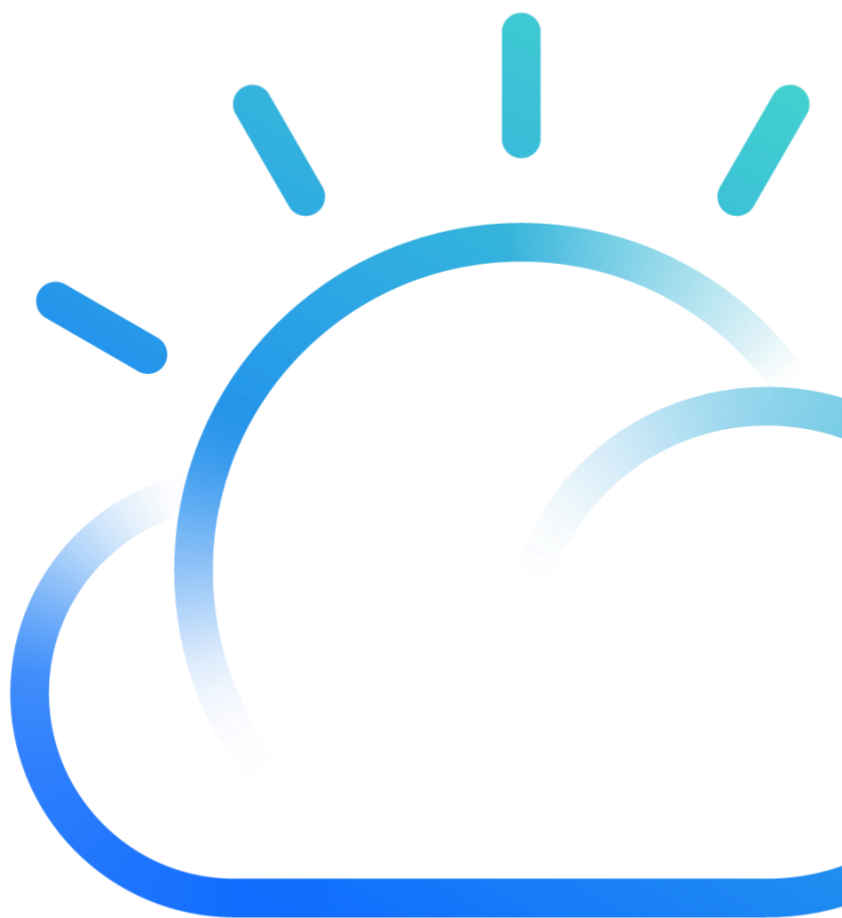


Lab Guide

*Automotive Data using
Jupyter Notebooks, Python
and Brunel Visualization*



The information contained in this document has not been submitted to any formal IBM test and is distributed on an “as is” basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

© Copyright International Business Machines Corporation 2019.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

SECTION 1. **PREFACE4**
 OVERVIEW.....4
 PREREQUISITES4

SECTION 2. **GETTING STARTED WITH WATSON STUDIO6**
 FIRST TIME ACCESSING WATSON STUDIO.....6
 RETURNING TO WATSON STUDIO6

SECTION 3. **STARTING WITH THE JUPYTER BOOKS.....7**

SECTION 4. **VISUALIZE THE DATA.....9**

Section 1. Preface

The purpose of this lab

By now you have used the data wrangling tools of Watson Studio and are familiar with a somewhat arduous, yet necessary task of cleansing, preparing and transforming your data. The activities below are typically done by the Data Engineer with close consultation with the Data journalist and the data scientist. In this lab, the Data Engineer performs many of the tasks that you completed in the prior exercises, except that it is done using Jupyter Notebooks.

