**Task 1: Instructions**

Import pandas then load the data.

* Read the notebook on the right before the instructions here on the left.
* Import pandas under the alias pd.
* Load the dataset's CSV files ('datasets/super\_bowls.csv', 'datasets/tv.csv', and 'datasets/halftime\_musicians.csv') into DataFrames.

# Task 2: Instructions

Display and inspect the summaries of the TV and halftime musician DataFrames for issues.

* Use the .info() method to inspect the DataFrame tv.
* Use the .info() method to inspect the DataFrame halftime\_musicians.

The .info() method wasn't covered in Intermediate Python for Data Science so if you're stuck, check out the hint for the full solution.

You don't need to use display() or print() with .info() in Jupyter Notebooks because it prints to the output by default. The '\n' prints a blank line in between the .info() summaries to make them more readable.

# Task 3: Instructions

Plot a histogram of combined points then display the rows with the most extreme combined point outcomes.

* From matplotlib, import the pyplot module under the alias plt.
* Create a histogram of the combined\_pts column from the super\_bowls DataFrame.
* Select the Super Bowl(s) where the combined score was less than 25.

%matplotlib inline is a magic Jupyter Notebook command that allows you to display your graphs without plt.show(). You only need to use plt.show() in this notebook if you want to display the plot before other outputs (which you do in this task).

# Task 4: Instructions

Modify and display the histogram of point differences, then display the rows with the most extreme point difference outcomes.

* Add a y-label with 'Number of Super Bowls'.
* Display the plot with plt.show().
* Select the Super Bowl(s) where the point difference was equal to 1.
* Select the Super Bowl(s) where the point difference was greater than or equal to 35.

Helpful links:

* Labels [exercise](https://campus.datacamp.com/courses/intermediate-python-for-data-science/matplotlib?ex=14)

# Task 5: Instructions

Import seaborn and plot household share vs. point difference.

* Import the seaborn module under the alias sns.
* Fill in the x argument of sns.regplot() with the point difference column
* Fill in the y argument of sns.regplot() with the household share column.

Remember column names are represented as strings!

seaborn's regplot() is like scatter plot except more specialized for [visualizing linear relationships](https://seaborn.pydata.org/tutorial/regression.html#functions-to-draw-linear-regression-models). It draws a scatterplot, then fits a regression model and plots the resulting regression line and a 95% confidence interval for that regression.

# Task 6: Instructions

Create three line plots using the tv DataFrame to compare viewers, rating, and ad cost.

* For the first plot, plot super\_bowl on the x-axis, avg\_us\_viewers on the y-axis, and set the line color to '#648FFF'.
* For the second plot, plot super\_bowl on the x-axis, rating\_household on the y-axis, and set the line color to '#DC267F'.
* For the third plot, plot super\_bowl on the x-axis, ad\_cost on the y-axis, and set the line color to '#FFB000'.

The colors for the lines were based on a palette suggestion from [Coloring for Colorblindness](https://davidmathlogic.com/colorblind/).

# Task 7: Instructions

Filter and display the musicians for halftime shows up to and including Super Bowl XXVII.

* Using halftime\_musicians, select the musicians that performed in halftime shows up to and including Super Bowl XXVII (27) (i.e. Michael Jackson's performance).

The last line of code in a Jupyter Notebook cell automatically gets it output displayed so you don't need to use display() here.

# Task 8: Instructions

Select and display the musicians with more than one halftime show appearance.

* The new halftime\_appearances DataFrame has two columns, musician and super\_bowl, where super\_bowl now contains the halftime show counts for each musician. Select the musicians that have appeared in more than one halftime show.

The halftime\_appearances code is preloaded because it wasn't covered in the prerequisite for this project, [Intermediate Python for Data Science](https://www.datacamp.com/courses/intermediate-python-for-data-science). Grouping and rearranging data are covered in [Manipulating DataFrames with pandas](https://www.datacamp.com/courses/manipulating-dataframes-with-pandas).

# Task 9: Instructions

Modify the histogram of number of songs performed for non-band musicians.

* In the plt.hist() function, set the number of bins argument equal to most\_songs (the most number of songs performed in a halftime show by a single musician).
* Add an x-label with 'Number of Songs Per Halftime Show Performance'.

You can't filter out "Band" because Bruce Springsteen and the E Street Band performed at Super Bowl XLIII.

The no\_bands code is preloaded because it wasn't covered in [Intermediate Python for Data Science](https://www.datacamp.com/courses/intermediate-python-for-data-science). The .str.contains() method is covered in [Cleaning Data in Python](https://campus.datacamp.com/courses/cleaning-data-in-python/case-study-5?ex=9).

# Task 10: Instructions

Who will win Super Bowl LIII?

* The patriots and rams are playing in Super Bowl LIII. Assign the variable of the team you think will win to the super\_bowl\_LIII\_winner variable.

Congratulations on reaching the end of the project! You just applied your Python skills in a real-world data analysis. The structure of this project (where code intersperses narrative) is an excellent structure for blog posts to add to your data science portfolio.

To continue building your Python skills, continue to the next course in your track. If you're not enrolled in a track, pick a new course from DataCamp's Python [library](https://www.datacamp.com/courses/tech:python).