Data Visualization: 3rd lesson – Bar Charts & Heatmaps

Now that you can create your own line charts, it's time to learn about more chart types!

By the way, if this is your first experience with writing code in Python, you should be very proud of all that you have accomplished so far, because it's never easy to learn a completely new skill! If you stick with the course, you'll notice that everything will only get easier (while the charts you'll build will get more impressive!), since the code is pretty similar for all of the charts. Like any skill, coding becomes natural over time, and with repetition.

In this tutorial, you'll learn about **bar charts** and **heatmaps**.

***Set up the notebook***

As always, we 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

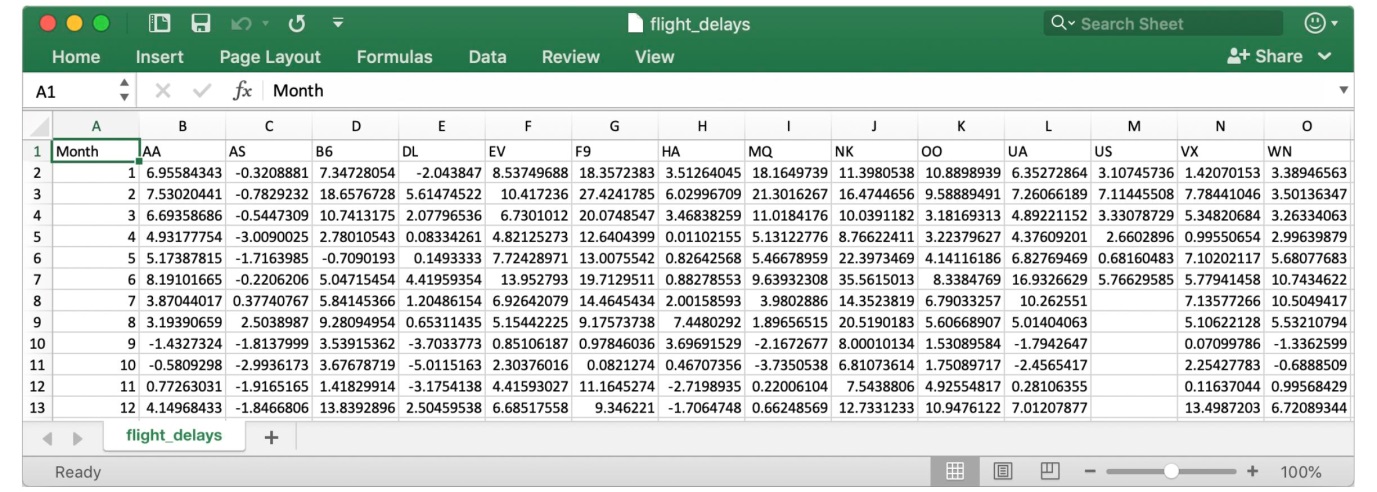
import seaborn as sns

print("Setup Complete")

Setup Complete

***Select a dataset***

In this tutorial, we'll work with a dataset from the US Department of Transportation that tracks flight delays. Opening this CSV file in Excel shows a row for each month (where 1 = January, 2 = February, etc.) and a column for each airline code.



Each entry shows the average arrival delay (in minutes) for a different airline and month (all in year 2015). Negative entries denote flights that (on average) tended to arrive early. For instance, the average American Airlines flight (airline code: **AA**) in January arrived roughly 7 minutes late, and the average Alaska Airlines flight (airline code: **AS**) in April arrived roughly 3 minutes early.

***Load the data***

As before, we load the dataset using the pd.read\_csv command.

*# 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")

You may notice that the code is slightly shorter than what we used in the previous tutorial. In this case, since the row labels (from the 'Month' column) don't correspond to dates, we don't add parse\_dates=True in the parentheses. But, we keep the first two pieces of text as before, to provide both:

* The *filepath* for the dataset (in this case, flight\_filepath)
* The name of the column that will be used to index the rows (in this case, index\_col=”Month”)

***Examine the data***

Since the dataset is small, we can easily print all of its contents. This is done by writing a single line of code with just the name of the dataset.

