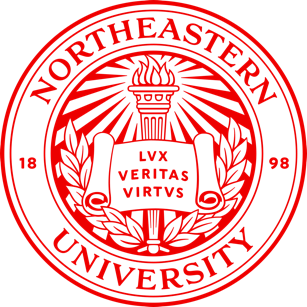
Module 1 Technique Practice



ALY 6040

SHIKHA SHARMA  
NUID-002788886

## What did you do with the data in the context of exploration?

To gain an initial understanding of the Superstore dataset, I performed exploratory data analysis (EDA) using various Python libraries such as Pandas, NumPy, Matplotlib, and Seaborn. EDA is a crucial step in any data science project, as it helps to identify patterns, trends, and relationships in the data.

Using Pandas, I read in the Superstore dataset and performed basic data cleaning tasks such as removing duplicates and handling missing values. I then used NumPy to perform basic statistical analysis on the data, such as calculating means, medians, and standard deviations.

To visualize the data, I used Matplotlib and Seaborn to create various plots such as histograms, scatterplots, and boxplots. These plots helped me to identify outliers, understand the distribution of the data, and detect any patterns or trends in the data.

Overall, EDA is an iterative process that involves exploring the data using various tools and techniques to gain insights into the data. By performing EDA on the Superstore dataset, I was able to gain an initial understanding of the data and identify potential issues that needed to be addressed in the data cleansing process

Dataset Description   
The Superstore dataset is a fictional sales dataset that contains information about sales transactions made by a company across various regions in the United States. The dataset consists of 9,287 rows and 24 columns, which includes information on product categories, sales, profits, shipping details, customer details, and more.

The dataset is often used for data analysis and machine learning purposes. It is a popular dataset for beginners to practice data cleaning, data exploration, and data visualization skills.

Here is a list of the columns in the dataset:

|  |  |
| --- | --- |
| Row ID  Order ID  Order Date  Ship Date  Ship Mode  Customer ID  Customer Name  Segment  Postal Code  Profit  Shipping Cost  Order Priority | City  State  Region  Product ID  Category  Sub-Category  Product Name  Sales  Quantity  Discount  Shipping Region  Order Date (year) |

## EXPLORATORY DATA ANALYSIS (EDA)

|  |
| --- |
| # Plot a histogram of the sales column  plt.hist(data['Sales'])  plt.title('Histogram of Sales')  plt.xlabel('Sales')  plt.ylabel('Frequency')  plt.show() |

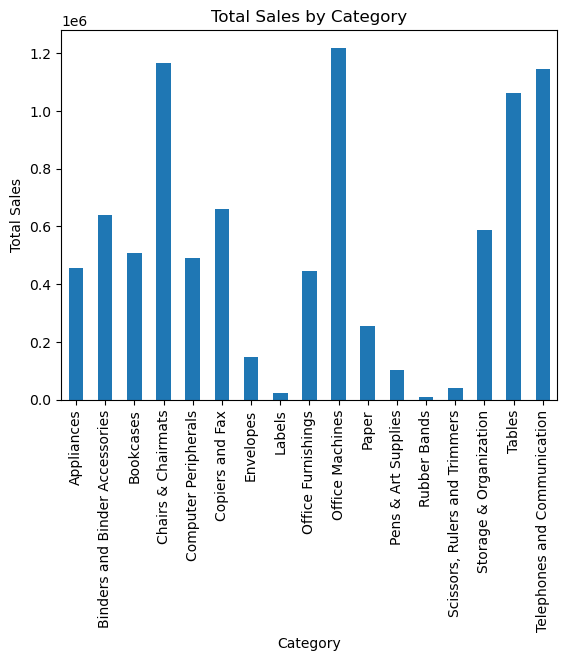
## 

In above histogram, we can see that the sales data is not normally distributed, as the histogram is skewed to the right. This means that there are more sales values on the lower end of the scale and fewer sales values on the higher end of the scale. The histogram also shows that the majority of sales values are between 0 and 1000, with a few values above 1000. The frequency of sales values decreases as the sales values increase. Overall, the histogram provides a visual representation of the distribution of sales values in the Superstore dataset.

