

Case Study - Online Retailer

Example - 1

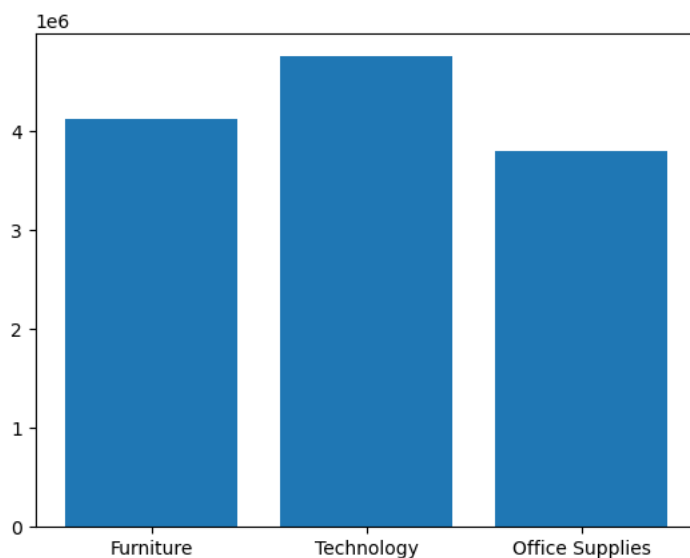
Bar Chart - Plot sales across each product category

- A bar chart uses bars to show comparisons between categories of data.
- A bar graph will always have two axis.
- One axis will generally have numerical values or measures,
- The other will describe the types of categories being compared or dimensions.

```
In [16]: import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

product_category = np.array(['Furniture', 'Technology', 'Office Supplies'])
sales = np.array ([4110451.90, 4744557.50, 3787492.52] )

plt.figure()
plt.bar(product_category,sales)
plt.show()
# your code here
```



Bar Chart - Plot sales across each product category

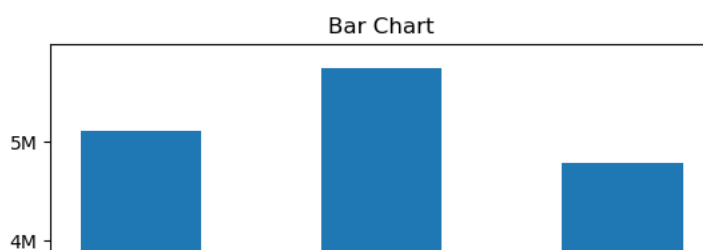
1. Adding labels to Axes
2. Reducing the bar width
3. Giving Title to the chart
4. Modifying the ticks to show information in (million dollars)

```
In [17]: import numpy as np
import matplotlib.pyplot as plt

product_category = np.array(['Furniture', 'Technology', 'Office Supplies'])
sales = np.array ([4110451.90, 4744557.50, 3787492.52] )

# plotting bar chart and setting bar width to 0.5 and aligning it to center
plt.figure()
plt.bar(product_category,sales,width=0.5,align="center")

# Adding and formatting title
plt.title("Bar Chart")
# Labeling Axes
plt.xlabel("Product Category")
plt.ylabel("Sales")
# Modifying the ticks to show information in (million dollars)
plt.yticks(np.arange(0,5000000,1000000),["1M","2M","3M","4M","5M"])
plt.show()
```



```

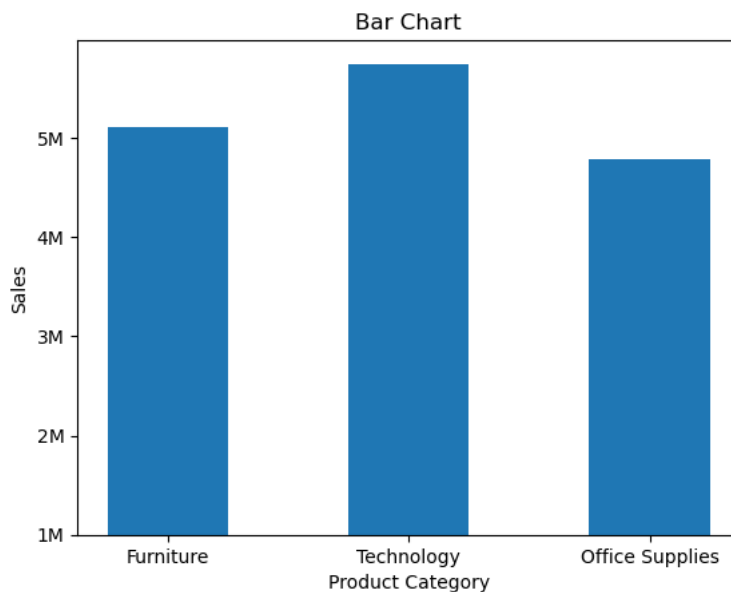
4. Modifying the ticks to show information in (million dollars)
In [17]: import numpy as np
import matplotlib.pyplot as plt

product_category = np.array(['Furniture', 'Technology', 'Office Supplies'])
sales = np.array ([4110451.90, 4744557.50, 3787492.52] )

# plotting bar chart and setting bar width to 0.5 and aligning it to center
plt.figure()
plt.bar(product_category,sales,width=0.5,align="center")

# Adding and formatting title
plt.title("Bar Chart")
# Labeling Axes
plt.xlabel("Product Category")
plt.ylabel("Sales")
# Modifying the ticks to show information in (million dollars)
plt.yticks(np.arange(0,5000000,1000000),["1M", "2M", "3M", "4M", "5M"])
plt.show()

