Learning Objectives

- **Create a Heat maps and correlograms**
- **Generate Mosaic plot**
- **Determine the best specialized chart to use based on the data provided**

definition

Assumptions

- Learners are comfortable extracting relevant data into data frames, and printing that data to the console.
- Learners are comfortable using ggplot2 to create visualization charts.
  ## Limitations
- This section will cover distribution charts in brief details only and will offer practical visualization functions for learners to start creating charts right away.

Heat Maps

Creating Heat Maps

Follow the directions below to create a heat map!

definition

Heat Maps are 2D graphical data use to show volume of data in data set. The magnitude of the values are represented by colors.

The basic syntax to create a heat map.

```
import matplotlib.pyplot as plt
import numpy as np
plt.imshow(data)
```

Some more documentation regarding the heat map is provided below.

Matplotlib Documentation

Data Import

```
import matplotlib.pyplot as plt
import numpy as np

# Some sector and geographies for illustration
sector=["Technology","Consumer Discretionary", "Real Estate",
        "Financial Services"]

geography=["Asia", "Europe", "US", "India"]



# Simulate market returns based on a random
returns= np.array([[40,13,20,37],
         [30,41,37,46],
         [89,50,65,55],
         [35,20,25,35]])


fig, ax = plt.subplots()
im = ax.imshow(returns)
```

Add on the following code into the text editor and then click the run button to see the result.

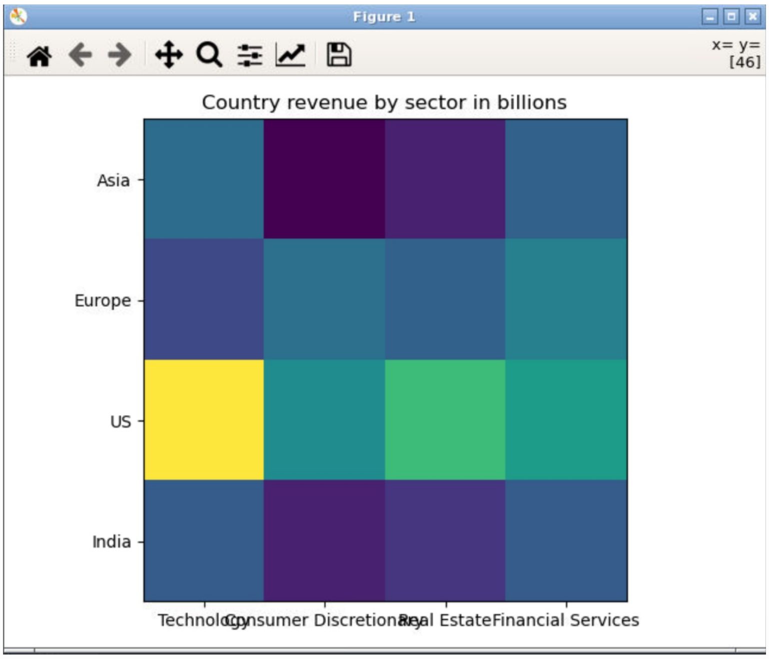
```

# We want to show all ticks...
ax.set_xticks(np.arange(len(sector)))
ax.set_yticks(np.arange(len(geography)))

# ... and label them with the respective list entries
ax.set_xticklabels(sector)
ax.set_yticklabels(geography)


ax.set_title("Country revenue by sector in billions")
fig.tight_layout()
plt.show()
```

Plot Result:



images/heatplot

Correlograms

Creating Correlograms

Follow the directions below to create a correlogram!

We usually use a correlogram to display correlation data over a time period.

