5.1.2   The Matplotlib Library

You've just received the work assignment of a lifetime: creating visualizations of rideshare data for PyBer to help improve access to ride-sharing services and determine affordability for underserved neighborhoods. As a new employee, this is a huge professional opportunity for you!

The first step is to make sure you have the right tools in place. A simple Excel chart isn't going to impress V. Isualize. Instead, you'll use the Python graphing library Matplotlib, which is a favorite tool among data scientists and data analysts because of its robust visualization features.

Omar has given you some backstory about the connection between V. Isualize and Matplotlib. Matplotlib was created as a Python alternative for MATLAB. MATLAB, which is short for matrix laboratory, was developed by the company MathWorks in the 1980s. It enabled scientists to perform linear algebra and numerical analysis without learning the programming language Fortran, which until then had been the best option for complex computations.

With this information, you realize that the CEO is an expert in creating data visualizations. You definitely can't afford to botch this up!

Matplotlib is a graphing and plotting library for Python that comes with the Anaconda installation. To use it with Jupyter Notebook, all we need to do is import it. Matplotlib is often used with another Python library, NumPy, a numerical mathematics library for making arrays or matrices. A **matrix** is a two-dimensional data structure, or a list of lists, in which numbers are arranged into rows and columns.

The following is an example of a 2 x 3 matrix. In this matrix there are two lists, `[1, 4, 7]` and `[-5, -2, 1]`.

```
A = [[1, 4, 7],
    [-5, -2, 1]]
```

Matplotlib has an advantage over Excel when it comes to graphing because of the robust features it offers. You can use it to create a wide range of graphs, including line plots, bar plots, scatter plots, bubble charts, pie charts, histograms, 3D plots, log plots, and polar plots—to name just a few.

Matplotlib also has rich styling options. You can customize annotations for chart axes, titles and legends, the color and size of lines, bars, and bubbles, and the chart's background. It also lets you save and print publication-quality charts. Because of these features, Matlplotlib is one of the most popular plotting libraries for Python.

Matplotlib has two methods for graphing data. One uses MATLAB's plotting syntax and functionality and the other uses an object-oriented interface. MATLAB's plotting syntax is concise and the most useful when creating simple plots that require little coding. This method is most effective when graphing data directly from a DataFrame. The object-oriented method is better suited to more complicated graphs that require more coding, such as those with multiple lines or bars, or multiple plots in one graph.

For now, we'll practice graphing simple plots like line, bar, scatter, and pie charts, and annotating charts. We'll use Matplotlib's plotting methods in the Jupyter Notebook environment with data from the ride-sharing dataset.

By the end of this module, you will be able to create a variety of visualizations. Watch the following video to learn more about the charts you'll be working on.
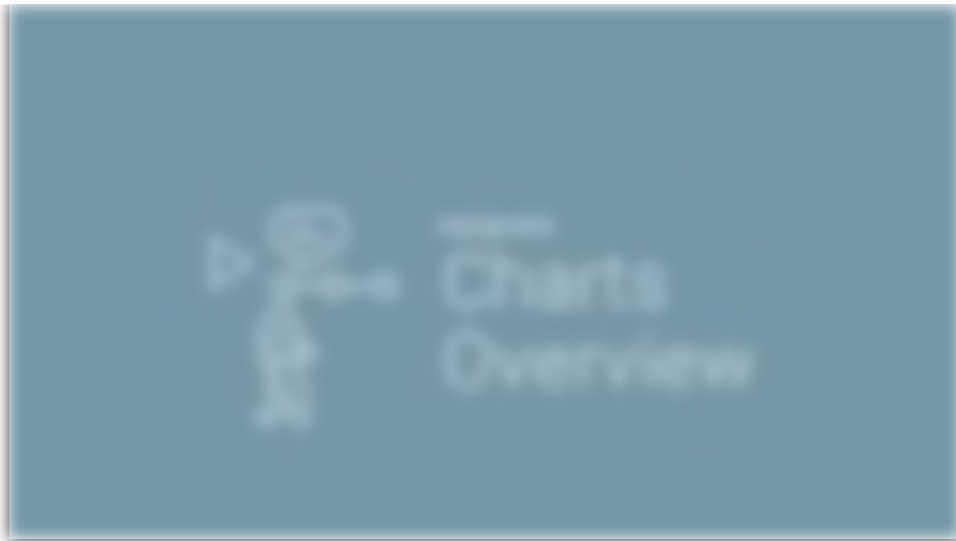
Check the Version of Matplotlib

Before we get started on any project, it's good practice to make sure we have the latest version of the software we'll be using. Because we'll be using Matplotlib, let's check to make sure we have version stable or greater.