```



Example - 2

Scatter Chart - Plot Sales versus Profits across various Countries and Product Categories

Scatter plots are used when you want to show the relationship between two facts or measures.

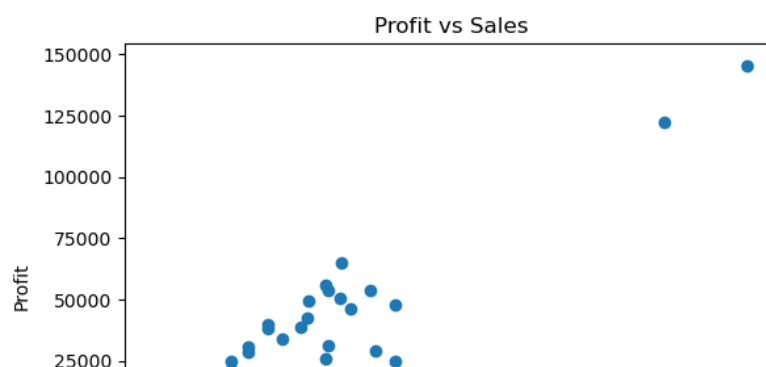
- Profit is measured across each product category

```

In [18]:
In [19]: # plotting scatter chart
plt.figure()
plt.scatter(sales,profit)
plt.title("Profit vs Sales")

# Labeling Axes
plt.xlabel("Sales")
plt.ylabel("Profit")
plt.show()

```



The scatter plot, titled "Profit vs Sales", displays the relationship between Sales (X-axis) and Profit (Y-axis). The X-axis ranges from 0 to 800,000, and the Y-axis ranges from -25,000 to 150,000. The data points are represented by blue dots. There is a dense cluster of points at low sales (below 200,000) and low profit (between -25,000 and 25,000). A few points show higher profit at low sales, reaching up to 75,000. As sales increase, profit generally increases, with a notable outlier at approximately (700,000, 120,000) and another at (850,000, 145,000). There are also a few points with negative profit at higher sales, around 300,000.

