Plotting Data 1: Univariate Plots

This lesson will describe the process of plotting univariate plots of the data from a csv file in Python.

WE'LL COVER THE FOLLOWING

Plotting

A **Plot** is a way of representing data visually. It can describe relationships between variables. Plots can give insight and information about the data that may not have been easily derived by looking at the data as a table.

Almost every tool that works with spreadsheets can plot data. Python has a complete package dedicated for plotting data known as *matplotlib*. We will be using this package with pandas to plot data. In this lesson, we will discuss **univariate plots**. These are plots where a single variable is plotted.

Let's look at an example of this using the Sample Sales Data. The data is in *sales_data_sample.csv* file. Have a look at the data.

sales_data.csv

```
import pandas as pd
# Read data
df = pd.read_csv('sales_data.csv')

# Print Information
print(df.info())
```

Plotting individual columns

We can easily plot distributions of a single individual column for our dataframe. Let's look at an example below. Run the code below and click on the output graph to view it in full screen.

Histograms

Histograms are plots that inform us of the underlying *frequency distribution* of the data. They group numbers into ranges and tells us how many values of a variable lie in every range.

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('sales_data.csv')

# plot YEAR ID
df['QUANTITYORDERED'].plot(kind = 'hist',bins = [0,5,15,25,35,45,55,65],rwidth = 0.8)

# display the plot
plt.show()
```

We import `matplotlib.pyplot` package in **line 2** as `plt`. We would use the keyword `plt` wherever we want to use it in the code.

We have plotted the histogram of `QUANTITYORDERED` in **line 7**. We have set `kind` to `hist`. We can provide two more inputs with `hist` to the function; `bins` to specify our ranges and `rwidth` to specify the space between each bar in the graph. If we do not provide these inputs, the function automatically decides some values for the `bins` and `rwidth`. Alternatively, we can provide the number of bins to this function instead of a list and it makes the provided number of bins automatically.

From the graph, we can see that the firm received more than 800 orders in which the quantity ordered was in the range of $25 - 35$.

Density plots

A **density plot** is a representation of the distribution of a numeric variable. It

uses a density estimate to show the probability density function of the

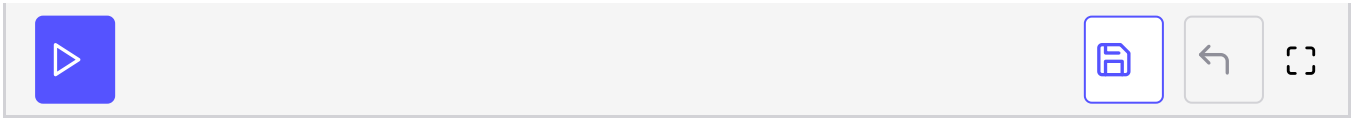
variable. It is a smoothed version of the histogram and is used in the same concept.

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('sales_data.csv')

# plot YEAR ID
df['MONTH_ID'].plot(kind = 'density')

# display the plot
plt.show()
```



In **line** 7 we plot the data in three simple steps:

- Select the column `MONTH_ID` by writing `df['MONTH_ID']`
- Use the `plot` function by writing `.plot()`
- Tell the function the type of graph we want as `kind = 'density'`

We use `plt.show()` to display the graph in **line 10**. From the density plot, we can see a continuous curve that explains the frequency distribution of `MONTH_ID`. It tells us how much data we have in each month. We see a peak at months 10 and 11, which tells us that most orders were made in these months.

Plotting multiple columns

We can plot distributions for all of the columns that have numerical values in a single line with pandas and matplotlib. Let's see an example of this.

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('sales_data.csv')
# plot all numerical columns
df.plot(kind = 'density',subplots = True, layout = (3,3), sharex = False, sharey = False)
# display the plot
plt.show()
```

In **line 6**:

- We use the `plot` function on the whole dataframe by writing `df.plot()`
- We specify the type of the plots as `kind= 'density'`
- Since we are using `plot` function on the whole dataframe, we need to provide some other inputs to the function that will tell it how to draw the plots.
    - We pass `subplots = True` which indicates that multiple plots are to be drawn.
    - We pass `sharex = False` and `sharey = False` so that the plots do not share x-axis and y-axis.
    - We pass `layout = (3,3)` so that the plots are arranged in a grid and look visually better. This is an optional argument.

Finally, we display the plots in **line 8**. The plots show us the distribution of variables by density plots.

Similarly, we can plot *box plots* for the dataframe by passing `kind = box` to the `plot` function. Run the below code to see them.

```
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('sales_data.csv')
# plot all numerical columns
df.plot(kind = 'box',subplots = True, layout = (3,3), sharex = False, sharey = False)
# display the plot
plt.show()
```

Box plots give us information about the spread of the data. They tell us about the mean and the quartiles of each variable.

We can now plot univariate plots in Python. In the next lesson, we will learn how to plot *bivariate* plots.