Follow the instructions below for your operating system.

Check out the macOS instructions below, or jump to the **Windows instructions**.

Check the Version on the Command Line in macOS

To begin, launch the command line and activate the PythonData environment.

REWIND

To activate the PythonData environment on the command line, type `conda activate PythonData`.

The command line should look something like this:

```
(PythonData) computer_name:~ home_directory$
```

At the prompt, type `$ python` to launch Python. The command line should look similar to this:

```
(PythonData) computer_name:~ home_directory$ python
Python 3.7.6 |Anaconda, Inc.| (default, Dec 29 2018, 19:04:46)
[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE_401/final)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

At the Python prompt (`>>>`), type `import matplotlib` and press Enter to import Matplotlib.

Next, to check the version of Matplotlib, type `matplotlib.__version__` (there are two underscores before and after "version") and press Enter:
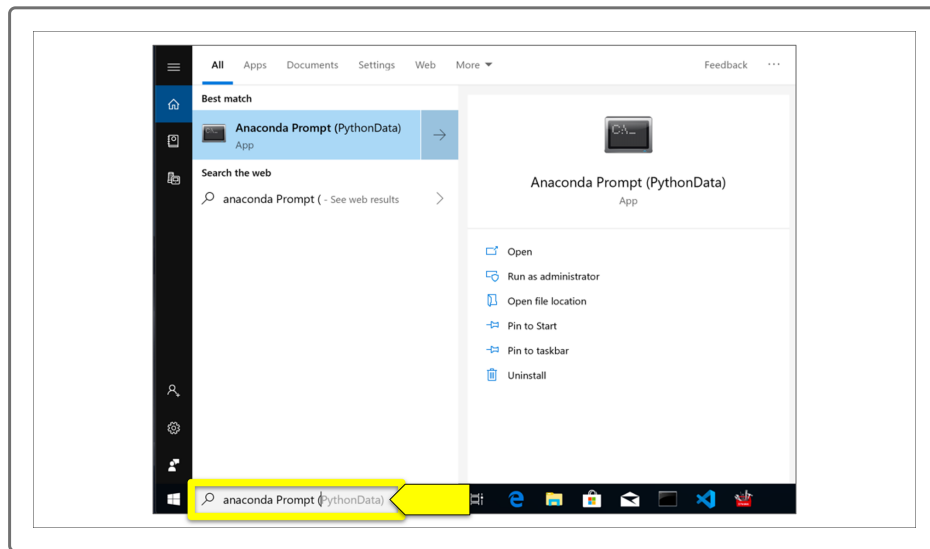
```
>>> matplotlib.__version__
```

The output should be stable or greater.

NOTE

> To update Matplotlib for your development environment; with your PythonData environment activated, type `conda install -c conda-forge matplotlib` at the command prompt and press Enter.

Check the Version on the Command Line in Windows

In the search menu, type and launch "Anaconda Prompt (PythonData)".

The Anaconda prompt will look something like this:

```
(PythonData) C:\Users\your_home_directory>
```

After the Python prompt (`>`), type `python` to launch Python. Your Anaconda prompt should look similar to this:

```
(PythonData) C:\Users\your_home_directory>python
Python 3.7.6 |Anaconda, Inc.| ((default, Feb 21 2019, 18:30:04) [MSC v.1916
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Now you'll import the Matplotlib library. At the Python prompt (`>>>`), type `import matplotlib` and press Enter.

Finally, to check the version of Matplotlib, type `matplotlib.__version__` (there are two underscores before and after "version") and press Enter:

```
>>> matplotlib.__version__
```

The output should be stable or greater.

NOTE

To update Matplotlib for your development environment; with your PythonData environment activated, type `conda install -c conda-forge matplotlib` at the command prompt and press Enter.

Check the Version in Jupyter Notebook

Alternatively, you can check the version of Matplotlib in Jupyter Notebook. To do that, follow these directions

REWIND

To start Jupyter Notebook, navigate to the Class folder on your computer using the command line or Anaconda prompt.

Activate the PythonData environment if it's not activated. Type and run `jupyter notebook`.

In Jupyter Notebook, create a new file if one hasn't been created. Add the following code to a new cell.

```
import matplotlib
matplotlib.__version__
```

When you run the cell, the output should be stable or later.

Now that you have an updated version of Matplotlib, let's dive into creating some visualizations!

NOTE

For more information about the latest Matplotlib version, see the **Matplotlib documentation (https://matplotlib.org/stable/index.html)** .