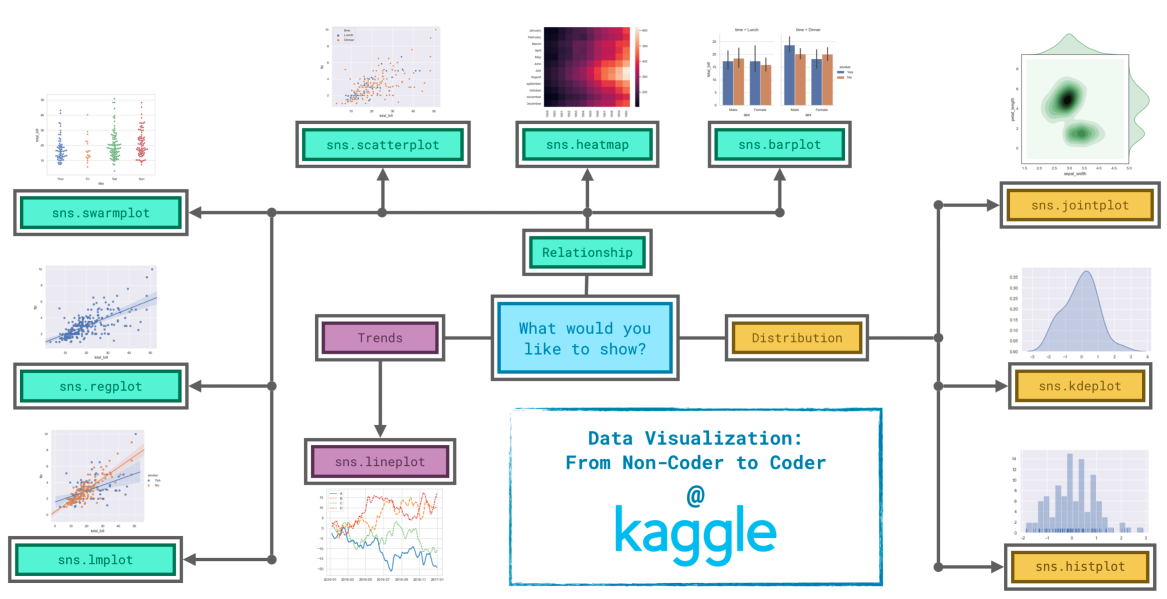
Data Visualization: 6th lesson – Choosing Plot Types & Custom Styles

In this course, you’ve learned how to create many different chart types. Now, you’ll organize your knowledge, before learning some quick commands that you can use to change the style of your charts.

***Lessons recap***



Since it's not always easy to decide how to best tell the story behind your data, we've broken the chart types into three broad categories to help with this:

* Trends

A trend is defined as a pattern of change. **Line charts** are best to show trends over a period of time, and multiple lines can be used to show trends in more than one group.

* Relationship

There are many different chart types that you can use to understand relationships between variables in your data.

* **Bar charts** are useful for comparing quantities corresponding to different groups.
* **Heatmaps** can be used to find color-coded patterns in tables of numbers.
* **Scatter plots** show the relationship between two continuous variables; if color-coded, we can also show the relationship with a third *categorical variable*.
* Including a **regression line** in the scatter plot makes it easier to see any linear relationship between two variables.
* Drawing multiple regression lines makes it easier to compare using scatter plot with multiple, color-coded groups.
* **Categorical scatter plots** show the relationship between a continuous variable and a categorical variable.
* Distribution

We visualize distributions to show the possible values that we can expect to see in a variable, along with how likely they are.

* **Histograms** show the distribution of a single numerical variable.
* **KDE plots** show an estimated, smooth distribution of a single numerical variable or two numerical variables.
* Simultaneously displaying a two-dimensional KDE plot with the corresponding KDE plots for each individual variable.

***Changing styles with seaborn***

All of the commands have provided a nice, default style to each of the plots. However, you may find it useful to customize how your plots look, and thankfully, this can be accomplished by just adding one more line of code!

As always, we need to begin by setting up the coding environment (this code is hidden, but you can un-hide it by clicking on the "Code" button immediately below this text, on the right).

import pandas as pd

pd.plotting.register\_matplotlib\_converters()

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

print("Setup Complete")

Setup Complete

We'll work with the same code that we used to create a line chart in a previous tutorial. The code below loads the dataset and creates the chart.