- Represent product category using different colors
- Adding a Legend to Product Categories

Scatter Plot of Sales vs Profit by Product Category

```
In [27]: # Create scatter plot for Technology category (green)
plt.scatter([sales[i] for i in range(len(sales)) if product_category[i] == "Technology"],
            [profit[i] for i in range(len(profit)) if product_category[i] == "Technology"],
            label="Technology", color="green", s=100)

# Create scatter plot for Office Supplies category (yellow)
plt.scatter([sales[i] for i in range(len(sales)) if product_category[i] == "Office Supplies"],
            [profit[i] for i in range(len(profit)) if product_category[i] == "Office Supplies"],
            label="Office Supplies", color="yellow", s=100)

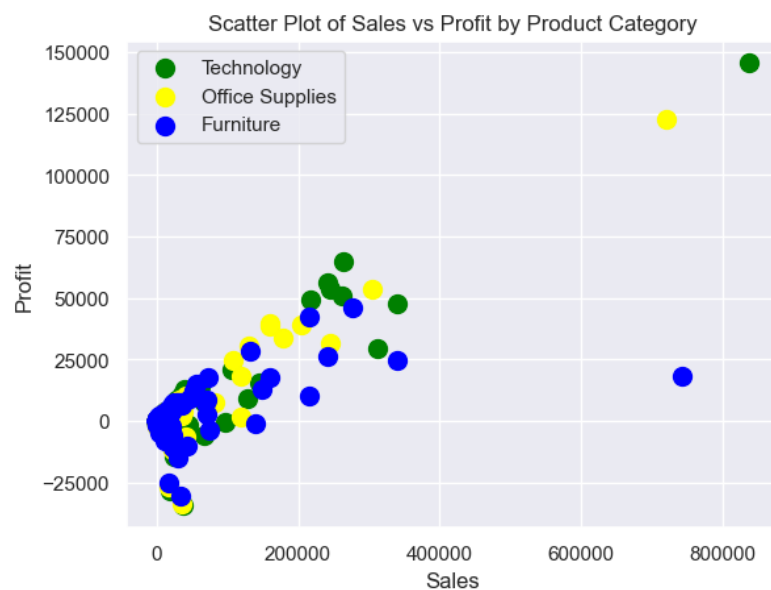
# Create scatter plot for Furniture category (blue)
plt.scatter([sales[i] for i in range(len(sales)) if product_category[i] == "Furniture"],
            [profit[i] for i in range(len(profit)) if product_category[i] == "Furniture"],
            label="Furniture", color="blue", s=100)

# Adding and formatting title
plt.title("Scatter Plot of Sales vs Profit by Product Category")

# Labeling Axes
plt.xlabel("Sales")
plt.ylabel("Profit")

# Adding Legend
plt.legend()

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



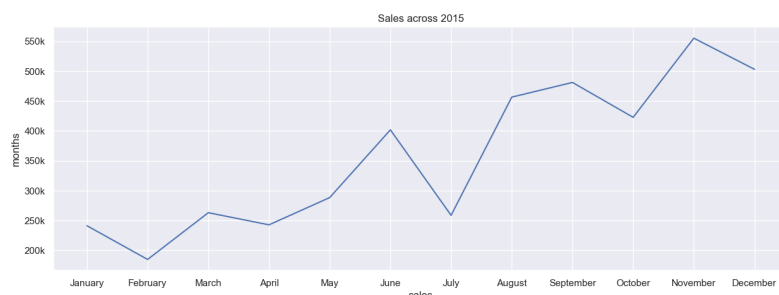
Example - 3

Line Chart - Plot Sales across 2015

```
In [35]: # Line plot months Vs sales
plt.figure(figsize=(15,5))
plt.plot(months,sales)

# Adding and formatting title
plt.title("Sales across 2015")

In [28]: import matplotlib.pyplot as plt
# Labeling Axes
plt.xlabel("months")
plt.ylabel("sales")
plt.xticks(months)
plt.yticks([200000, 250000, 300000, 350000, 400000, 450000, 500000])
plt.show()
```



Example - 4

