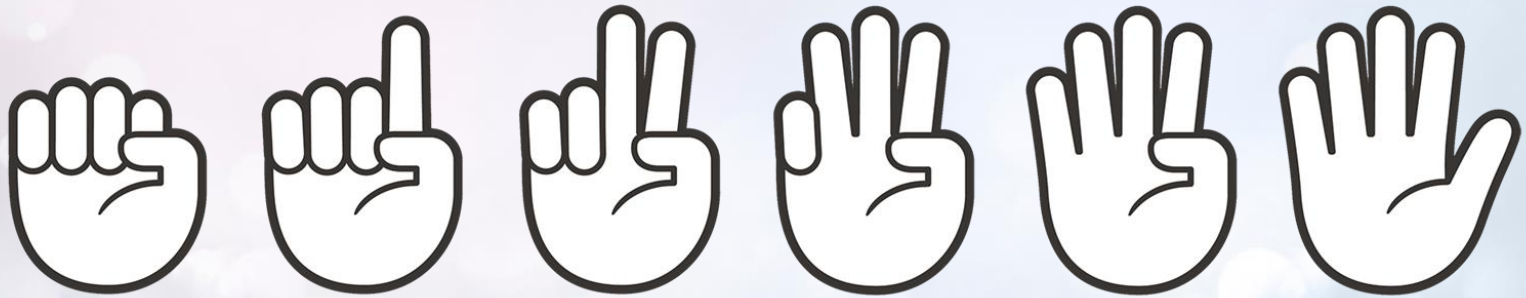
Plotting with Pandas

Data Boot Camp

Lesson 5.2



FIST TO FIVE:

How comfortable does everyone feel after last class?

Don’t worry; today is all about reinforcing concepts!

Warm-Up Activity: PyPlot

In this activity, you will use PyPlot to create the most effective visualization for a variety of datasets.

