Artık kendi çizgi grafiklerinizi oluşturabildiğinize göre, daha fazla grafik türünü öğrenme zamanı!

Bu arada, Python'da ilk defa kod yazıyorsanız, şimdiye kadar başardıklarınızla **gurur duymalısınız**, çünkü tamamen yeni bir beceri öğrenmek asla kolay değildir! Kursa devam ederseniz, her şeyin daha da kolaylaştığını (ve oluşturacağınız grafiklerin daha etkileyici hale geldiğini) fark edeceksiniz, çünkü tüm grafikler için kodlar oldukça benzer. Her beceri gibi, kodlama da zamanla ve tekrarla doğal hale gelir.

Set up the notebook

Her zamanki gibi, kodlama ortamını ayarlayarak başlıyoruz. (Bu kod gizlidir, ancak bu metnin hemen altında, sağda bulunan "Kod" butonuna tıklayarak onu tekrar görünür hale getirebilirsiniz.)

import pandas as pd

pd.plotting.register\_matplotlib\_converters()

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

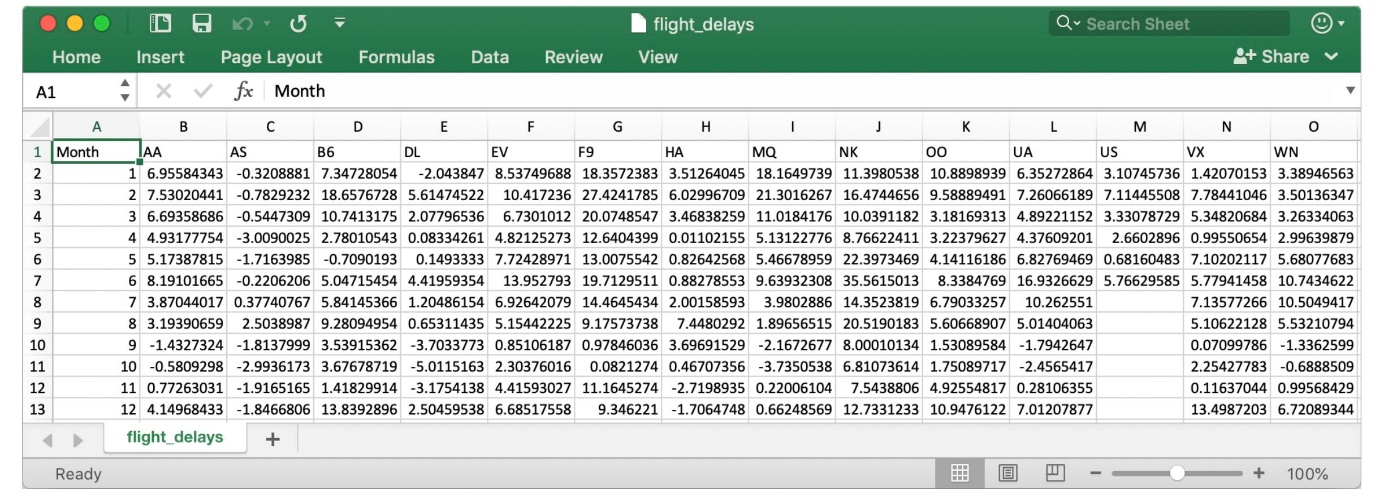
print("Setup Complete")

Setup Complete

# Select a dataset

Bu eğitimde, ABD Ulaştırma Bakanlığı'nın uçuş gecikmelerini takip eden bir veri setiyle çalışacağız.

Bu CSV dosyasını Excel'de açtığınızda, her ay için bir satır (1 = Ocak, 2 = Şubat vb.) ve her havayolu kodu için bir sütun görürsünüz.



Her bir girdi, farklı bir havayolu ve ay için (tümü 2015 yılına ait) ortalama varış gecikmesini (dakika cinsinden) gösterir. Negatif girdiler, (ortalama olarak) erken varmaya eğilimli olan uçuşları belirtir. Örneğin, Ocak ayındaki ortalama American Airlines uçuşu (havayolu kodu: **AA**) yaklaşık 7 dakika geç varmış, Nisan ayındaki ortalama Alaska Airlines uçuşu (havayolu kodu: **AS**) ise yaklaşık 3 dakika erken varmıştır.

# Load the data

Daha önce olduğu gibi, veri setini pd.read\_csv komutunu kullanarak yüklüyoruz.