Correlogram Documentation

info

The basic syntax is:

```
plt.xcorr(x, y)
```

Data Import

Add the following code into the text editor to see our chart

```
# data Import
# Fixing random state for reproducibility
np.random.seed(2434548631)
x, y = np.random.randn(2, 100)

# creating our subplots
fig, [ax1,ax2]  = plt.subplots(2,1)
#creating our correlogram
ax1.xcorr(x,y, usevlines=True, maxlags=50, normed=True, lw=2)
```

Adding another figure

As you can see we previously used :

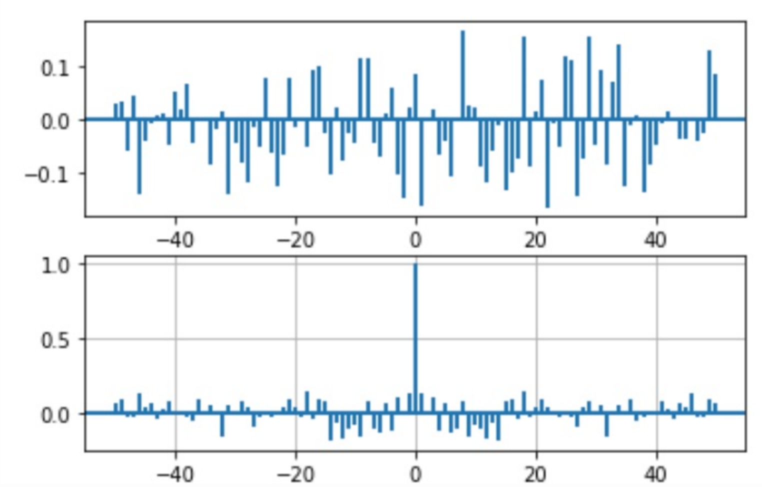
```
fig, [ax1,ax2]  = plt.subplots(2,1)
```

Which is use to create multiple subplots. Lets add another plot to our figure. If you used `plt.show()` previously to check the earlier plot, remove it or else the second subplot will be empty.

```
#adding another subplot

ax2.acorr(x, usevlines=True, normed=True, maxlags=50, lw=2)
ax2.grid(True)
plt.show()
```

Plot Result:



images/Correlogram

Mosaic Plots

Creating Mosaic Plots

Follow the directions below to create a **Mosaic plot.**

> info
>
> **Mosaic Plots** allow you to examine the relationship among two or more categorical variables. For example, country and gender.

Data Import

Let create a simple data where 3 countries (USA, Mexico and Canada) create a council and you trying to create a graphical representation of the councils is split by gender.

Use the code below to create your data frame, and visualize it using a `print` statement.

```
myDataframe = pd.DataFrame({'Country' : ['USA', 'Canada', 'USA',
        'Mexico', 'USA', 'Canada'], 'Gender' : ['Male',
        'Female', 'Male', 'Female', 'Female', 'Female']})
print (myDataframe)
```

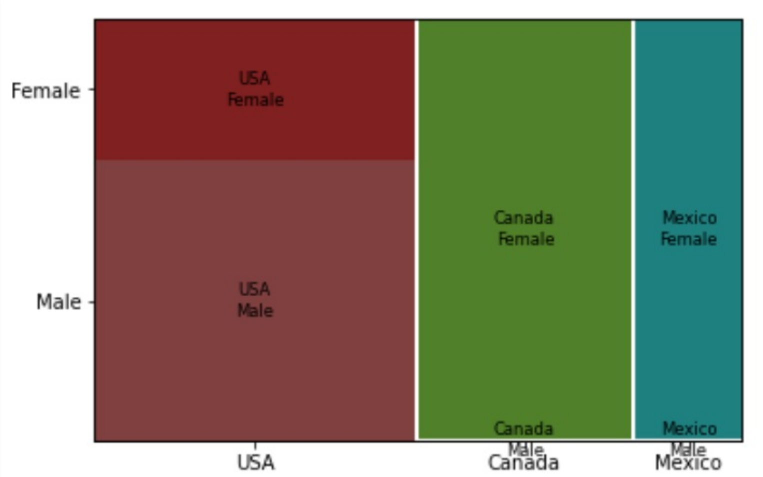
After having our data we need to import our library :

```
from statsmodels.graphics.mosaicplot import mosaic #library for
        mosaic
```

We can try the following, in order to generate our mosaic plot:

```
mosaic(myDataframe, ['Country', 'Gender'])
```

Plot Result:



images/mosaic