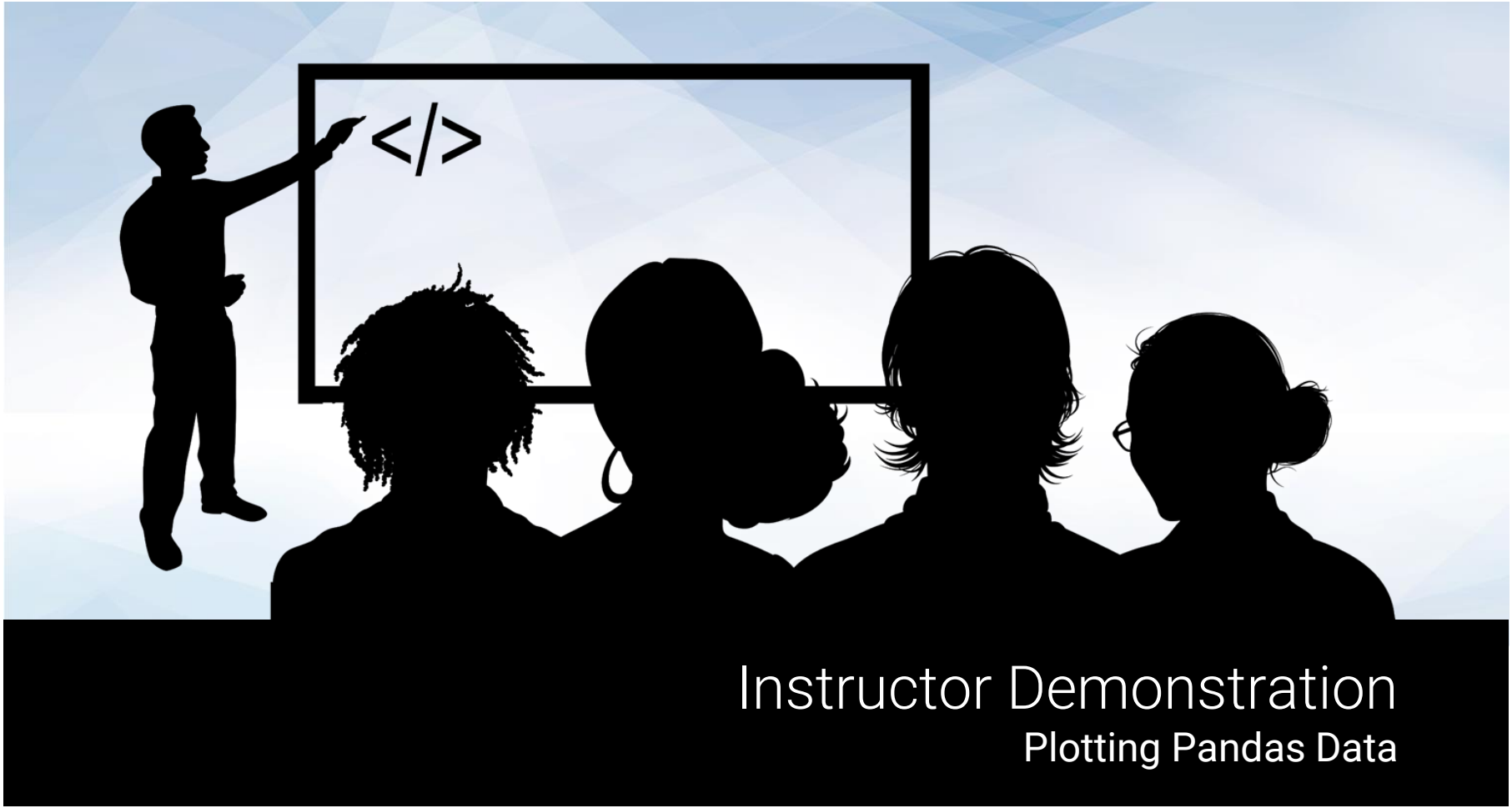
Suggested Time:
Sufficient Time

Warm-Up Activity: PyPlot

- Open the unsolved activity file and look at the starter code for each dataset.
  File: `Unsolved/plot_drills.ipynb`
- Determine what chart or plot best fits with the starter code of each dataset.
- Complete the code block to create a plot for each dataset.
- Be sure to provide each plot with a title and labels.

Hint: If you are unsure what type of data is contained in the starter code, print it out!

Time’s Up! Let’s Review.



Instructor Demonstration

Plotting Pandas Data

Imaginary datasets are good to practice with...

...But we will deal with real-world data more often.

- Strange formats
- Messy
- Missing data
- Misleading headers

How to Work with Messy Data

Pandas enables us to quickly and easily:

- Rename headers
- Remove missing data
- Convert and clean up column data

In most cases, we will work with real-world data in Pandas.

The Creators of Pandas Are Geniuses!

Most people who read in datasets into Pandas will:

- Clean and preprocess their data
- Visualize their findings using Matplotlib

Pandas creators added Matplotlib functionality directly into Pandas, which:

- Speeds up the process of creating lists and aesthetics
- Still allows for Pyplot customizability

Best of both worlds

<Time to Code>

Speaking of real-world datasets…

Activity: Battling Kings

In this activity, you will create a bar chart that visualizes which kings within the Game of Thrones universe have participated in the most battles.

Suggested Time:

Winter is Coming Minutes

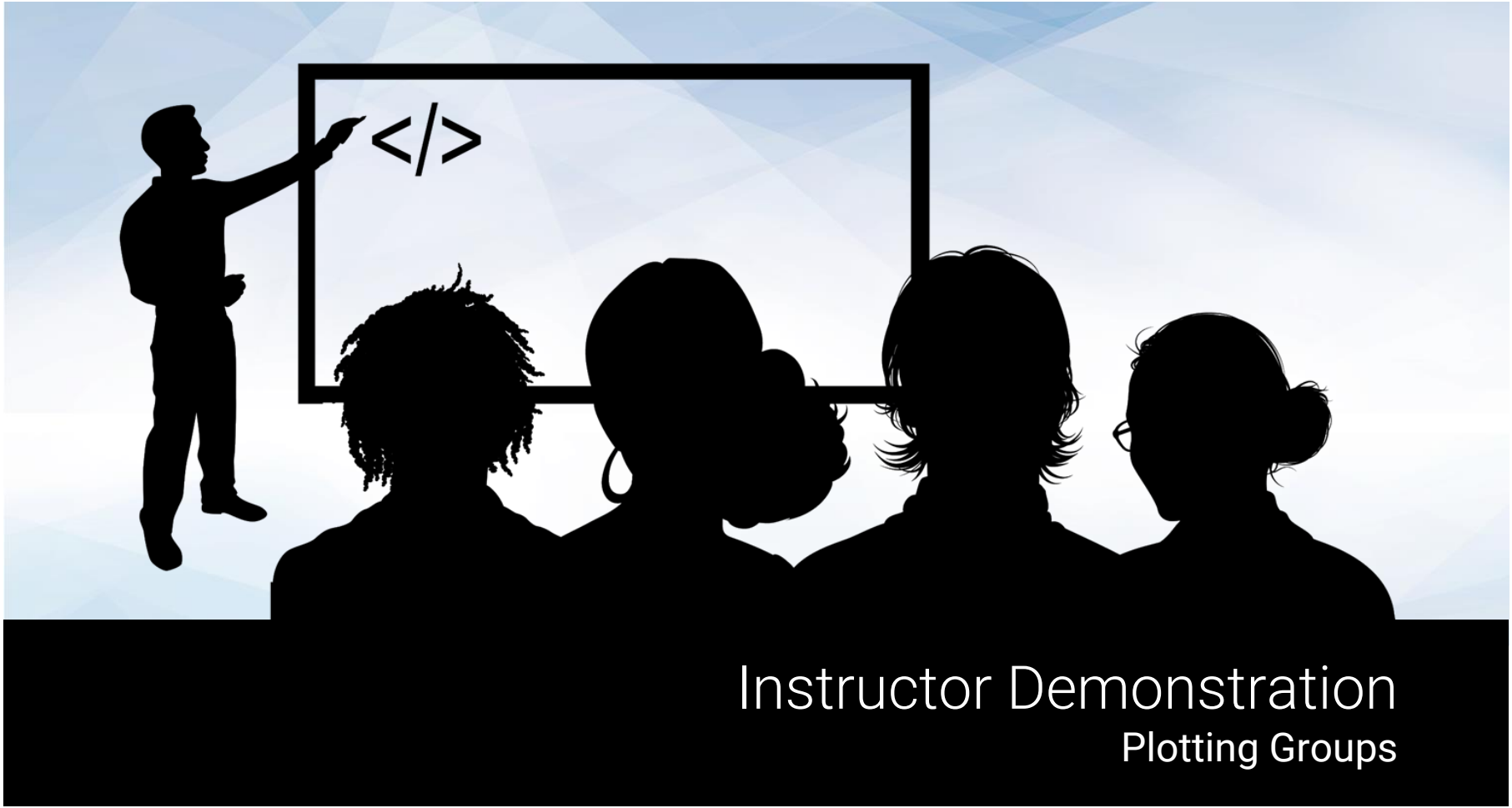
Activity: Battling Kings

- Use Pandas to load the `got.csv` dataset.
- Create a series containing the number of times each king was an attacker.
- Create a series containing the number of times each king was a defender.
- Combine these two variables into a single series.

Hint: How should you combine these two series? Can you add series in Pandas?

- Use your combined data to retrieve labels for your *x*-ticks.
- Create a red bar chart, and set its *x*-tick labels appropriately.
- Add a title and labels to your plot.
- Display your plot. Who participated in the most battles? The least?

Time’s Up! Let’s Review.



Instructor Demonstration

Plotting Groups

How do we group data in Pandas?

Grouping and Summarizing in Pandas

The `dataframe.groupby()` function allows us to group data.

- Data can be grouped by function or category.
- The output of the function is a GroupBy object.

```
<pandas.core.groupby.groupby.DataFrameGroupBy object at 0x10cde6278>
```

	datetime	city	country	shape	duration (seconds)	duration (hours/min)	comments
state							
ak	311	311	311	311	311	311	311
al	629	629	629	629	629	629	629
ar	578	578	578	578	578	578	578
az	2362	2362	2362	2362	2362	2362	2362
ca	8683	8683	8683	8683	8683	8683	8683
co	1385	1385	1385	1385	1385	1385	1385
ct	865	865	865	865	865	865	865
dc	7	7	7	7	7	7	7
de	165	165	165	165	165	165	165
fl	3754	3754	3754	3754	3754	3754	3754

We can plot directly from a **Pandas DataFrame...**

... *or* we can plot directly from a **Pandas GroupBy object!**

<Time to Code>

Activity: Bike Trippin’

In this activity, you will create a pair of charts based on community bike data* collected from Seattle.