*# Path of the file to read*

flight\_filepath = "../input/flight\_delays.csv"

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

flight\_data = pd.read\_csv(flight\_filepath, index\_col="Month")

Kodun, önceki derste kullandığımızdan biraz daha kısa olduğunu fark edebilirsiniz. Bu durumda, satır etiketleri ('Month' sütunundan) tarihlere karşılık gelmediği için, parantez içine **parse\_dates=True** eklemiyoruz. Ancak, hem veri setinin dosya yolunu (bu durumda, **flight\_filepath**) hem de satırları indekslemek için kullanılacak sütunun adını (bu durumda, **index\_col="Month"**) sağlamak için önceki gibi ilk iki metin parçasını koruyoruz.

# Examine the data[¶](https://www.kaggle.com/code/alexisbcook/bar-charts-and-heatmaps" \l "Examine-the-data" \t "_self)

Veri kümesi küçük olduğundan, tüm içeriğini kolayca yazdırabiliriz. Bunu, yalnızca veri kümesinin adını içeren tek bir satır kod yazarak yapabiliriz.

*# Print the data*

flight\_data

# Bar chart

Spirit Airlines (havayolu kodu: NK) uçuşları için ay bazında ortalama varış gecikmesini gösteren bir çubuk grafik oluşturmak istediğimizi varsayalım.

*# Set the width and height of the figure*

plt.figure(figsize=(10,6))

*# Add title*

plt.title("Average Arrival Delay for Spirit Airlines Flights, by Month")

*# Bar chart showing average arrival delay for Spirit Airlines flights by month*

sns.barplot(x=flight\_data.index, y=flight\_data['NK'])

*# Add label for vertical axis*

plt.ylabel("Arrival delay (in minutes)")

Text(0, 0.5, 'Arrival delay (in minutes)')

Metni (başlık ve dikey eksen etiketi) ve şeklin boyutunu özelleştirme komutları önceki eğitimden tanıdık geliyor. Çubuk grafiği oluşturan kod ise yeni:

*# Bar chart showing average arrival delay for Spirit Airlines flights by month*

sns.barplot(x=flight\_data.index, y=flight\_data['NK'])

Üç ana bileşeni vardır:

* **sns.barplot**: Bu, notebook'a bir çubuk grafik oluşturmak istediğimizi söyler. **sns**'in **seaborn** paketine atıfta bulunduğunu ve bu kursta grafik oluşturmak için kullandığınız tüm komutların bu önekle başlayacağını unutmayın.
* **x=flight\_data.index**: Bu, yatay eksende neyin kullanılacağını belirler. Bu durumda, satırları indeksleyen sütunu (yani ayları içeren sütun) seçtik.
* **y=flight\_data['NK']**: Bu, her bir çubuğun yüksekliğini belirlemek için veri setindeki hangi sütunun kullanılacağını ayarlar. Bu durumda, **'NK'** sütununu seçiyoruz.

**Önemli Not:** İndeksleme sütununu **flight\_data.index** ile seçmelisiniz ve **flight\_data['Month']** kullanmak mümkün değildir (bu bir hata döndürecektir). Bunun nedeni, veri setini yüklediğimizde, "Month" sütununun satırları indekslemek için kullanılmış olmasıdır. İndeksleme sütununu seçmek için her zaman bu özel gösterimi kullanmak zorundayız.

# Heatmap

Öğreneceğimiz bir grafik türü daha var: **ısı haritaları**!

Aşağıdaki kod hücresinde, **flight\_data**'daki örüntüleri hızlı bir şekilde görselleştirmek için bir ısı haritası oluşturuyoruz. Her hücre, karşılık gelen değerine göre renk kodludur.

*# Set the width and height of the figure*

plt.figure(figsize=(14,7))

*# Add title*

plt.title("Average Arrival Delay for Each Airline, by Month")

*# Heatmap showing average arrival delay for each airline by month*

sns.heatmap(data=flight\_data, annot=True)

*# Add label for horizontal axis*

plt.xlabel("Airline")

Text(0.5, 47.7222222222222, 'Airline')

Isı haritasını oluşturmak için gerekli kod aşağıdaki gibidir:

*# Heatmap showing average arrival delay for each airline by month*

sns.heatmap(data=flight\_data, annot=True)

Bu kodun üç ana bileşeni vardır:

* **sns.heatmap**: Bu, notebook'a bir ısı haritası oluşturmak istediğimizi söyler.
* **data=flight\_data**: Bu, notebook'a ısı haritasını oluşturmak için **flight\_data**'daki tüm girdileri kullanmasını söyler.
* **annot=True**: Bu, her hücrenin değerlerinin grafikte görünmesini sağlar. (Bunu dahil etmezseniz, sayılar hücrelerden kalkar!)

Peki, tabloda hangi örüntüleri görebilirsiniz? Örneğin, dikkatli bakarsanız, yılın sonuna doğru olan aylar (özellikle 9-11. aylar) tüm havayolları için nispeten koyu renkte görünür. Bu, havayollarının bu aylarda (ortalama olarak) tarifeye daha iyi uyduğunu gösteriyor!