Objectives

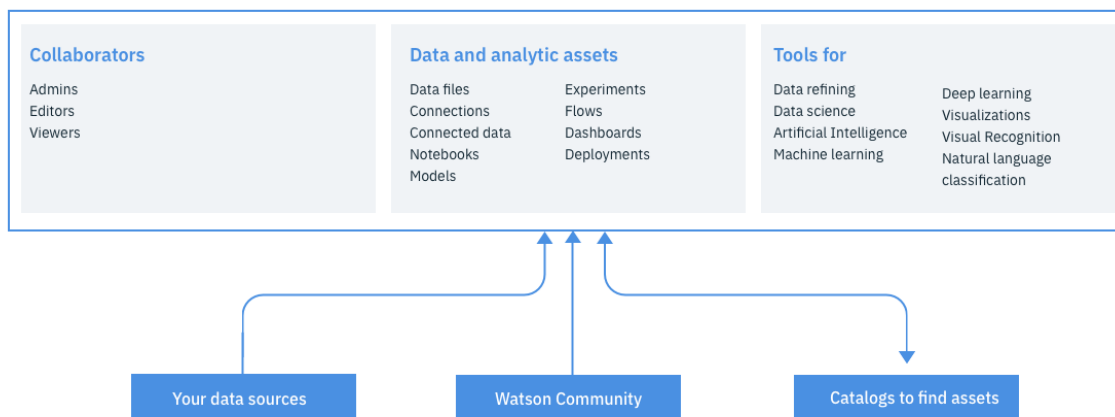
The objective of this lab is to complete the data cleansing and transformation steps that you performed in the earlier labs, except using Jupyter Notebooks as your development environment.

- Create a new project in Watson Studio
- Import into Pandas dataframe the automobile data set from the prior exercise
- Use open source technologies, such as Brunel to visualize your data

Overview

Watson Studio provides you with the environment and tools to solve your business problems by collaboratively working with data. You can choose the tools you need to analyze and visualize data, to cleanse and shape data, to ingest streaming data, or to create, train, and deploy machine learning models. This illustration shows how the architecture of Watson Studio is centered around the project. A project is where you organize your resources and work with data.

Projects: work with data



Prerequisites

These are the most important resources in a project:

- Collaborators are the team who works with the data. Three roles provide [different permissions](#).
- Data assets point to your data. Here's what you can do to prepare your data:
 - Access data from connections to your [cloud or on-premises data sources](#)
 - Access data from your organization's [catalogs](#)
 - Upload files to the project's object storage
 - Cleanse and shape data with the [Data Refinery tool](#)

- Analytical assets and tools are how you derive insights from data. You customize your project with [the tools](#) you need. Here's what you can do to analyze your data:
 - Analyze data with [Jupyter notebooks](#) or [RStudio](#).
 - Build, train, test, and deploy machine learning and deep learning [models](#).
 - Run [deep learning model experiments](#) in parallel with neural networks.
 - [Classify images](#) by training deep learning models to recognize image content.
 - Create and share [dashboards](#) of data visualizations without coding.
 - [Classify text](#) by training a model to classify text according to classes you define.

You can also bring in data and analytic assets from the IBM Watson [Community](#).

Section 2. Getting Started with Watson Studio

By now you have already registered with IBM Cloud and applied your promo code. Let's begin our journey:

1. Login into IBM Cloud: <https://console.bluemix.net/catalog/>

Note: Ensure that you have the [promocode](#) applied.

First time accessing Watson Studio	Returning to Watson Studio
Follow either one of the paths outlined below depending on whether you are new to Watson Studio or that you have already provisioned the service and are about to work with projects	
<ol style="list-style-type: none"> Click the Catalog tab and remove the label:lite filter. Search for the Watson Studio service and click that tile. Click Create. Click the Get Started button and when Done, click Get Started. Click the New Project tile. Select the Complete tile. Click OK. Specify a name. In this example, it is Automotive data. Click Add to add a storage. While the Lite plan is selected, click Create. In the Confirm Creation dialog, select the default region. Click Refresh. Now you have a cloud object storage available to you, click Create. Continue with the steps below. 	<ol style="list-style-type: none"> Access the Dashboard from the menu icon in the top left corner Search for the previously created Watson Studio service and open it. Click the Get Started button. Open the latest existing project, for example: Automotive Data engineering. Continue with the steps below.

Section 3. Starting with the Jupyter books

1. From the **Add to project** drop-down list (top right), select **Notebook**.
2. Again, specify a new Notebook name; for example, Automotive Data and choose Python 3.6 from the drop-down list and click **Create Notebook**.

You are now in your notebook and ready to start working. When you use a notebook in Watson Studio, you can run a cell only by selecting it, then going to the toolbar and clicking on the Run Cell (▶) button. When a cell is running, an [*] is shown beside the cell. Once the cell has finished the asterisks is replaced by a number.

