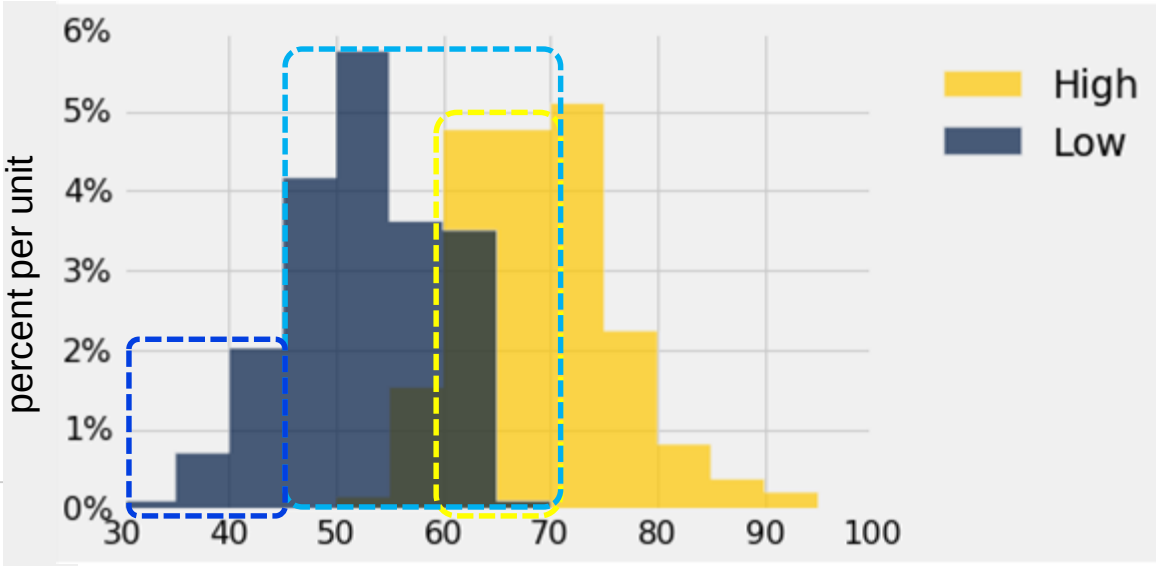
- The height measures the percent of data in the bin ***relative to the amount of space in the bin*****.**
- So height measures crowdedness, or **density**.

Discussion Question

This histogram describes a **year** of daily temperatures

Try to answer these questions:

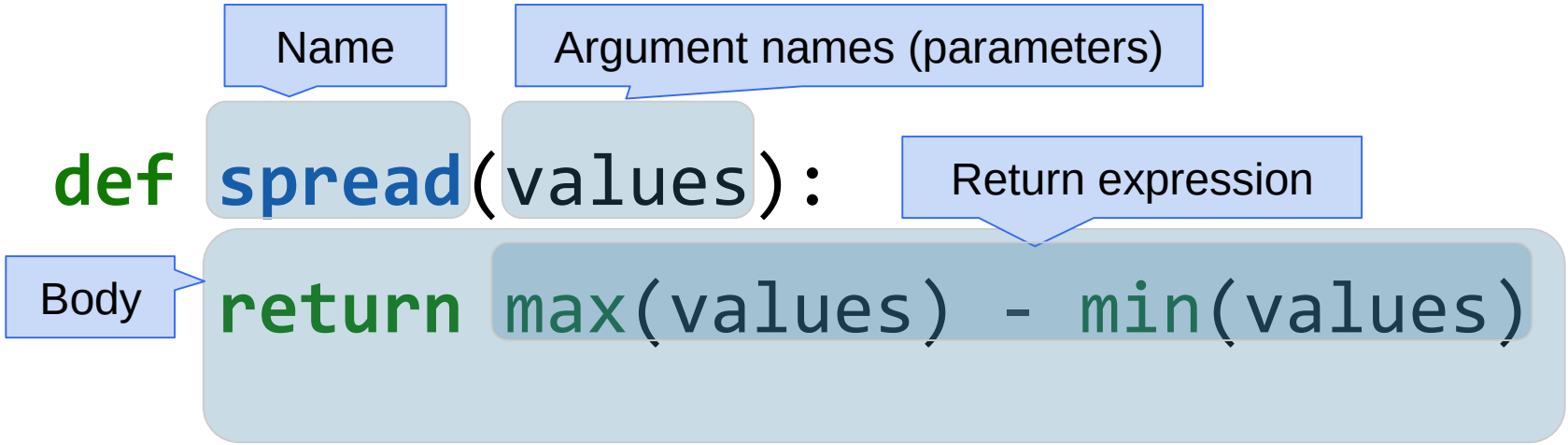
- What proportion of days had a high temp in the range 60-69?
- What proportion had a low of 45 or more?
- How many days had a difference of more than 20 degrees between their high & low temperatures?



