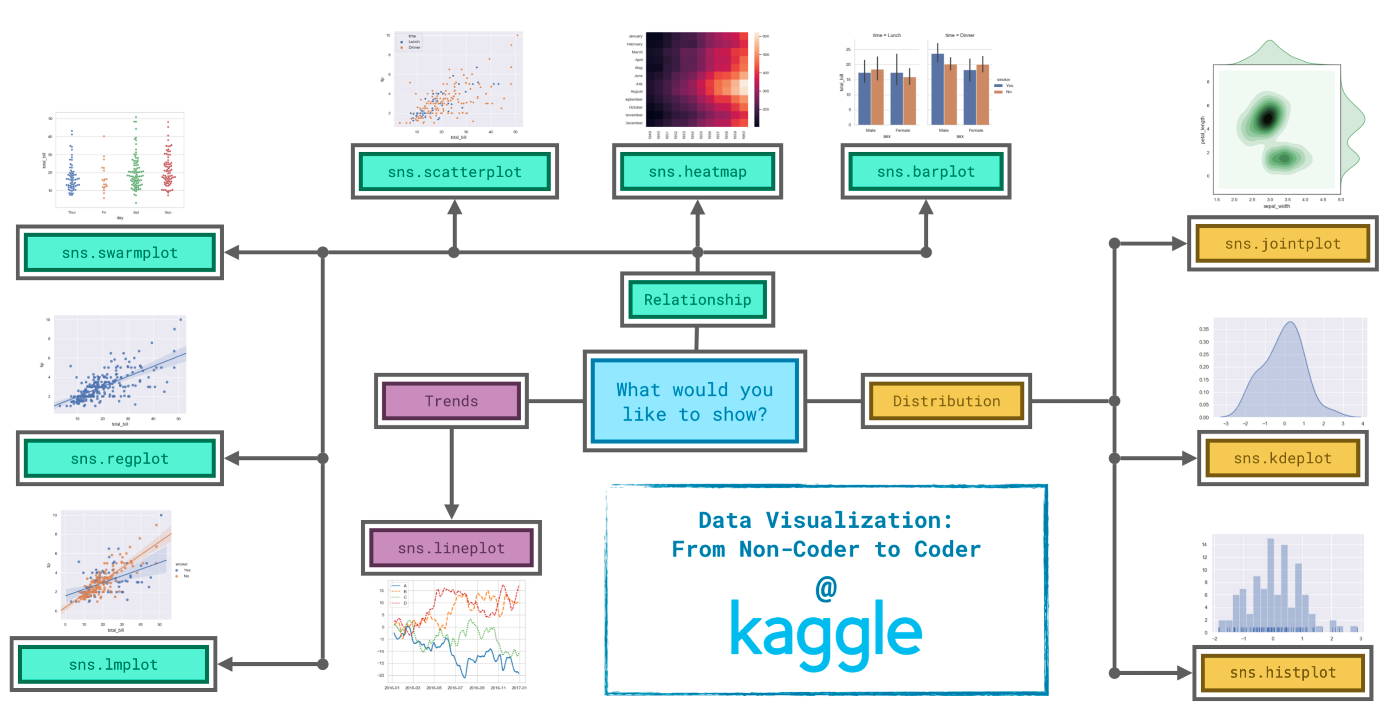
Bu kursta, birçok farklı grafik türünü nasıl oluşturacağınızı öğrendiniz. Şimdi, grafiklerinizin stilini değiştirmek için kullanabileceğiniz bazı hızlı komutları öğrenmeden önce, bilgilerinizi düzenleyeceksiniz.

What have you learned?



Verilerinizdeki hikayeyi en iyi nasıl anlatacağınıza karar vermek her zaman kolay olmadığından, bu konuda size yardımcı olmak için grafik türlerini üç geniş kategoriye ayırdık.

### Eğilimler (Trends)

Eğilim, bir değişim örüntüsü olarak tanımlanır.

* **sns.lineplot**: **Çizgi grafikleri**, belirli bir zaman dilimindeki eğilimleri göstermek için en iyisidir ve birden fazla gruptaki eğilimleri göstermek için çoklu çizgiler kullanılabilir.

### İlişki (Relationship)

Verilerinizdeki değişkenler arasındaki ilişkileri anlamak için kullanabileceğiniz birçok farklı grafik türü vardır.

* **sns.barplot**: **Çubuk grafikleri**, farklı gruplara karşılık gelen nicelikleri karşılaştırmak için kullanışlıdır.
* **sns.heatmap**: **Isı haritaları**, sayı tablolarında renk kodlu örüntüler bulmak için kullanılabilir.
* **sns.scatterplot**: **Nokta grafikleri**, iki sürekli değişken arasındaki ilişkiyi gösterir; eğer renk kodluysa, üçüncü bir **kategorik değişken** ile olan ilişkiyi de gösterebiliriz.
* **sns.regplot**: Nokta grafiğine bir **regresyon çizgisi** eklemek, iki değişken arasındaki doğrusal ilişkiyi görmeyi kolaylaştırır.
* **sns.lmplot**: Bu komut, nokta grafiği birden fazla renk kodlu grup içeriyorsa, çoklu regresyon çizgileri çizmek için kullanışlıdır.
* **sns.swarmplot**: **Kategorik nokta grafikleri**, sürekli bir değişken ile kategorik bir değişken arasındaki ilişkiyi gösterir.

### Dağılım (Distribution)

Dağılımları, bir değişkende görmeyi bekleyebileceğimiz olası değerleri ve bunların ne kadar olası olduğunu göstermek için görselleştiririz.

* **sns.histplot**: **Histogramlar**, tek bir sayısal değişkenin dağılımını gösterir.
* **sns.kdeplot**: **KDE grafikleri** (veya **2D KDE grafikleri**), tek bir sayısal değişkenin (veya iki sayısal değişkenin) tahmini, pürüzsüz bir dağılımını gösterir.
* **sns.jointplot**: Bu komut, 2D bir KDE grafiğini, her bir değişken için ilgili KDE grafikleriyle birlikte eş zamanlı olarak görüntülemek için kullanışlıdır.

# Changing styles with seaborn

Tüm komutlar, her bir grafiğe hoş, varsayılan bir stil sağladı. Ancak, grafiklerinizin görünümünü özelleştirmeyi faydalı bulabilirsiniz ve neyse ki bu, yalnızca bir kod satırı daha eklenerek gerçekleştirilebilir!

Her zaman olduğu gibi, kodlama ortamını ayarlayarak başlamamız gerekiyor. (Bu kod gizlidir, ancak bu metnin hemen altında, sağdaki "Kod" düğmesine tıklayarak gizliliğini kaldırabilirsiniz.)

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

Önceki bir eğitimde çizgi grafik oluşturmak için kullandığımız aynı kodla çalışacağız. Aşağıdaki kod, veri setini yükler ve grafiği oluşturur.

*# 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'>

Sadece tek bir kod satırıyla şeklin stilini farklı bir temaya kolayca değiştirebiliriz.

*# 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'un beş farklı teması vardır: (1) **"darkgrid"**, (2) **"whitegrid"**, (3) **"dark"**, (4) **"white"** ve (5) **"ticks"**. Temayı değiştirmek için, yukarıdaki kod hücresindekine benzer bir komut kullanmanız (seçtiğiniz temayı doldurarak) yeterlidir.

Yaklaşan alıştırmada, en çok hangisini beğendiğinizi görmek için bu temalarla denemeler yapacaksınız!