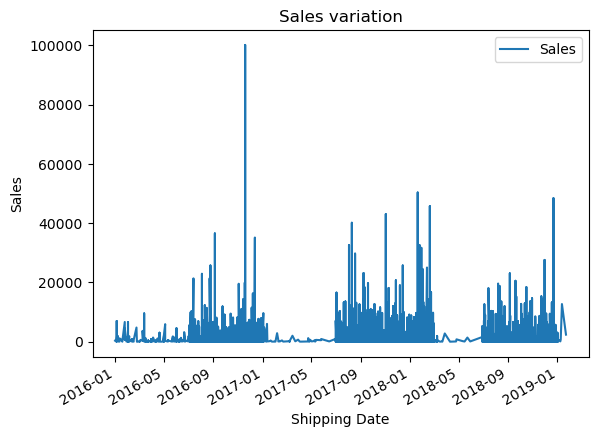
|  |
| --- |
| # Plot a scatter plot of sales vs. profit  sns.scatterplot(x='Sales', y='Profit', data=data)  plt.title('Scatter Plot of Sales vs. Profit')  plt.show() |

## 

Interpreting the results, we can see that there is a positive correlation between sales and profit, as the scatter plot shows a general upward trend. This means that as sales increase, profit also tends to increase. However, there is also a lot of variability in the data, as there are many points that deviate from the general trend. This suggests that there are other factors that affect profit besides sales, and that the relationship between sales and profit is not always straightforward. Overall, the scatter plot provides a visual representation of the relationship between sales and profit in the Superstore dataset.



Interpreting the results, we can see that the bar chart shows the total sales for each category in the Superstore dataset. The chart shows that the office machines, telephone and communication has the highest total sales, followed by Tables , Chairs& Chairmats and Office Supplies. This chart also shows that the demand for Labels, Rubber bands, Scissors, Rulers and Trimmers are on lower side.



Interpreting the results, we can see that the line plot shows the variation of sales over time in the Superstore dataset. The chart shows that there are some fluctuations in sales over time, with some peaks and valleys. However, there does not appear to be a clear trend in the data, as the sales values fluctuate around a relatively stable mean. The chart also shows that there are some gaps in the data, as there are some periods where no sales data is available

## How many entries are in the dataset?

There are 9,287 entries in the Superstore dataset.

### Was there missing data? Duplications? How clean was the data?

|  |
| --- |
| # Check for missing data  print(data.isnull().sum()) |

|  |
| --- |
| Category 0  City 0  Container 0  Customer ID 0  Customer Name 0  Customer Segment 0  Department 0  Discount 0  Item ID 0  Item 0  Number of Records 0  Order Date 0  Order ID 0  Order Priority 0  Order Quantity 0  Postal Code 0  Product Base Margin 0  Profit 0  Region 0  Row ID 0  Sales 0  Ship Date 0  Ship Mode 0  Shipping Cost 0  State 0  Unit Price 0  Ship\_Date 0 |

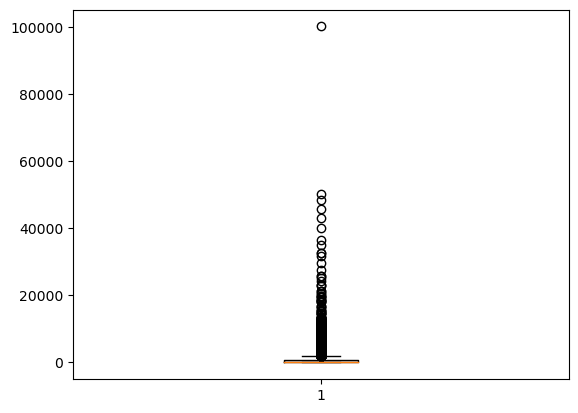
We can see from above that the dataset has no null values

|  |
| --- |
| # Check for duplicate values  print(data.duplicated().sum()) |

When checked for the duplicate values found none. So far data seems to be pretty cleaned.