*actual real-world data

Suggested Time:
Part of an hour

Activity: Bike Trippin'

Open the starter file provided and follow the prompts to import, split, and summarize the community bike dataset. **File:** `Unsolved/bike_trippin.ipynb`

Create a bar chart using Pandas and Matplotlib that visualizes how many individual bike trips were taken by each gender.

Create a pie graph using Pandas and Matplotlib that can be used to visualize the trip duration of a single bike, split up by gender.

Hint: There is a buggy value stored in the "gender" column of the original DataFrame. In order to create an accurate chart, this value will need to be found and removed.

Time’s Up! Let’s Review.


Break

Activity: Miles Per Gallon

In this activity, you will create a scatter plot using vehicle data, Pandas, and Matplotlib.

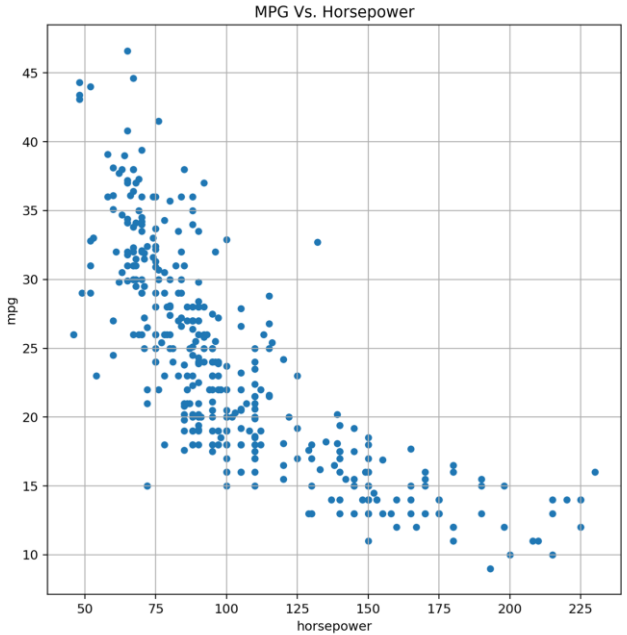
Suggested Time:

A different part of an hour

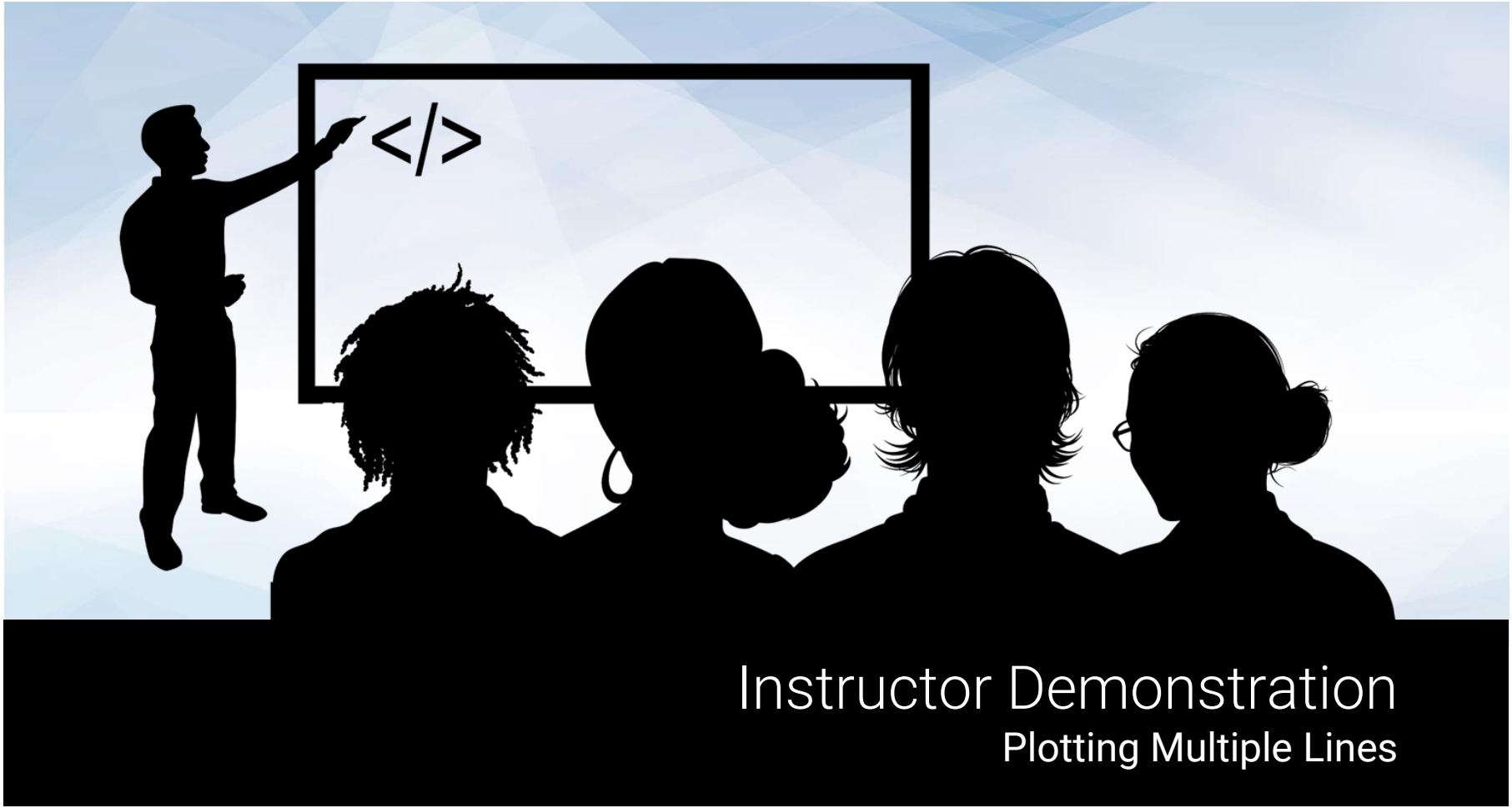
Activity: Miles Per Gallon

- Create a scatter plot using the data provided, Pandas, and Matplotlib, which compares the MPG of a vehicle with its horsepower.

- **File:** `Resources/mpg.csv`



Time’s Up! Let’s Review.



Instructor Demonstration

Plotting Multiple Lines

Adding Multiple Plots Using Pandas Is Easy!

We have learned how to use Pandas to:

- Generate line, bar, pie, and scatter plots.
- Plot summary figures using GroupBy objects.

But what if we want to combine multiple plots in the same figure?

Adding multiple plots to Pandas is the same as in Pyplot:

- Add multiple plots in the same code block.
- Show the combined figure using `pyplot.show()`.

<Time to Code>

Group Mini-Project: Winner Wrestling

The remainder of class will consist of a group mini-project.

The mini-project will be broken up into three parts.

Every few minutes we will review the previous part.

Suggested Time:
The remainder of class!

Winner Wrestling: Part 1 Instructions

Follow the instructions for the activity in the unsolved Jupyter Notebook file.

File: `Unsolved/winning_wrestlers.ipynb`

Your output should look like the following table:

	Wrestler	2013 Wins	2013 Losses	2013 Draws	2014 Wins	2014 Losses	2014 Draws	2015 Wins	2015 Losses	2015 Draws	2016 Wins	2016 Losses	2016 Draws
0	Daniel Bryan	177.0	37.0	6.0	35.0	16.0	2.0	51.0	7.0	0.0	NaN	NaN	NaN
1	Dean Ambrose	70.0	134.0	4.0	129.0	36.0	2.0	150.0	63.0	5.0	133.0	67.0	4.0
2	Antonio Cesaro	80.0	126.0	1.0	5.0	24.0	0.0	NaN	NaN	NaN	NaN	NaN	NaN
3	Seth Rollins	50.0	150.0	4.0	87.0	105.0	4.0	51.0	124.0	1.0	39.0	75.0	4.0
4	Randy Orton	129.0	63.0	8.0	33.0	87.0	5.0	81.0	10.0	1.0	39.0	21.0	0.0
5	Roman Reigns	49.0	140.0	5.0	118.0	28.0	4.0	187.0	19.0	7.0	142.0	12.0	5.0
6	Ryback	103.0	88.0	3.0	43.0	114.0	1.0	138.0	34.0	2.0	37.0	17.0	1.0
7	Damien Sandow	39.0	147.0	3.0	3.0	113.0	0.0	50.0	9.0	2.0	11.0	18.0	0.0
8	Alberto Del Rio	126.0	53.0	4.0	31.0	82.0	1.0	26.0	13.0	2.0	24.0	65.0	2.0
9	Dolph Ziggler	62.0	117.0	1.0	134.0	60.0	2.0	115.0	52.0	2.0	114.0	56.0	2.0