Defining Functions

Def Statements

User-defined functions give names to blocks of code



(Demo)

Discussion Question

What does this function do? What kind of input does it take? What output will it give? What's a reasonable name?

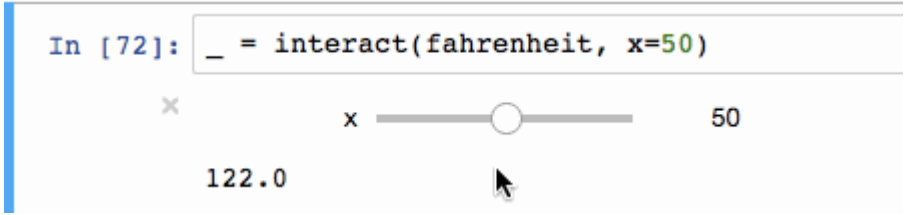
```
def f(s):
    return np.round(s / sum(s) * 100, 2)
```

(Demo)

Brief Aside: `interact`

The `interact` function calls your functions interactively:

```
_ = interact(fahrenheit, x=50)
```



Completely optional to know, but may be handy.

(Demo)

Apply

Apply

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')
```

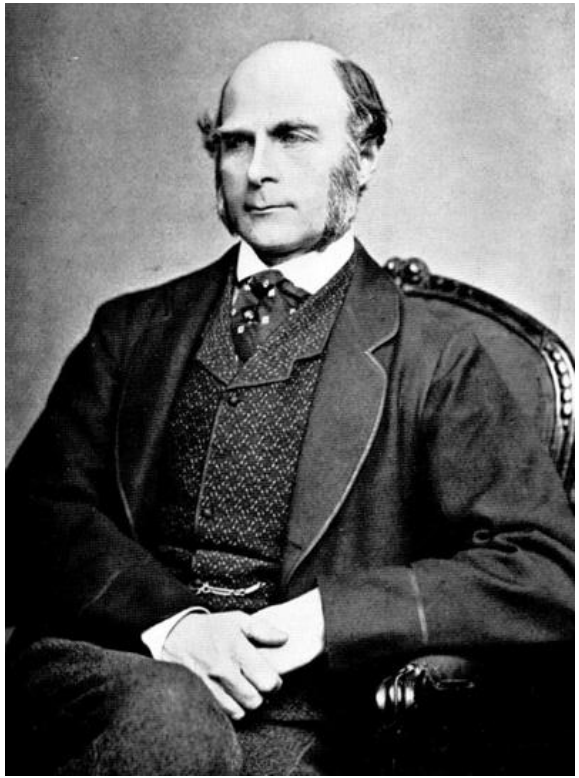
(Demo)

Example: Prediction

Sir Francis Galton

- 1822 - 1911 (knighted in 1909)
- A pioneer in making predictions
- Particular interest in heredity
- Charles Darwin's half-cousin

(Demo)



Apply with Multiple Arguments

Apply

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

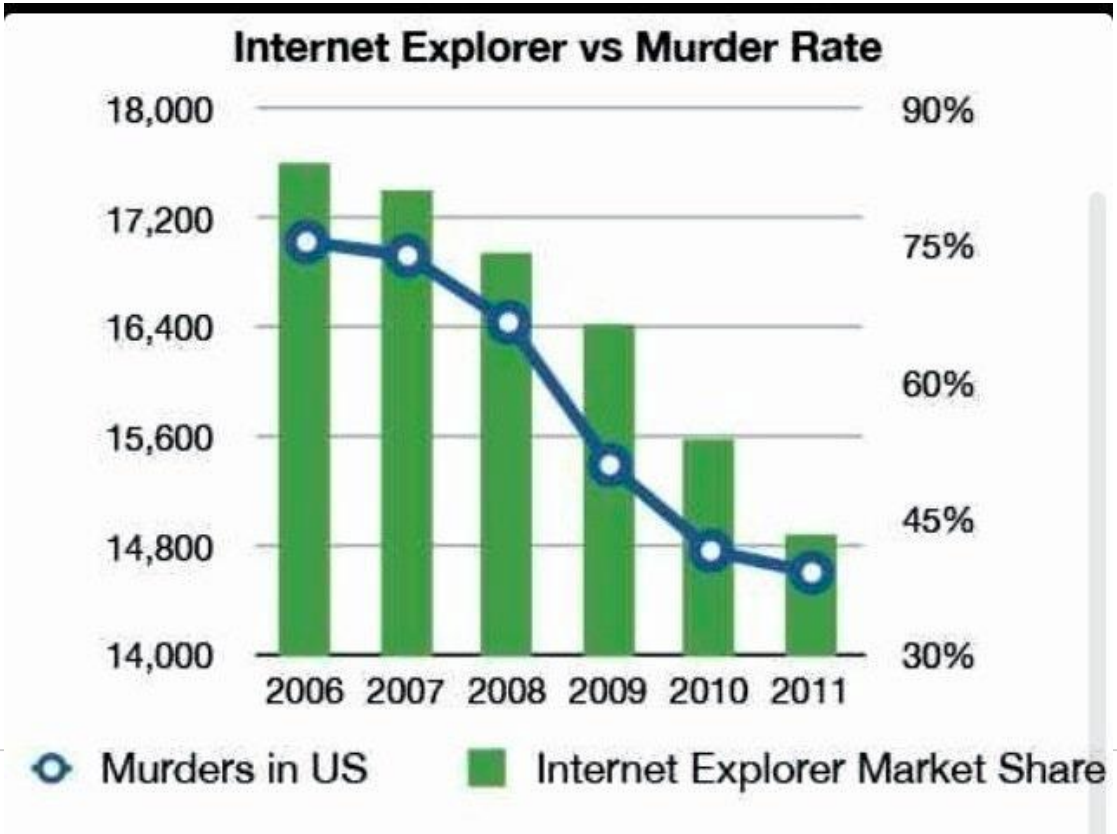
- First argument: Function to apply
- Other arguments: The input column(s)

```
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)

Data of the Day



Grouping Rows

Group

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 spend the most on their starters?

- 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

PLAYER	POSITION	TEAM	SALARY
Paul Millsap	PF	Atlanta Hawks	18.6717
Al Horford	C	Atlanta Hawks	12
Tiago Splitter	C	Atlanta Hawks	9.75625

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

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

Joins

Joining Two Tables

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

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

… for the value in this column ...

… somewhere in this other table's ...

… column that contains matching values.

drinks

Drink	Cafe	Price
Milk Tea	Tea One	4
Espresso	Nefeli	2
Latte	Nefeli	3
Espresso	Abe's	2

discounts

Coupon	Location
25%	Tea One
50%	Nefeli
5%	Tea One

Only the first match in the joined table is ever used.

Cafe	Drink	Price	Coupon
Nefeli	Espresso	2	50%
Nefeli	Latte	3	50%
Tea One	Milk Tea	4	25%

The joined column is sorted automatically

(Demo)

Maps

Maps

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

```
_____.map_table(table, ...)
```

Either `Marker` or `Circle`

Column 0: latitudes
Column 1: longitudes
Column 2: labels
Column 3: colors
Column 4: sizes

Applies to all features:
`color='blue'`
`size=200`