*# Print the data*

flight\_data

AA AS B6 DL EV F9 HA MQ NK OO

UA US VX WN

Month

1 6.955843 -0.320888 7.347281 -2.043847 8.537497 18.357238 3.512640 18.164974 11.398054 10.889894 6.352729 3.107457 1.420702 3.389466

2 7.530204 -0.782923 18.657673 5.614745 10.417236 27.424179 6.029967 21.301627 16.474466 9.588895 7.260662 7.114455 7.784410 3.501363

3 6.693587 -0.544731 10.741317 2.077965 6.730101 20.074855 3.468383 11.018418 10.039118 3.181693 4.892212 3.330787 5.348207 3.263341

4 4.931778 -3.009003 2.780105 0.083343 4.821253 12.640440 0.011022 5.131228 8.766224 3.223796 4.376092 2.660290 0.995507 2.996399

5 5.173878 -1.716398 -0.709019 0.149333 7.724290 13.007554 0.826426 5.466790 22.397347 4.141162 6.827695 0.681605 7.102021 5.680777

6 8.191017 -0.220621 5.047155 4.419594 13.952793 19.712951 0.882786 9.639323 35.561501 8.338477 16.932663 5.766296 5.779415 10.743462

7 3.870440 0.377408 5.841454 1.204862 6.926421 14.464543 2.001586 3.980289 14.352382 6.790333 10.262551 NaN 7.135773 10.504942

8 3.193907 2.503899 9.280950 0.653114 5.154422 9.175737 7.448029 1.896565 20.519018 5.606689 5.014041 NaN 5.106221 5.532108

9 -1.432732 -1.813800 3.539154 -3.703377 0.851062 0.978460 3.696915 -2.167268 8.000101 1.530896 -1.794265 NaN 0.070998 -1.336260

10 -0.580930 -2.993617 3.676787 -5.011516 2.303760 0.082127 0.467074 -3.735054 6.810736 1.750897 -2.456542 NaN 2.254278 -0.688851

11 0.772630 -1.916516 1.418299 -3.175414 4.415930 11.164527 -2.719894 0.220061 7.543881 4.925548 0.281064 NaN 0.116370 0.995684

12 4.149684 -1.846681 13.839290 2.504595 6.685176 9.346221 -1.706475 0.662486 12.733123 10.947612 7.012079 NaN 13.498720 6.720893

***Bar chart***

Say we'd like to create a bar chart showing the average arrival delay for Spirit Airlines (airline code: **NK**) flights, by month.

*# 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")

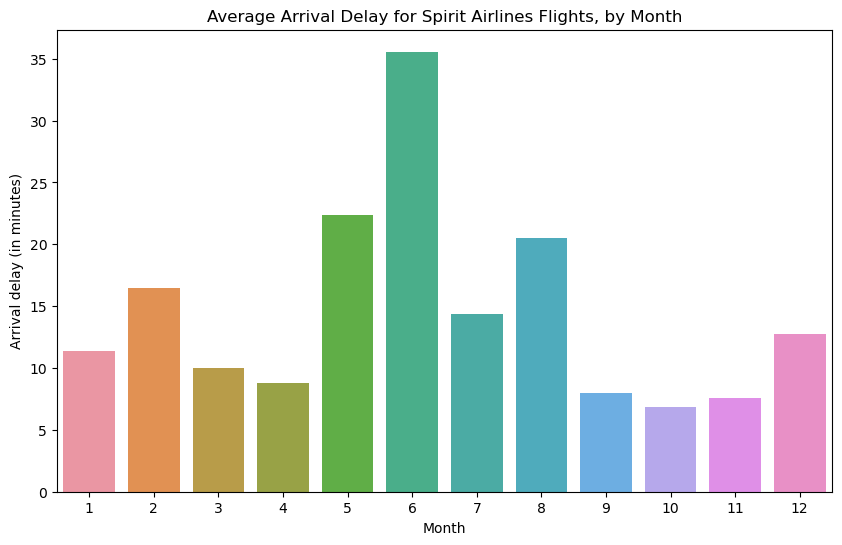
*# Bar chart showing average arrival delay for Spirit Airlines flights by month*

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)')



The commands for customizing the text (title and vertical axis label) and size of the figure are familiar from the previous tutorial. The code that creates the bar chart is new.