*# Path of the file to read*

spotify\_filepath = "../input/spotify.csv"

*# Read the file into a variable spotify\_data*

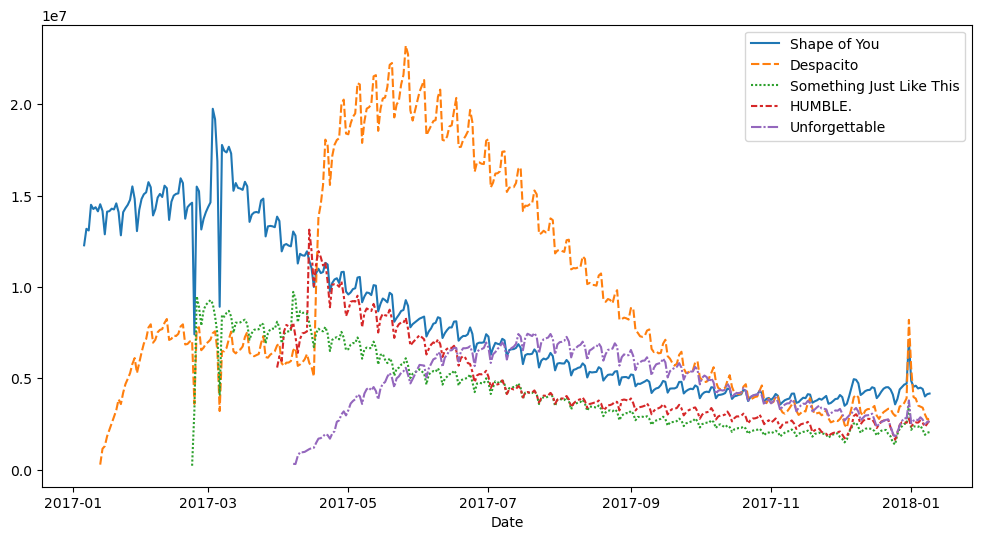
spotify\_data = pd.read\_csv(spotify\_filepath, index\_col="Date", parse\_dates=True)

*# Line chart*

plt.figure(figsize=(12,6))

sns.lineplot(data=spotify\_data)

<AxesSubplot:xlabel='Date'>



We can quickly change the style of the figure to a different theme with only a single line of code.

*# Change the style of the figure to the "dark" theme*

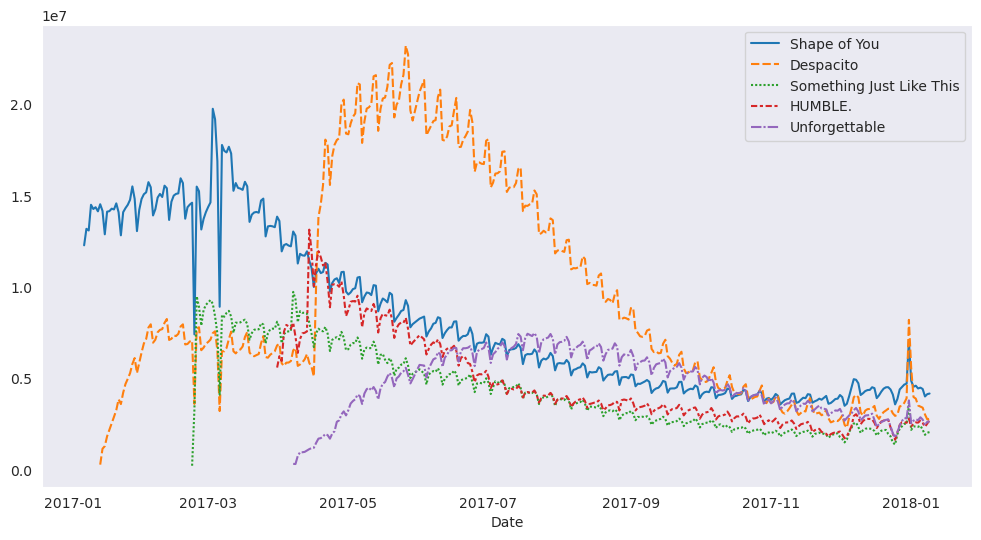
sns.set\_style("dark")

*# Line chart*

plt.figure(figsize=(12,6))

sns.lineplot(data=spotify\_data)

<AxesSubplot:xlabel='Date'>



Seaborn has five different themes: "darkgrid", "whitegrid", "dark", "white", and "ticks", and you need only use a command similar to the one in the code cell above (with the chosen theme filled in) to change it.