```

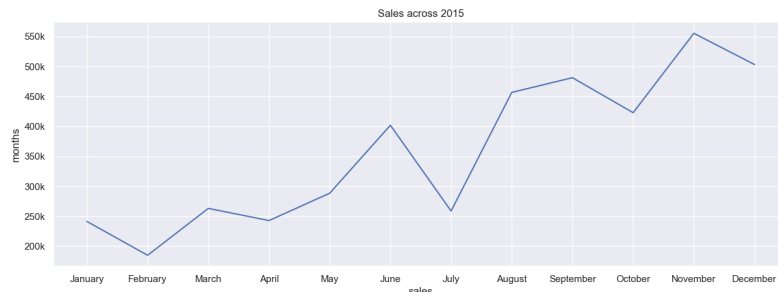
In [35]: # Line plot months Vs sales
plt.figure(figsize=(15,5))
plt.plot(months,sales)

# Adding and formatting title
plt.title("Sales across 2015")

In [28]: import numpy as np
import matplotlib.pyplot as plt
# Labeling Axes
plt.xlabel("months")
plt.ylabel("sales")
# add x and y ticks in given format
plt.xticks(['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December'])
plt.yticks([20000, 25000, 30000, 35000, 40000, 45000, 50000, 55000])

plt.show()

```



Example - 4

Box and Whisker Chart - Sales across Countries and Product Categories

A Box and Whisker Plot (or Box Plot) is a convenient way of visually displaying the data distribution through their quartiles. The lines extending parallel from the boxes are known as the "whiskers", which are used to indicate variability outside the upper and lower quartiles. Outliers are sometimes plotted as individual dots that are in-line with whiskers. Box Plots can be drawn either vertically or horizontally.

```

In [ ]: import numpy as np
import matplotlib.pyplot as plt

# Data
sales_technology = np.array ([1013.14, 8298.48, 875.51, 22320.83, 9251.6, 4516.72, 1770.13, 7527.18, 1433.65, 423.3, 21601.72, 981.84, 10209.84, 156.56, 243.06, 21287.52, 7300.46])
sales_office_supplies = np.array ([1770.13, 7527.18, 1433.65, 423.3, 21601.72, 981.84, 10209.84, 156.56, 243.06, 21287.52, 7300.46])
sales_furniture = np.array ([981.84, 10209.84, 156.56, 243.06, 21287.52, 7300.46])

```

```

In [ ]: # plot boxplot for sales_technology, sales_office_supplies, sales_furniture

# Adding and formatting title

# Labeling Axes
#add x ticks

```

```

In [36]: import numpy as np
import matplotlib.pyplot as plt

# Data
profit = np.array ([-5428.79, 7001.73, -3706.46, 300.42, -1697.31, -11222.71, 100000])

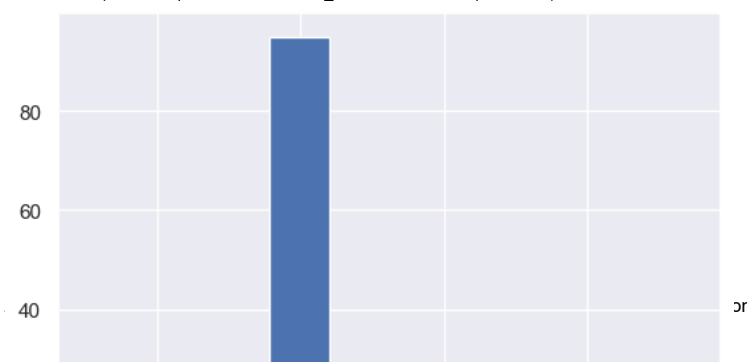
plt.show()

```

```

In [37]: # plot hist plot for profit
plt.figure()
plt.hist(profit)
plt.show()