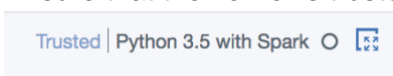
If you don't see the Jupyter toolbar showing the Run Cell (▶) button and other notebook controls, you are not in edit mode. Go to the dark blue toolbar above the notebook and click the edit (pencil) icon.


The Brunel Visualization language makes it easy to build interactive charts and diagrams that you can deploy rapidly. This notebook contains the steps and code to get you started with visualizing data with Brunel.

Some familiarity with Python is recommended. This notebook runs on Python 3 with Spark.

You will use data about cars to graph the relationships between various properties, for example, how horsepower affects gas mileage. The cars data set was used for the 1983 American Statistical Association Data Exposition. This data set was collected by Ernesto Ramos and David Donoho and obtained from StatLib.

3. Ensure that the kernel is trusted and running. Click the untrusted link and change it to trusted:



4. The first step is to obtain the data. Click **Find add data** .
5. From the **Files** section, select a csv file; for example, the **Car performance data.csv**.
6. Select **Insert pandas dataframe** option.

7. Click Run (▶). Allow enough time for the kernel to complete its function.

The screenshot shows a Jupyter notebook interface with a code cell being executed. The code imports sys, types, and pip, then checks for 'ibm-cos-sdk' and installs it if not present. It then imports pandas and boto3, and defines a function to read a CSV file from IBM Cloud Object Storage. The output shows the first five rows of the CSV data.

```

In [7]:
import sys
import types
import pip

if not('ibm-cos-sdk' in [package.project_name for package in pip.get_installed_distributions()]):
    pip install ibm-cos-sdk==2.0.0 -q

import pandas as pd
from boto3.client import Config
import boto3

def __iter__(self): return 0

# hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove these credentials before you share your notebook.
client_83e962ab6ce84e5fae94250fc219e400 = boto3.client(service_name='s3',
    ibm_api_key_id='c330x2NUTu3pm0-J4wx5Y-oMY87lpTkV8JG52m2to',
    ibm_auth_endpoint='https://iam.ng.bluemix.net/oidc/token',
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3-api.us-geo.objectstorage.service.networklayer.com')

body = client_83e962ab6ce84e5fae94250fc219e400.get_object(Bucket='automotivedataengineering-donotdelete-pr-sjzsl3tydolqb9',Key='Car performance data.csv')
# add missing __iter__ method, so pandas accepts body as file-like object
if not hasattr(body, '__iter__'): body.__iter__ = types.MethodType( __iter__, body )

df_data_1 = pd.read_csv(body)
df_data_1.head()

```

Out[7]:

	mpg	cylinders	engine	horsepower	weight	acceleration	year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	American	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	American	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	American	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	American	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	American	ford torino

8. Click **Insert Cell Below**. Also note that the name of your dataframe is now: df_data_1. You will need this name (and yes it will change to df_data_2 and basically n+1 each time you run the import) to insert within some of the code snippets below.
9. Install and import Brunel. Enter the following command and click Run (▶). Allow for the star in the cell mark to turn into a number. This could take upwards of three to five minutes.

```
!pip install brunel==2.3
import brunel
```


Section 4. Visualize the data

You'll create some charts and diagrams with Brunel commands.

The basic format of each call to Brunel is simple. Whether the command is a single line or a set of lines, the commands are concatenated together, and the result interpreted as one command.

Here are some of the rules for using Brunel that you'll need in this notebook:

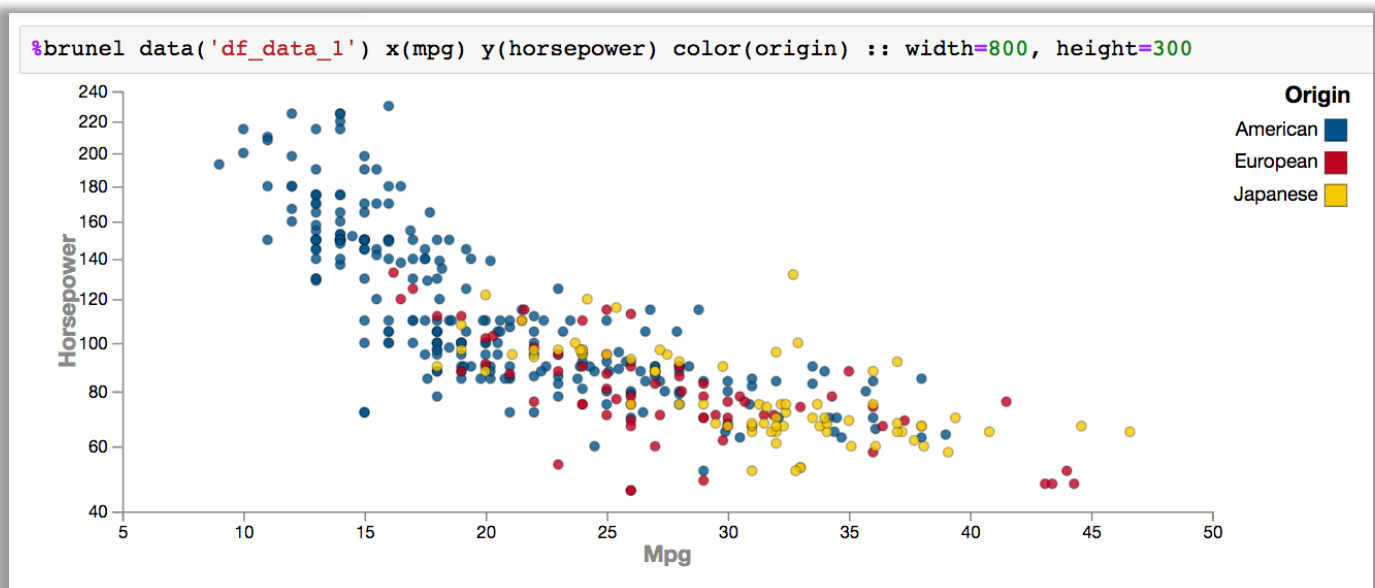
- **DataFrame:** Use the data command to specify the pandas DataFrame.
- **Chart type:** Use commands like chord and treemap to specify a chart type. If you don't specify a type, the default chart type is a scatterplot.
- **Chart definition:** Use the x and y commands to specify the data to include on the x-axis and the y-axis.
- **Styling:** Use commands like color, tooltip, and label to control the styling of the graph.
- **Size:** Use the width and height key-value pairs to specify the size of the graph. The key-value pairs must be preceded with two colons and separated with a comma, for example: `:: width=800, height=300`

See detailed documentation on the Brunel Visualization language at [Introduction to Brunel](#).

Run the cells to show the relationship between the miles per gallon and the horsepower of the vehicles in a scatter plot. The color identifies the origin of the vehicles.

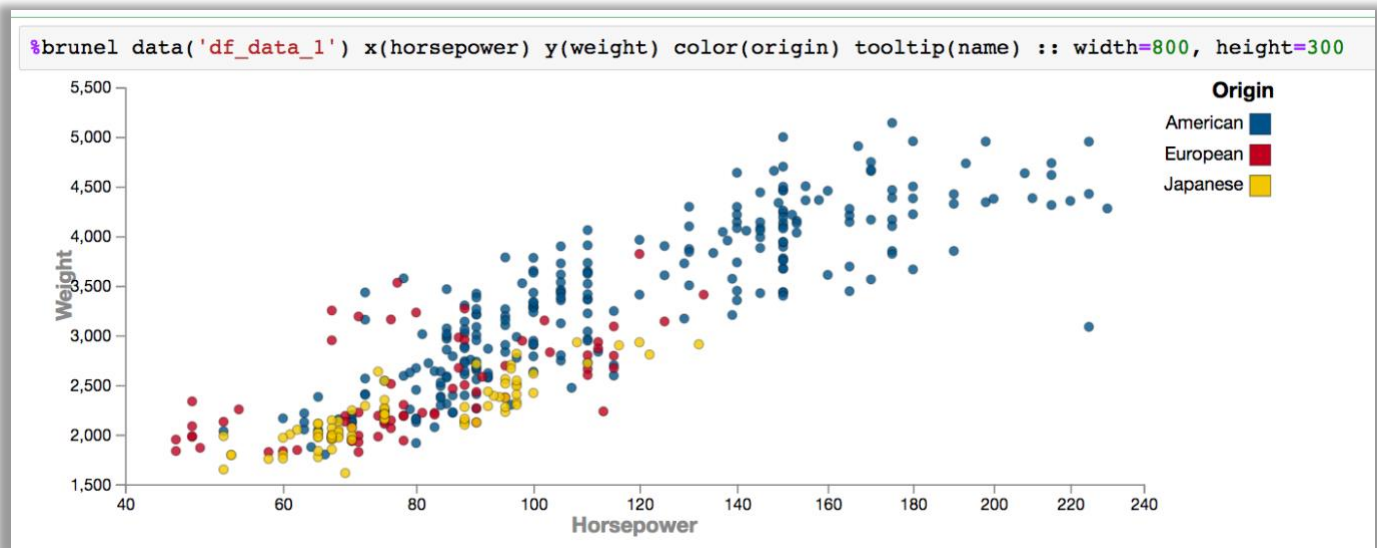
1. In a new cell enter the command below (copy/paste) and run the cell (one line). In this example, your dataframe name is: **df_data_1**

```
%brunel data('<enter the dataframe name here>') x(mpg) y(horsepower) color(origin) :: width=800, height=300
```



