Instructions for the NFL notebook

Please read, understand the instructions and apply what you learn from the crash course. The following exercise uses 3 different datasets from the National Football League (NFL) compiled from WIKIPEDIA. The following exercise is from DATACAMP and created by David Venturi (Instructor at DataCamp). Any doubt be free to ask.

Task 1: Instructions

Import `pandas` then load the data.

- Read the notebook and understand the topic.
- 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`.

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.

Helpful links:

- `.info()` method **documentation**
- Inspecting a DataFrame **exercise** (in another course)

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 greater than 70.
- 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).

Helpful links:

- Basic plots with `matplotlib` **lesson**
- Histograms **lesson**
- Filtering Pandas DataFrame **lesson**

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.

Task 5: Instructions

Import `seaborn` and plot household share vs. point difference.

- Join game and TV data, filtering out SB I because it was split over two networks. Use `merge`.
- 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**. 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**.

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**. Grouping and rearranging data are covered in **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**. The `.str.contains()` method is covered in **Cleaning Data in Python**.

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.