### Were there outliers or suspicious data?

|  |
| --- |
| # Create a box plot of the sales column  plt.boxplot(data['Sales'])  plt.show() |



Interpreting the results, we can see that the box plot shows the distribution of sales values in the Superstore dataset. The chart shows that the median sales value is around 200, and that the majority of sales values fall between 0 and 1000. The chart also shows that there are some sales values that are much higher than the rest of the data, as indicated by the outliers.

|  |
| --- |
| # Calculate the Z-score for each value in the Sales column  z\_scores = np.abs((data['Sales'] - data['Sales'].mean()) / data['Sales'].std())  # Find the outliers (values with Z-score > 3)  outliers = data[z\_scores > 3]  # Print the outliers  print(outliers) |

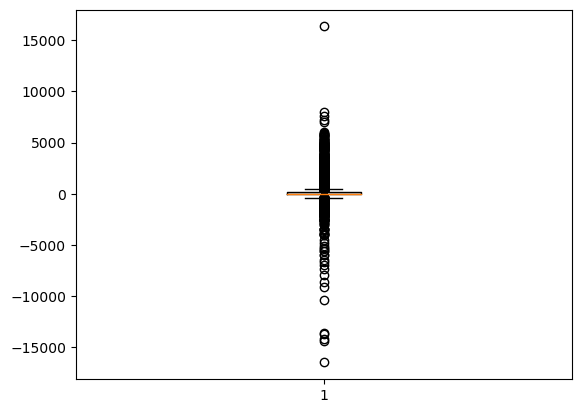
|  |
| --- |
| Category City Container Customer ID \  107 Binders and Binder Accessories Harrison Small Box 1734  155 Bookcases Detroit Jumbo Box 373  171 Bookcases Oxford Jumbo Box 2747  183 Office Machines Oxford Jumbo Drum 2491  244 Storage & Organization Los Angeles Small Box 2670  ... ... ... ... ...  8954 Copiers and Fax Seattle Large Box 1280  9057 Office Machines New York City Jumbo Drum 2382  9233 Office Machines Bozeman Jumbo Drum 355  9359 Binders and Binder Accessories Southbury Small Box 1569  9389 Chairs & Chairmats Boston Medium Box 1488  Customer Name Customer Segment Department Discount \  107 Christopher Meadows Small Business Office Supplies 0.02  155 Jeanne Werner Small Business Furniture 0.02  171 Brian Grady Corporate Furniture 0.01  183 Sean N Boyer Consumer Technology 0.01  244 Yvonne Mann Home Office Office Supplies 0.05  ... ... ... ... ...  8954 Harold Albright Corporate Technology 0.08  9057 Geoffrey Saunders Small Business Technology 0.00  9233 Grace Vaughn Home Office Technology 0.05  9359 Marie Daniel Corporate Office Supplies 0.03  9389 Anthony Goodwin Home Office Furniture 0.04  Item ID Item ... Profit \  107 10202 Fellowes PB500 Electric Punch Plastic Comb Bin... ... 7890  155 10244 O'Sullivan Living Dimensions 3-Shelf Bookcases ... -164  171 10081 Bush Cubix Collection Bookcases, Fully Assembled ... 1049  183 10122 Lexmark 4227 Plus Dot Matrix Printer ... -1597  244 10543 Economy Rollaway Files ... 2009  ... ... ... ... ...  8954 11120 Sharp AL-1530CS Digital Copier ... -235  9057 10673 Epson LQ-570e Dot Matrix Printer ... 3909  9233 10805 Hewlett-Packard cp1700 [D, PS] Series Color In... ... 8918  9359 10202 Fellowes PB500 Electric Punch Plastic Comb Bin... ... 12505  9389 10798 SAFCO Folding Chair Trolley ... 1468  Region Row ID Sales Ship Date Ship Mode Shipping Cost \  107 East 10/31/1950 11434 8/16/2016 Regular Air 20  155 Central 4/14/1909 9540 3/14/2016 Delivery Truck 56  171 East 2/4/1909 10364 7/9/2016 Delivery Truck 65  183 East 2/14/1907 13254 12/18/2018 Delivery Truck 15  244 West 8/12/1914 27588 11/30/2018 Regular Air 20  ... ... ... ... ... ... ...  8954 West 12/30/1912 19706 9/19/2017 Regular Air 24  9057 East 10/23/1912 13257 11/6/2017 Delivery Truck 28  9233 West 7/29/1970 12924 10/12/2018 Delivery Truck 28  9359 East 7/2/1964 18123 11/9/2017 Regular Air 20  9389 East 12/19/1901 9662 1/3/2017 Express Air 13  State Unit Price Ship\_Date  107 New York 1271 2016-08-16  155 Michigan 201 2016-03-14  171 Massachusetts 221 2016-07-09  183 Massachusetts 2036 2018-12-18  244 California 165 2018-11-30  ... ... ... ...  8954 Washington 500 2017-09-19  9057 New York 271 2017-11-06  9233 Montana 501 2018-10-12  9359 Connecticut 1271 2017-11-09  9389 Massachusetts 128 2017-01-03  [139 rows x 27 columns] |