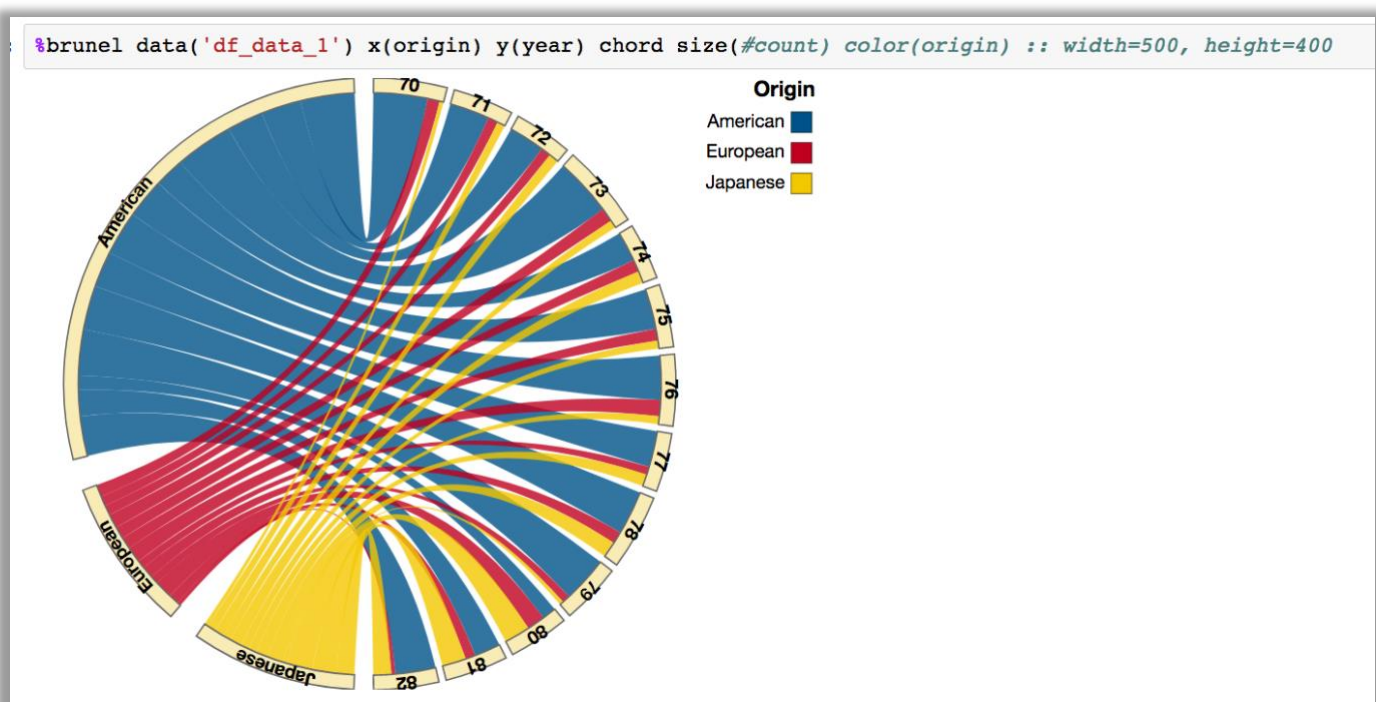
2. In a new cell enter the command below (copy/paste) and run the cell (one line). In this example, your dataframe name is: df_data_1

```
%brunel data('df_data_1') x(horsepower) y(weight) color(origin) tooltip(name) :: width=800, height=300
```