Time’s Up! Let’s Review.

Winner Wrestling: Part 2 Instructions

Follow the instructions for the activity in the unsolved Jupyter Notebook file.

File: `Unsolved/winning_wrestlers.ipynb`

Your output should look like the following table:

Wrestler	2013 Wins	2013 Losses	2013 Draws	2014 Wins	2014 Losses	2014 Draws	2015 Wins	2015 Losses	2015 Draws	2016 Wins	2016 Losses	2016 Draws	Total Wins	Total Losses	Total Draws	Total Matches
Dean Ambrose	70.0	134.0	4.0	129.0	36.0	2.0	150.0	63.0	5.0	133.0	67.0	4.0	482.0	300.0	15.0	797.0
Seth Rollins	50.0	150.0	4.0	87.0	105.0	4.0	51.0	124.0	1.0	39.0	75.0	4.0	227.0	454.0	13.0	694.0
Randy Orton	129.0	63.0	8.0	33.0	87.0	5.0	81.0	10.0	1.0	39.0	21.0	0.0	282.0	181.0	14.0	477.0
Roman Reigns	49.0	140.0	5.0	118.0	28.0	4.0	187.0	19.0	7.0	142.0	12.0	5.0	496.0	199.0	21.0	716.0
Ryback	103.0	88.0	3.0	43.0	114.0	1.0	138.0	34.0	2.0	37.0	17.0	1.0	321.0	253.0	7.0	581.0

Time’s Up! Let’s Review.

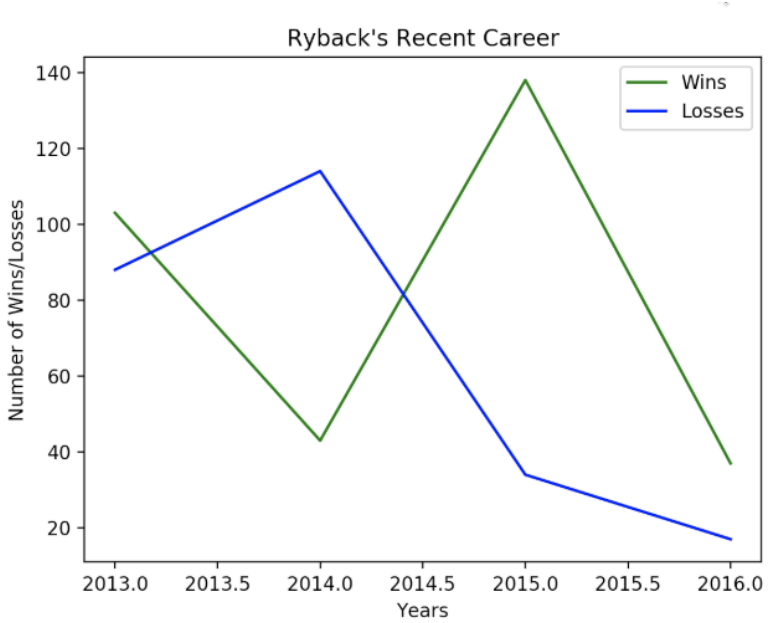
Winner Wrestling: Part 3 Instructions

Follow the instructions for the activity in the unsolved Jupyter Notebook file.

File:

`Unsolved/winning_wrestlers.ipynb`

Your output should look similar to the figure shown, depending on the user's input variable.



Time’s Up! Let’s Review.