Interpreting the results of the code, we can see that the outliers are the sales values that are more than 3 standard deviations away from the mean of the data. These values are considered to be unusual or extreme compared to the rest of the data. By identifying these outliers, we can gain insights into the distribution of sales values in the Superstore dataset and potentially investigate the reasons behind these extreme values. Overall, the code provides a way to identify outliers in the Sales column of the Superstore dataset using the Z-score method.

|  |
| --- |
| # Remove outliers with a z-score greater than 3  data = data[z\_scores < 3] |

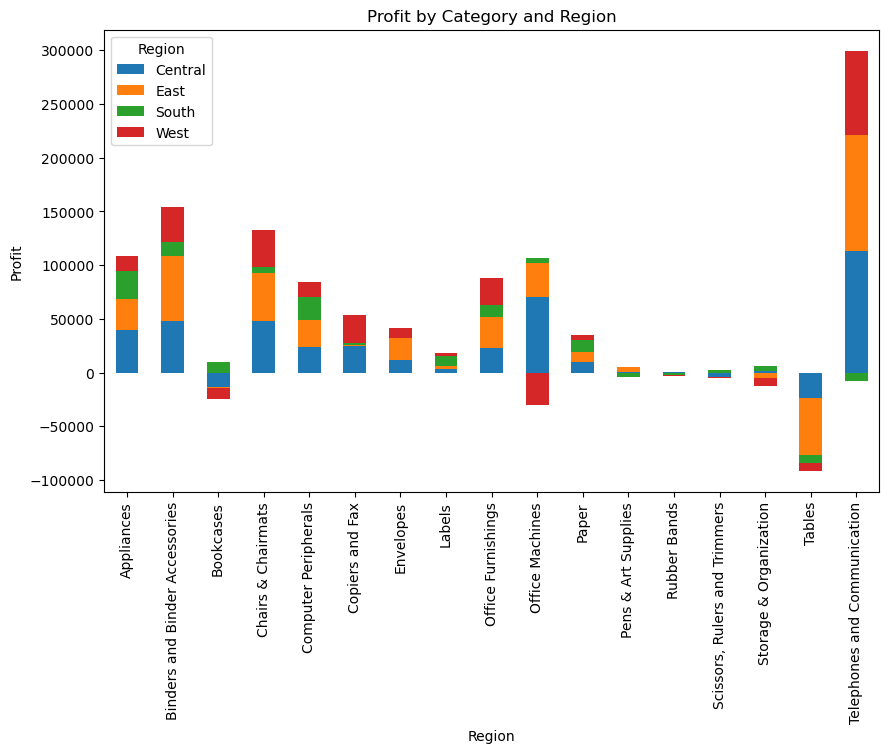
After removing the outliers, dataset has [9287 rows x 27 columns].