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

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

It has three main components:

* sns.barplot

This tells the notebook that we want to create a bar chart.

* x=flight\_data.index

This determines what to use on the horizontal axis. In this case, we have selected the column that indexes the rows (in this case, the column containing the months).

* y=flight\_data['NK']

This sets the column in the data that will be used to determine the height of each bar. In this case, we select the 'NK' column.

Important note:

You must select the indexing column with flight\_data.index, and it is not possible to use flight\_data['Month'] (which will return an error). This is because when we loaded the dataset, the "Month" column was used to index the rows. We always have to use this special notation to select the indexing column.

***Heatmap***

We have one more plot type to learn about: heatmaps!

In the code cell below, we create a heatmap to quickly visualize patterns in flight\_data. Each cell is color-coded according to its corresponding value.

*# 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")

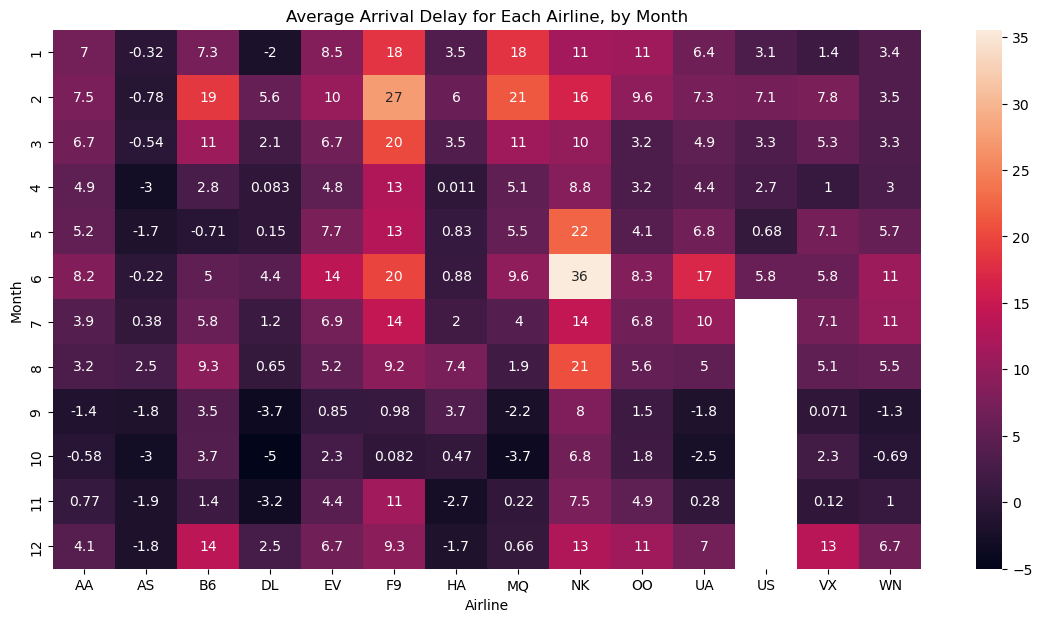
*# Heatmap showing average arrival delay for each airline by month*

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

*# Add label for horizontal axis*

plt.xlabel("Airline")

Text(0.5, 47.7222222222222, 'Airline')



The relevant code to create the heatmap is as follows.

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

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

This code has three main components:

* sns.heatmap

This tells the notebook that we want to create a heatmap.

* data=flight\_data

This tells the notebook to use all of the entries in flight\_data to create the heatmap.

* annot=True

This ensures that the values for each cell appear on the chart (leaving this out removes the numbers from each of the cells!)

What patterns can you detect in the table? For instance, if you look closely, the months toward the end of the year (especially months 9-11) appear relatively dark for all airlines. This suggests that airlines are better (on average) at keeping schedule during these months!