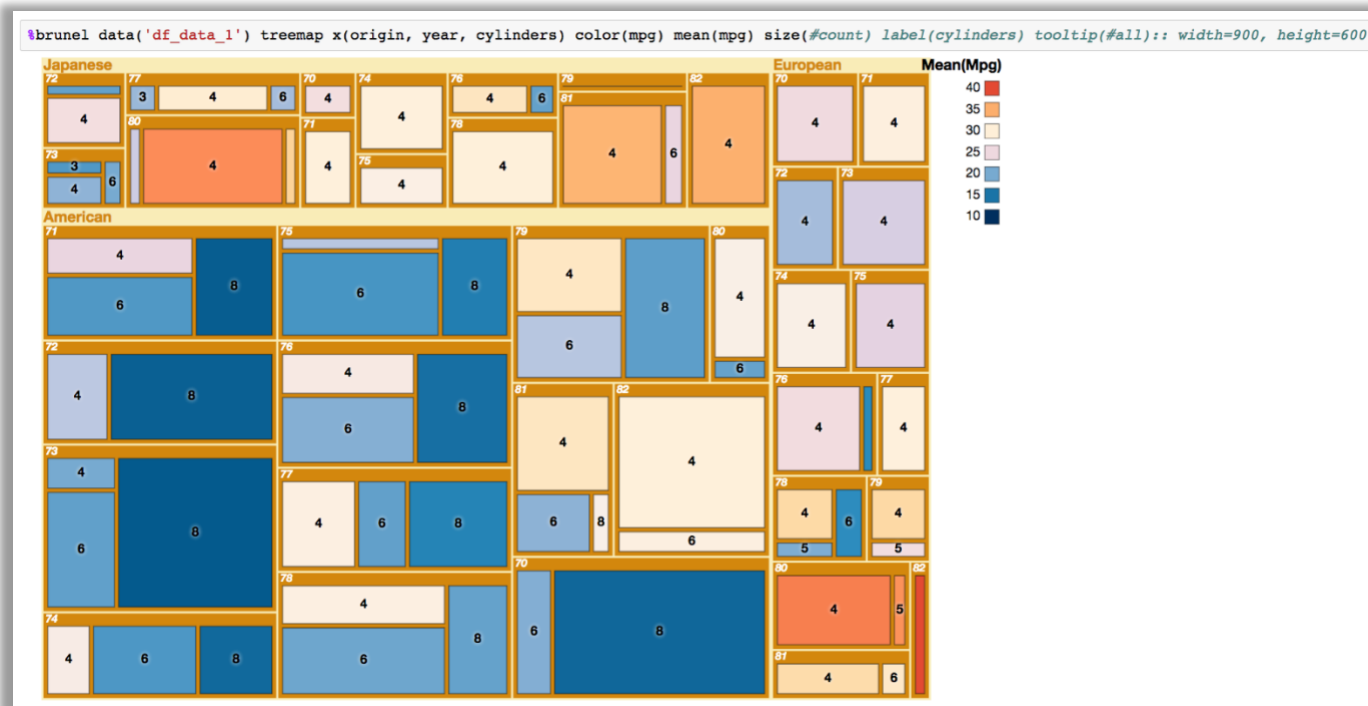
3. In a new cell enter the command below (copy/paste) and run the cell (one line). In this example, your dataframe name is: df_data_1

```
%brunel data('df_data_1') x(origin) y(year) chord size(#count) color(origin) :: width=500, height=400
```



4. In a new cell enter the command below (copy/paste) and run the cell (one line). In this example, your dataframe name is: `df_data_1`

```
%brunel data('df_data_1') treemap x(origin, year, cylinders) color(mpg) mean(mpg) size(#count)
label(cylinders) tooltip(#all):: width=900, height=600
```



You explored different types of charts and formatting and learned how you can use the pandas DataFrame to refine your charts. Try changing the formatting of these charts, or creating your own.



© Copyright IBM Corporation 2019.

The information contained in these materials is provided for informational purposes only and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.



Please Recycle