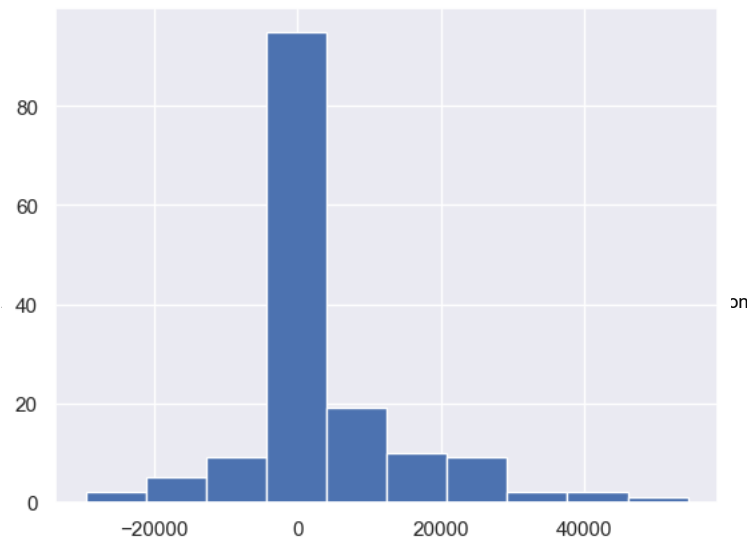
```



```
In [36]: import numpy as np
import matplotlib.pyplot as plt

# Data
profit = np.array([-5428.79, 7001.73, -3706.46, 300.42, -1697.31, -11222.71, 100000])

In [37]: # plot hist plot for profit
plt.figure()
plt.hist(profit)
plt.show()
```



Example - 6

SubPlots - Plot Sales for various markets for years 2012 to 2015

```
In [38]: import numpy as np
import matplotlib.pyplot as plt

years = np.array(['2012', '2013', '2014', '2015'])

sales_africa = np.array([127187.27, 144480.70, 229068.79, 283036.44])

sales_USCA = np.array([492756.60, 486629.30, 627634.98, 757108.13])

sales_LATAM = np.array([385098.15, 464733.29, 608140.77, 706632.93])

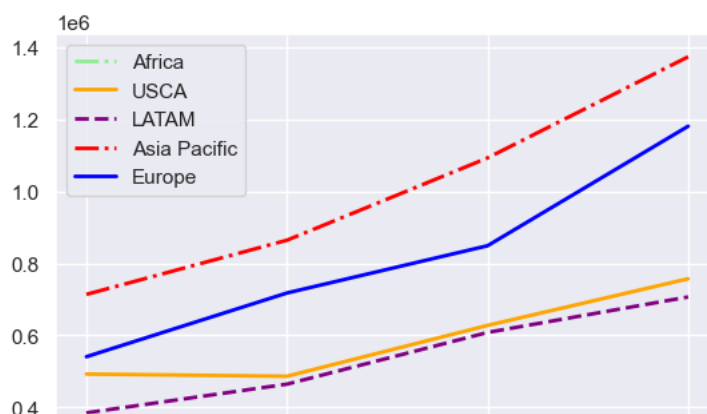
sales_Asia_Pacific = np.array([713658.22, 863983.97, 1092231.65, 1372784.40])

sales_Europe = np.array([540750.63, 717611.40, 848670.24, 1180303.95])
```

```
In [43]: # Subplots are shown in the same graph as line charts, identified by different colours
plt.plot(years, sales_africa, linewidth=2, color="lightgreen", linestyle="-.", label="Africa")
plt.plot(years, sales_USCA, linewidth=2, color="orange", linestyle="--", label="USCA")
plt.plot(years, sales_LATAM, linewidth=2, color="purple", linestyle="--", label="LATAM")
plt.plot(years, sales_Asia_Pacific, linewidth=2, color="red", linestyle="dashdot", label="Asia Pacific")
plt.plot(years, sales_Europe, linewidth=2, color="blue", label="Europe")

