Plotting Data 2: Bivariate Plots

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

WE'LL COVER THE FOLLOWING

- Bivariate plots
  - Bivariate plots with `dataframe.plot`
  - Scatter plot
  - Bar plot with `groupby`
  - Bar plot with two-level `groupby`

Bivariate plots

Bivariate plots are plots that aim to graph the relationship between two variables. These plots can help us establish relationships and patterns in the data. Some common bivariate plots are:

- *scatter plots*
- *bar plots*
- *histograms*

Let's look at some of these plots and how we can plot them using Pandas and matplotlib. We will be using Sample Sales Data. The data is in *sales_data_sample.csv* file. Have a look at the data.

sales_data.csv

Bivariate plots with `dataframe.plot`

In Pandas, we only need to use the `plot` function with the dataframe to plot bivariate plots. The function expects the following arguments:

- `kind`: the type of plot, i.e. scatter, bar etc

- `kind`: the type of plot, i.e., scatter, bar etc.
- `x`: name of the column to place at x-axis.
- `y`: name of the column to place at y-axis.

Let's look at some of these bivariate plots below.

Scatter plot

A **scatter plot** is the simplest kind of plot between two variables. One variable is plotted on the x-axis while the other is plotted on the y-axis. The pattern of the plot reveals a general trend or any correlation present between the two variables.

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

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

# plot scatter plot
df.plot(kind = 'scatter',x = 'PRICEEACH',y = 'QUANTITYORDERED')
#display plot
plt.show()
```

In the code above, we have plotted two quantities, `PRICEEACH` on the x-axis and `QUANTITYORDERED` on the y-axis in **line** 7. The graph shows that the market in which this firm is operating favors products in the price range of $60 - 80$, as that part of the plot is very dense.

Bar plot with `groupby`

A **bar plot** is similar to a *histogram*, but it is a bivariate plot. The length of the bars represents the quantity on the y-axis. Bar plots are usually used with a grouped-by variable that is plotted on the x-axis. Let's look at an example.

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

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

# Group data and take sum
new_df = df.groupby('PRODUCTLINE').sum()

# Draw a bar plot
```

```
# Draw a bar plot
new_df.plot(kind = 'bar',y = 'SALES')
plt.show()
```

In **line** 7, we grouped the data by the variable `PRODUCTLINE` and then added all the sales for them so that we have total sales for each product line. Then we plot in **line 10**. Here we do not need to provide the `x` input as it is already a grouped dataframe and it is implied that the grouping variable, `PRODUCTLINE`, will be on the x-axis. This plot gives us a good comparison of the total sales of each product line.

Bar plot with two-level `groupby`

We know that we can group data by two variables instead of just one which can give us great insights into the data. We can also plot this two-level grouping. For instance, we want to see the total sales for each product line, year wise. Let's plot this below.

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

df = pd.read_csv('sales_data.csv')
grouped_df = df.groupby(['YEAR_ID','PRODUCTLINE']).sum()

# plot scatter plot
grouped_df.unstack().plot(kind = 'bar',y = 'SALES')

#display plot
plt.show()
```

In **line 5** we group data by `YEAR_ID` and `PRODUCTLINE` and take the sum since we want to calculate total sales. In **line 8** we plot it. But before using the `plot` function, we have used the `unstack` function. `unstack`, changes the hierarchical structure of the grouped dataframe. If we had plotted without using `unstack` then the bars for a single year would not have been grouped together as they are in the plot.

Now that we know how to plot and handle CSV data in Python, we will move to the next chapter where we will learn how to clean our data for analysis.

But before that, you can take a quiz in the next lesson to test yourself on the concepts learned in this chapter.