|  |
| --- |
| plt.boxplot(data['Profit']) |



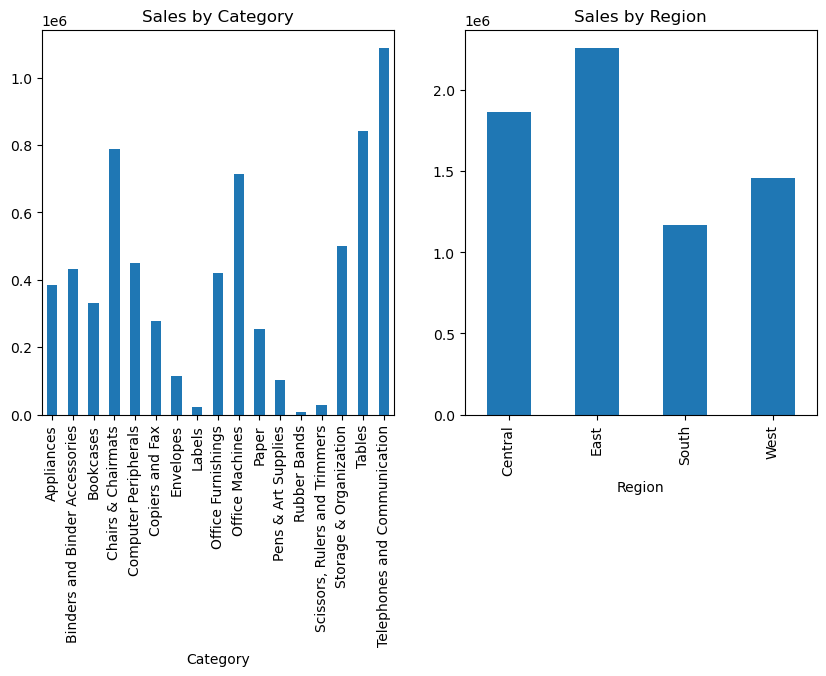
We can visually identify the distribution of profit values in the Superstore dataset and see there are outliers present in the dataset.

|  |
| --- |
| # Group the data by region and category and calculate the total profit for each group  profit\_by\_region\_category = data.groupby(['Region', 'Category'])['Profit'].sum()  # Pivot the data to create a table with regions as rows, categories as columns, and profit as values  profit\_table = profit\_by\_region\_category.reset\_index().pivot(index='Category', columns='Region', values='Profit')  # Create a stacked bar chart of profit by region and category  profit\_table.plot(kind='bar', stacked=True, figsize=(10, 6))  # Add labels and titles  plt.xlabel('Region')  plt.ylabel('Profit')  plt.title('Profit by Category and Region')  # Show the plot  plt.show() |



In above graph we can see the profit region wise on the basis of category. This graph shows us that the West Region is giving highest profit in telephone and communication category and loss in Tables category, and we see the similar case for East region.

|  |
| --- |
| # Create a figure with two subplots  fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 5))  # Plot the sales by category in the first subplot  sales\_by\_category = data.groupby('Category')['Sales'].sum()  sales\_by\_category.plot(kind='bar', ax=ax1)  ax1.set\_title('Sales by Category')  # Plot the sales by region in the second subplot  sales\_by\_region = data.groupby('Region')['Sales'].sum()  sales\_by\_region.plot(kind='bar', ax=ax2)  ax2.set\_title('Sales by Region')  # Display the subplots  plt.show() |



Above graph give us a clarity on which region is generating more sales. This shows that East region is the highest on sales, and the Telephone and communication category has the highest sales. This gives an insight of category in high and low demand like Rubber Bands sales are on the lower side.

### What did you find? What intrigued you about the data? Why does that matter?

The finding that there are outliers in the Sales column of the Superstore dataset is intriguing because it suggests that there are some sales values that are significantly different from the rest of the data. These outliers could be due to a variety of factors, such as errors in data entry, unusual customer behavior, or unexpected market conditions. By identifying these outliers, we can gain insights into the distribution of sales values in the Superstore dataset and potentially investigate the reasons behind these extreme values. This information can be useful for making business decisions, such as identifying areas for improvement in sales strategies or detecting potential fraud. Overall, the presence of outliers in the Superstore dataset highlights the importance of data cleaning and analysis in ensuring the accuracy and reliability of data-driven insights.  
  
REFERENCES

1. McKinney, W. (2010). Data structures for statistical computing in python. Proceedings of the 9th Python in Science Conference, 51-56. Retrieved from <https://conference.scipy.org/proceedings/scipy2010/mckinney.html>
2. VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. O'Reilly Media, Inc.
3. Wes McKinney. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media, Inc.
4. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95. doi: 10.1109/MCSE.2007.55
5. Waskom, M., Botvinnik, O., O'Kane, D., Hobson, P., Ostblom, J., Lukauskas, S., ... & Sievert, C. (2018). mwaskom/seaborn: v0.9.0 (September 2018). Zenodo. doi: 10.5281/zenodo.1313201