plt.legend(["Africa", "USCA", "LATAM", "Asia Pacific", "Europe"])
```

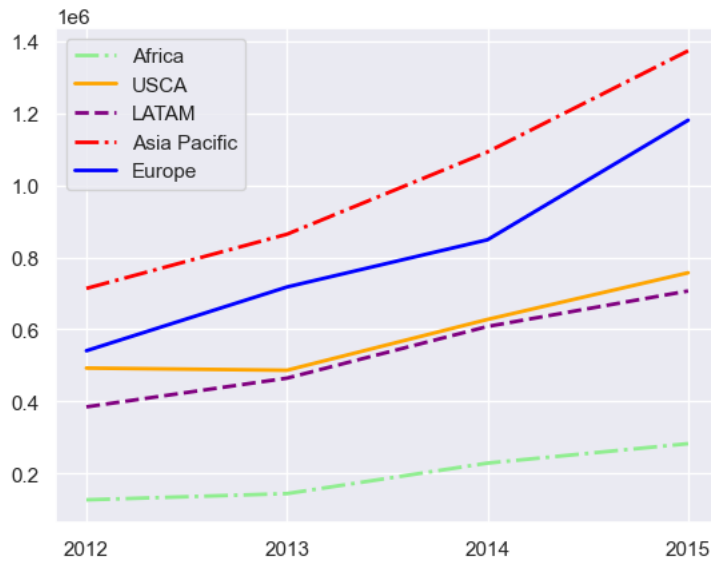
Out[43]: <matplotlib.legend.Legend at 0x22789bf4580>



```
In [43]: Subplots are shown in the same graph as line charts, identified by different colours
plt.plot(years,sales_africa,linewidth=2,color="lightgreen",linestyle="--",label="Africa")
plt.plot(years,sales_USCA,linewidth=2,color="orange",linestyle="--",label="USCA")
plt.plot(years,sales_LATAM,linewidth=2,color="purple",linestyle="--",label="LATAM")
plt.plot(years,sales_Asia_Pacific,linewidth=2,color="red",linestyle="dashdot",label="Asia Pacific")
plt.plot(years,sales_Europe,linewidth=2,color="blue",label="Europe")

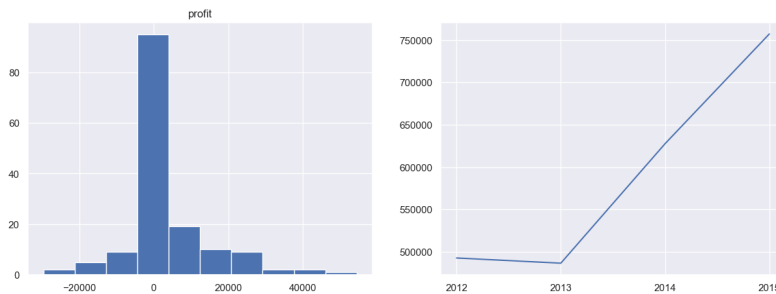
plt.legend(["Africa","USCA","LATAM","Asia Pacific","Europe"])
```

Out[43]: <matplotlib.legend.Legend at 0x22789bf4580>

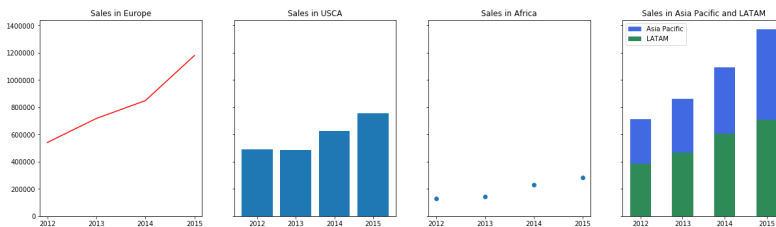


Subplots are shown in separate graphs. Each chart can be of different types.

```
In [44]: plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
plt.hist(profit)
plt.title("profit")
plt.subplot(1,2,2)
plt.plot(years,sales_USCA)
plt.show()
```



In []:



In []: