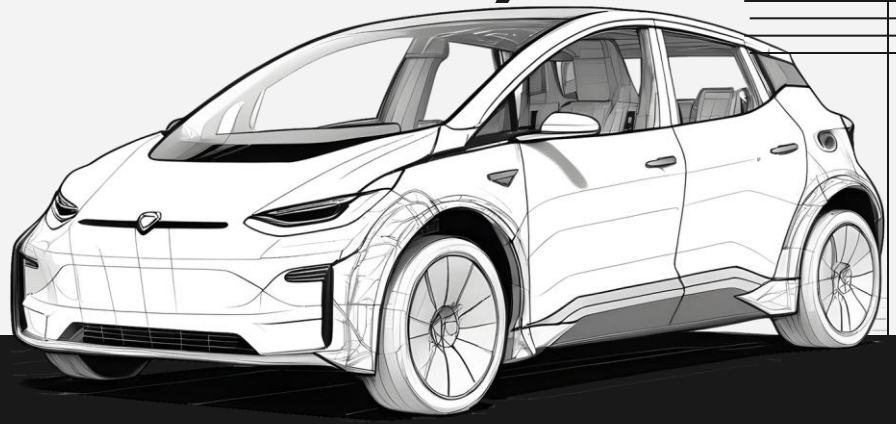
VITYARTHI PROJECT
(CAR MANAGEMENT SYSTEM)

MADE BY- MOHAMMAD.ALI
REG. NUMBER – 25MIB10009



INDEX

PROBLEM STATEMENT

- Organizations and service centers currently struggle with manual methods for managing vehicle and customer data, leading to inefficient workflows, data inaccuracies, and a lack of quick access to crucial information.
- This manual approach results in problems such as lost or damaged records, time-consuming data entry and retrieval, and difficulty in generating comprehensive reports, which increases operational costs and reduces customer satisfaction.

DESIGN

1. Main Menu (Function: main_menu)

- **START**
- Initialize global DataFrame df to None.
- Check if CAR_SALES.csv exists. If yes, call READ CSV FILE (Step 2) automatically.
- **Loop** (Main Menu): a. Display Main Menu options (1. Read CSV File, 2. Data Analysis Menu, 3. Graph Menu, 4. Exit). b. Read user choice. c. **If** choice is '1': Call READ CSV FILE (Step 2). d. **If** choice is '2': i. **If** df is not None: Call DATA ANALYSIS MENU (Step 3). ii. **Else**: Display "Data not loaded." e. **If** choice is '3': i. **If** df is not None: Call GRAPH MENU (Step 4). ii. **Else**: Display "Data not loaded." f. **If** choice is '4': Display "Exiting." **BREAK LOOP**. g. **Else**: Display "Invalid choice." h. Prompt user to "Press ENTER to continue."
- **END**

DESIGN

2. Read CSV File (Function: read_csv_file)

- Read filename from user (defaulting to CAR_SALES.csv).
- Attempt to read the CSV into df using pandas.read_csv, setting 'MODEL' as index_col.
- Convert all sales columns to numeric types.
- **If** successful: a. Display success message. b. Print the entire DataFrame df and its dimensions.
- **If** error (e.g., FileNotFoundError): a. Display error message.
- Return to Main Menu loop.

DESIGN

3. Data Analysis Menu (Function: data_analysis_menu)

- **Loop** (Data Analysis Menu): a. Display Data Analysis Menu options (1 to 8). b. Read user choice. c. **If** choice is '1': Call SHOW COLUMNS (Step 3.1). d. **If** choice is '2': Call SHOW TOP ROWS (Step 3.2). e. **If** choice is '3': Call SHOW BOTTOM ROWS (Step 3.3). f. **If** choice is '4': Call SHOW SPECIFIC COLUMN (Step 3.4). g. **If** choice is '5': Call ADD NEW RECORD (Step 3.5). h. **If** choice is '6': Call DELETE COLUMN (Step 3.6). i. **If** choice is '7': Call DATA SUMMARY (Step 3.7). j. **If** choice is '8': **BREAK LOOP**. k. **Else**: Display "Invalid choice." l. Prompt user to "Press ENTER to continue.“
- Return to Main Menu loop.

DESIGN

3.4. SHOW SPECIFIC COLUMN

- Read column_name from user.
- **If** column_name exists in df.columns: Print df[column_name].
- **Else**: Print "Column not found."

DESIGN

3.5. ADD NEW RECORD

- Read new_model_name.
- **If** model already exists: Print error.
- **Else**: a. For each year (2015-2020): Read sales value from user. b. Create a new pandas Series from the sales data. c. Add the Series as a new column to df with new_model_name. d. Update global CAR_MODELS and MANUFACTURERS lists. e. Print updated df.

3.6. DELETE COLUMN

- Read column_name to delete.
- **If** column_name exists: a. Drop the column using df.drop(columns=[column_name], inplace=True). b. Update global model lists. c. Print updated df.
- **Else**: Print "Column not found."

3.7. DATA SUMMARY

DESIGN

- Call df.describe().transpose().
- Print the resulting summary table.

4. Graph Menu (Function: graph_menu)

- **Loop** (Graph Menu): a. Display Graph Menu options (1. Line Graph, 2. Bar Graph, 3. Exit) and available Manufacturers. b. Read user choice (graph_choice). c. **If** graph_choice is '3': **BREAK LOOP**. d. **If** graph_choice is '1' or '2': i. Set graph_type ('line' or 'bar'). ii. Prompt user to select a Manufacturer number. iii. Prompt user to select a Car Model number from the selected Manufacturer's list. iv. Retrieve selected_model. v. Call PLOT GRAPH (Step 4.1) with selected_model and graph_type. e. **Else**: Display "Invalid choice." f. Prompt user to "Press ENTER to continue."
- Return to Main Menu loop.

DESIGN

4.1. PLOT GRAPH (Function: plot_graph)

- Extract sales data for model from df.
- Create matplotlib figure and axes.
- Determine the Manufacturer for the title.
- **If** graph_type is 'line': Plot YEARS vs sales_data as a line graph.
- **If** graph_type is 'bar': Plot YEARS vs sales_data as a bar graph.
- Set labels, title, and grid.
- Display the plot using plt.show().

INPUT AND OUTPUT CODE

cse vityarthi project.py - C:\Users\Ali\AppData\Local\Programs\Python\Python314\cse vityarthi project.py (3.14.0)

File Edit Format Run Options Window Help

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

# Global variable to hold the DataFrame
df = None
# Define the expected year range for plotting
YEARS = [2015, 2016, 2017, 2018, 2019, 2020]
# List of known car models/columns for easy selection (Based on CAR_SALES.csv mock file)
CAR_MODELS = ['AMAZE', 'JAZZ', 'WR-V', 'CITY', 'CIVIC', 'CR-V', 'HR-V', 'A8', 'Q2', 'Q8', 'RS7', 'KICKS', 'TERR
# Group the models by manufacturer/type for structured selection in the graph menu
MANUFACTURERS = {
    'HONDA': ['AMAZE', 'JAZZ', 'WR-V', 'CITY', 'CIVIC', 'CR-V', 'HR-V'],
    'AUDI': ['A8', 'Q2', 'Q8', 'RS7'],
    'NISSAN': ['KICKS', 'TERRA'],
    'TATA': ['NANO', 'TIGOR'],
    'MARUTI SUZUKI': ['CELERIO X']
}

def clear_screen():
# For Windows
    if os.name == 'nt':
        os.system('cls')
# For Linux/Mac
    else:
        os.system('clear')

def press_any_key():
    """Waits for user input before continuing."""
    try:
        if sys.version_info[0] < 3:
            raw_input("Press ENTER to continue...")
```

Ln: 1 Col: 0

```
            raw_input("Press ENTER to continue...")
        else:
            input("Press ENTER to continue...")
    except EOFError:
# Handle case where input is redirected
        print("\nContinuing...")

def read_csv_file(filename):
    """Reads the CSV file into a global DataFrame."""
    global df
    try:
# Read the CSV file, setting the 'MODEL' column as the index
        df = pd.read_csv("C:/Users/Ali/Downloads/car analysis 1.csv", index_col='MODEL')
# Ensure all columns are numeric (sales data)
        for col in df.columns:
            try:
                df[col] = pd.to_numeric(df[col], errors='coerce')
            except Exception:
                pass # Keep non-numeric columns as they are
        print(f"\nSuccessfully read {filename}.")
        print(df)
        print(f"\n[{df.shape[0]} rows x {df.shape[1]} columns]")
    except FileNotFoundError:
        print(f"\nERROR: File '{filename}' not found. Please ensure it's in the correct directory.")
    except Exception as e:
        print(f"\nAn error occurred while reading the CSV: {e}")

# --- Data Analysis Menu Functions ---

def show_columns():
    """Displays all column names in the DataFrame."""
```

```
    global df
    if df is None:
        print("\nData not loaded. Please read the CSV file first (Option 1 in Main Menu).")
        return
    print("\n2,1- SHOW COLUMNS")
    print(df.columns)

def show_top_rows():
    """Displays the first N rows."""
    global df
    if df is None:
        print("\nData not loaded. Please read the CSV file first (Option 1 in Main Menu).")
        return
    print("\n2,2- SHOW TOP ROWS")
    try:
        n = int(input("Enter Total rows you want to show: "))
        if n < 0:
            print("Please enter a positive number.")
            return
        print(df.head(n))
        print(f"\n[{df.head(n).shape[0]} rows x {df.shape[1]} columns]")
    except ValueError:
        print("Invalid input. Please enter a number.")

def show_bottom_rows():
    """Displays the last N rows."""
    global df
    if df is None:
        print("\nData not loaded. Please read the CSV file first (Option 1 in Main Menu).")
        return
```

```
        return
    print("\n2,4- SHOW PARTICULAR COLUMN")
    print("Available columns:")
    print(df.columns.values)
    column_name = input("Enter Column Name that You want to print: ").strip().upper()

    if column_name in df.columns:
        print(df[column_name])
    else:
        print(f"Column '{column_name}' not found.")

def add_new_record():
    """Adds a new car model/column to the DataFrame."""
    global df
    if df is None:
        print("\nData not loaded. Please read the CSV file first (Option 1 in Main Menu).")
        return
    print("\n2,5- ADD A NEW RECORD (Car Model)")
    new_model_name = input("Enter New Model Name: ").strip().upper()

    if new_model_name in df.columns:
        print(f"Model '{new_model_name}' already exists.")
        return

    sales_data = {}
    valid_data = True
    for year in YEARS:
        while True:
            try:
```

```
                # The index names in the provided image are 'APPL. SALES IN 2015', etc.
                prompt = f"Enter Sales in {year}: "
                sales = int(input(prompt).strip())
                sales_data[f"APPL. SALES IN {year}"] = sales
                break
            except ValueError:
                print("Invalid input. Sales must be a whole number.")

# Create a new Series to be appended as a column
    new_column = pd.Series(sales_data, name=new_model_name)

# Align the index of the new Series to the existing DataFrame index
# (Since the index is the sales years string, this is crucial)
    df[new_model_name] = new_column

# Update the global list of car models and manufacturers
    if new_model_name not in CAR_MODELS:
        CAR_MODELS.append(new_model_name)
# Assuming new model is generic until user specifies manufacturer
        MANUFACTURERS['OTHER'] = MANUFACTURERS.get('OTHER', []) + [new_model_name]

    print(f"\nSuccessfully added model '{new_model_name}'.")
    print(df)
    print(f"\n[{df.shape[0]} rows x {df.shape[1]} columns]")


def delete_column():
    """Deletes a user-specified column (car model) from the DataFrame."""
    global df
    if df is None:
```

Ln: 1 Col: 0

```
        print("\nData not loaded. Please read the CSV file first (Option 1 in Main Menu).")
        return
    print("\n2,6- DELETE A COLUMN (Car Model)")
    print("Available columns:")
    print(df.columns.values)
    column_name = input("Enter column Name to delete :").strip().upper()

    if column_name in df.columns:
        df.drop(columns=[column_name], inplace=True)
        print(f"\nSuccessfully deleted column '{column_name}'.")
        print(df)
        print(f"\n[{df.shape[0]} rows x {df.shape[1]} columns]")
# Optionally update CAR_MODELS list, though it's less critical for functionality
        try:
            CAR_MODELS.remove(column_name)
            for key in MANUFACTURERS:
                if column_name in MANUFACTURERS[key]:
                    MANUFACTURERS[key].remove(column_name)
        except ValueError:
            pass
# Model might not have been in the initial list
    else:
        print(f"Column '{column_name}' not found.")

def data_summary():
    """Displays descriptive statistics (summary) of the numeric columns."""
    global df
    if df is None:
        print("\nData not loaded. Please read the CSV file first (Option 1 in Main Menu).")
        return
```

Ln: 1 Col: 0

```
    print("\n2,7- DATA SUMMARY")
# Tidy up the output to match the screenshot format
    summary_df = df.describe().transpose()
# Format the numerical output for better readability
    pd.set_option('display.float_format', '{:.4f}'.format)
    print(summary_df)
    print(f"\n[{summary_df.shape[0]} rows x {summary_df.shape[1]} columns]")
    pd.reset_option('display.float_format') # Reset the float format

# --- Graph Menu Functions ---

def plot_graph(model, graph_type='line'):
    """Generates a line or bar graph for a single car model."""
    global df
    if df is None:
        print("\nData not loaded. Please read the CSV file first (Option 1 in Main Menu).")
        return

    if model not in df.columns:
        print(f"Model '{model}' not found in data.")
        return

# Extract sales data for the selected model
    sales_data = df[model].values

# Create the figure and axes
    plt.figure(figsize=(10, 6))

# Define title components
    manufacturer = [m for m, models in MANUFACTURERS.items() if model in models]
    manufacturer = manufacturer[0] if manufacturer else "UNKNOWN"
```

cse vityarthi project.py - C:\Users\Ali\AppData\Local\Programs\Python\Python314\cse vityarthi project.py (3.14.0)

File Edit Format Run Options Window Help

```
    manufacturer = [m for m, models in MANUFACTURERS.items() if model in models]
    manufacturer = manufacturer[0] if manufacturer else "UNKNOWN"

    plt.title(f"{manufacturer}- {model}")

# Determine plot type
    if graph_type == 'line':
        plt.plot(YEARS, sales_data, marker='o', linestyle='-', color='b')
        plt.title(f"Car Wise Sales Count (Line Graph) - {manufacturer}- {model}")
    elif graph_type == 'bar':
        plt.bar(YEARS, sales_data, color='skyblue')
        plt.title(f"Company wise sales count (Bar Graph) - {manufacturer}- {model}")

# Set common labels and grid
    plt.xlabel("Year")
    plt.ylabel("Total Sales")
    plt.xticks(YEARS)
    plt.grid(True)
    plt.show()

def graph_menu():
    """Handles the sub-menu for generating graphs."""
    global df
    if df is None:
        print("\nData not loaded. Please read the CSV file first (Option 1 in Main Menu).")
        return

    while True:
        clear_screen()
        print("3. GRAPH MENU")
        print("------------------------------")
        print("1. Car Wise Line Graph")
        print("2. Car Wise Bar Graph")
```

Ln: 1 Col: 0

cse vityarthi project.py - C:\Users\Ali\AppData\Local\Programs\Python\Python314\cse vityarthi project.py (3.14.0)

File Edit Format Run Options Window Help

```
        print("2. Car Wise Bar Graph")
        print("3. Exit (Move to main menu)")
        print("\nAvailable Manufacturers/Groups:")
        for i, (mfg, models) in enumerate(MANUFACTURERS.items()):
            print(f"  {i+1}. {mfg} ({', '.join(models)})")
        print("-------------------------------")

        choice = input("Enter your choice: ").strip()

        if choice == '5':
            break

        if choice not in ['1', '2','3','4']:
            print("Invalid choice. Please try again.")
            press_any_key()
            continue

        graph_type = 'line' if choice == '1' else 'bar'

# Manufacturer selection
        print("\nSelect a Manufacturer to see available models:")

        mfg_choice = input("Enter Manufacturer choice number: ").strip()
        try:
            mfg_index = int(mfg_choice) - 1
            manufacturer_name = list(MANUFACTURERS.keys())[mfg_index]
            available_models = MANUFACTURERS[manufacturer_name]
        except (ValueError, IndexError):
            print("Invalid manufacturer selection. Returning to Graph Menu.")
            press_any_key()
            continue

        print(f"\nAvailable {manufacturer_name} Models:")
```

cse vityarthi project.py - C:\Users\Ali\AppData\Local\Programs\Python\Python314\cse vityarthi project.py (3.14.0)

File Edit Format Run Options Window Help

```
        print(f"\nAvailable {manufacturer_name} Models:")
        for i, model in enumerate(available_models):
            print(f"  {i+1}. {model}")

# Model selection
        model_choice = input("Enter Car Model choice number: ").strip()
        try:
            model_index = int(model_choice) - 1
            selected_model = available_models[model_index]

            # Generate the plot
            plot_graph(selected_model, graph_type)
            press_any_key()

        except (ValueError, IndexError):
            print("Invalid model selection.")
            press_any_key()

# --- Main Menu Functions ---

def data_analysis_menu():
    """Handles the sub-menu for data analysis operations."""
    while True:
        clear_screen()
        print("2. DATA ANALYSIS MENU")
        print("----------------------")
        print("Data Analysis MENU")
        print("1. Show Columns")
        print("2. Show Top Rows")
        print("3. Show Bottom Rows")
        print("4. Show Specific Column")
        print("5. Add a New Record")
        print("6. Delete a Column")
```

Ln: 1 Col: 0

cse vityarthi project.py - C:\Users\Ali\AppData\Local\Programs\Python\Python314\cse vityarthi project.py (3.14.0)

File Edit Format Run Options Window Help

```
        print("6. Delete a Column")
        print("7. Data Summary")
        print("8. Exit (Move to main menu)")
        print("----------------------")

        choice = input("Enter your choice: ").strip()

        if choice == '1':
            show_columns()
        elif choice == '2':
            show_top_rows()
        elif choice == '3':
            show_bottom_rows()
        elif choice == '4':
            show_specific_column()
        elif choice == '5':
            add_new_record()
        elif choice == '6':
            delete_column()
        elif choice == '7':
            data_summary()
        elif choice == '8':
            break
        else:
            print("\nInvalid choice. Please select an option from 1 to 8.")

        press_any_key()


def main_menu():
    """Displays the main application menu and handles top-level navigation."""
    global df
    print("Welcome to the Car Sales Data Analyzer")
```

Ln: 1 Col: 0

cse vityarthi project.py - C:\Users\Ali\AppData\Local\Programs\Python\Python314\cse vityarthi project.py (3.14.0)

File Edit Format Run Options Window Help

```
# Attempt to load the data immediately if the mock file exists, for convenience
    if os.path.exists("C:/Users/Ali/Downloads/car analysis 1.csv"):
        print("Attempting to load 'CAR_SALES.csv' automatically...")
        read_csv_file("C:/Users/Ali/Downloads/car analysis 1.csv")
        press_any_key()
    else:
        print("NOTE: 'CAR_SALES.csv' not found. Please select option 1 or create the file using the mock conter

    while True:
        clear_screen()
        print("OUTPUTS")
        print("MAIN MENU")
        print("-------------------------------")
        print("MAIN MENU")
        print("1. Read CSV File")
        print("2. Data Analysis Menu")
        print("3. Graph Menu")
        print("4. Exit")
        print("-------------------------------")

        choice = input("Enter your choice: ").strip()

        if choice == '1':
# This is the point where the user would specify the file name
            filename = input("Enter CSV file name (e.g., CAR_SALES.csv): ").strip()
            if not filename:
                filename = "CAR_SALES.csv"
            read_csv_file(filename)
        elif choice == '2':
            if df is not None:
                data_analysis_menu()
            else:
```

cse vityarthi project.py - C:\Users\Ali\AppData\Local\Programs\Python\Python314\cse vityarthi project.py (3.14.0)

File Edit Format Run Options Window Help

```
            if not filename:
                filename = "CAR_SALES.csv"
            read_csv_file(filename)
        elif choice == '2':
            if df is not None:
                data_analysis_menu()
            else:
                print("\nData not loaded. Please read the CSV file first (Option 1).")
        elif choice == '3':
            if df is not None:
                graph_menu()
            else:
                print("\nData not loaded. Please read the CSV file first (Option 1).")
        elif choice == '4':
            print("\nExiting the application. Goodbye!")
            break
        else:
            print("\nInvalid choice. Please select an option from 1 to 4.")

        press_any_key()

if __name__ == "__main__":
# Check for required libraries
    try:
        import pandas as pd
        import matplotlib.pyplot as plt
    except ImportError:
        print("ERROR: This script requires 'pandas' and 'matplotlib'.")
        print("Please install them using: pip install pandas matplotlib")
        sys.exit(1)

    main_menu()
```

Ln: 1 Col: 0

IDLE Shell 3.14.0

File Edit Shell Debug Options Window Help

```
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.
>>>
========== RESTART: C:\Users\Ali\AppData\Local\Programs\Python\Python314\cse vityarthi project.py =========
Welcome to the Car Sales Data Analyzer
Attempting to load 'CAR_SALES.csv' automatically...

Successfully read C:/Users/Ali/Downloads/car analysis 1.csv.
                AMAZE    JAZZ    WR-V    CITY  ...      A8      Q2      Q8     RS7
MODEL                                          ...
SALES IN 2015  257375  828899  260454  257375  ...  828899  260454  257375  828899
SALES IN 2016  541657  139340  692542  541657  ...  139340  692542  541657  139340
SALES IN 2017  911253  106315  495652  911253  ...  106315  495652  911253  106315
SALES IN 2018   51308  601791  601654   51308  ...  601791  601654   51308  601791
SALES IN 2019   28095  490049  453240   28095  ...  490049  453240   28095  490049
SALES IN 2020  610947  602686  954005  610947  ...  602686  954005  610947  602686

[6 rows x 47 columns]

[6 rows x 47 columns]
Press ENTER to continue...
OUTPUTS
MAIN MENU
------------------------------
MAIN MENU
1. Read CSV File
2. Data Analysis Menu
3. Graph Menu
4. Exit
------------------------------
Enter your choice:
```

IDLE Shell 3.14.0

File Edit Shell Debug Options Window Help

```
SALES IN 2019   28095  490049  453240   28095  ...  490049  453240   28095  490049
SALES IN 2020  610947  602686  954005  610947  ...  602686  954005  610947  602686

[6 rows x 47 columns]

[6 rows x 47 columns]
Press ENTER to continue...
OUTPUTS
MAIN MENU
------------------------------
MAIN MENU
1. Read CSV File
2. Data Analysis Menu
3. Graph Menu
4. Exit
------------------------------
Enter your choice: 1
Enter CSV file name (e.g., CAR_SALES.csv): C:/Users/Ali/Downloads/car analysis 1.csv.

Successfully read C:/Users/Ali/Downloads/car analysis 1.csv..
               AMAZE    JAZZ    WR-V    CITY  ...      A8      Q2      Q8      RS7
MODEL                                         ...
SALES IN 2015  257375  828899  260454  257375  ...  828899  260454  257375  828899
SALES IN 2016  541657  139340  692542  541657  ...  139340  692542  541657  139340
SALES IN 2017  911253  106315  495652  911253  ...  106315  495652  911253  106315
SALES IN 2018   51308  601791  601654   51308  ...  601791  601654   51308  601791
SALES IN 2019   28095  490049  453240   28095  ...  490049  453240   28095  490049
SALES IN 2020  610947  602686  954005  610947  ...  602686  954005  610947  602686

[6 rows x 47 columns]

[6 rows x 47 columns]
Press ENTER to continue...
```

```
------------------------------
MAIN MENU
1. Read CSV File
2. Data Analysis Menu
3. Graph Menu
4. Exit
------------------------------
Enter your choice: 2
2. DATA ANALYSIS MENU
---------------------
Data Analysis MENU
1. Show Columns
2. Show Top Rows
3. Show Bottom Rows
4. Show Specific Column
5. Add a New Record
6. Delete a Column
7. Data Summary
8. Exit (Move to main menu)
---------------------
Enter your choice: 1

2,1- SHOW COLUMNS
Index(['AMAZE', 'JAZZ', 'WR-V', 'CITY', 'CIVIC', 'CR-V', 'HR-V', 'i10', 'i20',
       'CRETA', 'VENUE', 'VERNA', 'SANTRO', 'ELANTRA', 'XCENT', 'SONATA',
       'PALISADE', 'KONA', 'BALENO', 'SWIFT', 'DZIRE', 'ALTO', 'CIAZ', 'EECO',
       'IGNIS', 'CELERIO X', 'S-CROSS', 'XL6', 'TIAGO', 'NEXON', 'TIGOR',
       'HARRIER', 'ALTROZ', 'NANO', 'SAFARI', 'HBX', 'SUNNY', 'MAGNITE',
       'KICKS', 'GT-R', 'TERRA', 'A4', 'A6', 'A8', 'Q2', 'Q8', 'RS7'],
      dtype='object')
Press ENTER to continue...
```

```
4. Show Specific Column
5. Add a New Record
6. Delete a Column
7. Data Summary
8. Exit (Move to main menu)
----------------------
Enter your choice: 5

2,5- ADD A NEW RECORD (Car Model)
Enter New Model Name: HUMMER
Enter Sales in 2015: 68000
Enter Sales in 2016: 70000
Enter Sales in 2017: 55
Enter Sales in 2018:
Invalid input. Sales must be a whole number.
Enter Sales in 2018: 55000
Enter Sales in 2019: 90000
Enter Sales in 2020: 89000

Successfully added model 'HUMMER'.
```

IDLE Shell 3.14.0

File Edit Shell Debug Options Window Help

```
4. Show Specific Column
5. Add a New Record
6. Delete a Column
7. Data Summary
8. Exit (Move to main menu)
----------------------
Enter your choice: 8
Press ENTER to continue...
OUTPUTS
MAIN MENU
------------------------------
MAIN MENU
1. Read CSV File
2. Data Analysis Menu
3. Graph Menu
4. Exit
------------------------------
Enter your choice: 3
3. GRAPH MENU
------------------------------
1. Car Wise Line Graph
2. Car Wise Bar Graph
3. Exit (Move to main menu)

Available Manufacturers/Groups:
  1. HONDA (AMAZE, JAZZ, WR-V, CITY, CIVIC, CR-V, HR-V)
  2. AUDI (A8, Q2, Q8, RS7)
  3. NISSAN (KICKS, TERRA)
  4. TATA (NANO, TIGOR)
  5. MARUTI SUZUKI (CELERIO X)
  6. OTHER (HUMMER)
------------------------------
Enter your choice:
```

IDLE Shell 3.14.0

File Edit Shell Debug Options Window Help

```
1. Read CSV File
2. Data Analysis Menu
3. Graph Menu
4. Exit
------------------------------
Enter your choice: 3
3. GRAPH MENU
------------------------------
1. Car Wise Line Graph
2. Car Wise Bar Graph
3. Exit (Move to main menu)

Available Manufacturers/Groups:
  1. HONDA (AMAZE, JAZZ, WR-V, CITY, CIVIC, CR-V, HR-V)
  2. AUDI (A8, Q2, Q8, RS7)
  3. NISSAN (KICKS, TERRA)
  4. TATA (NANO, TIGOR)
  5. MARUTI SUZUKI (CELERIO X)
------------------------------
Enter your choice: 1

Select a Manufacturer to see available models:
Enter Manufacturer choice number: 1

Available HONDA Models:
  1. AMAZE
  2. JAZZ
  3. WR-V
  4. CITY
  5. CIVIC
  6. CR-V
  7. HR-V
Enter Car Model choice number: 1
```

LINE GRAPH

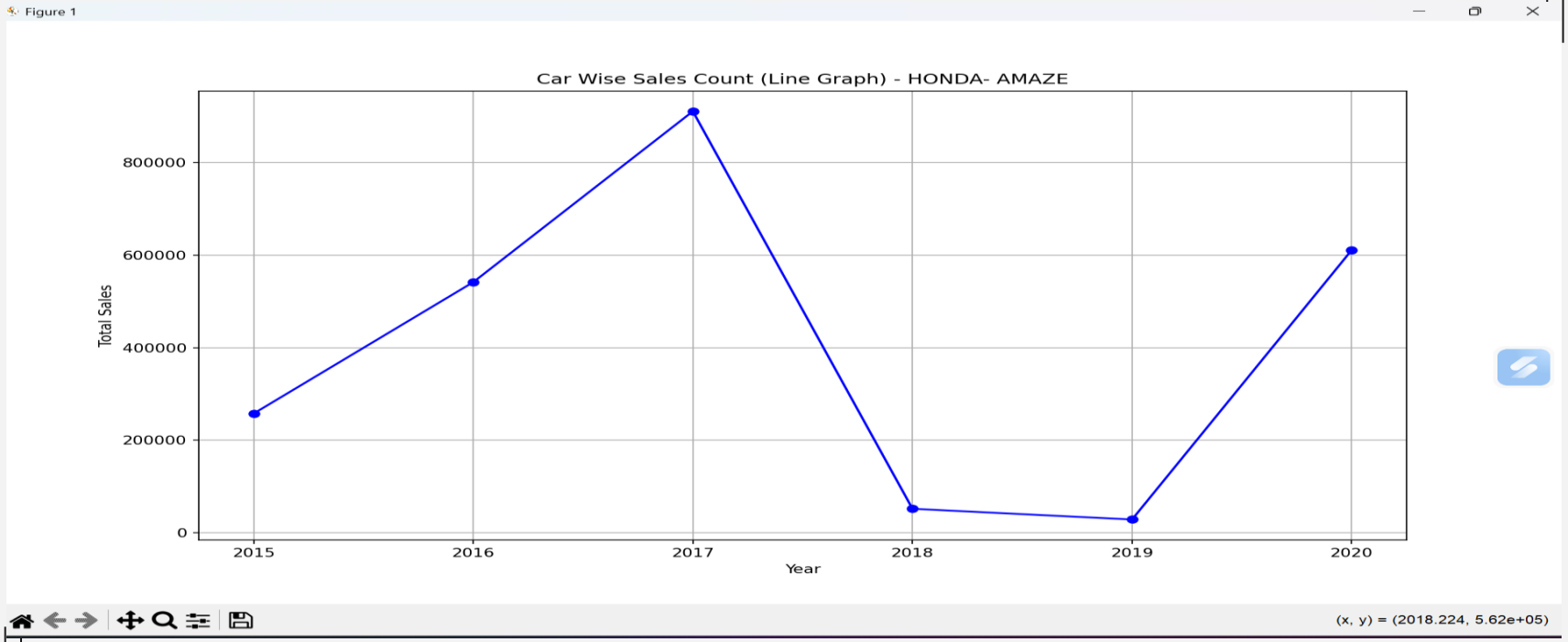


Figure 1
Car Wise Sales Count (Line Graph) - HONDA- AMAZE
Total Sales
800000
600000
400000
200000
0
2015
2016
2017
2018
2019
2020
Year
(x, y) = (2018.224, 5.62e+05)

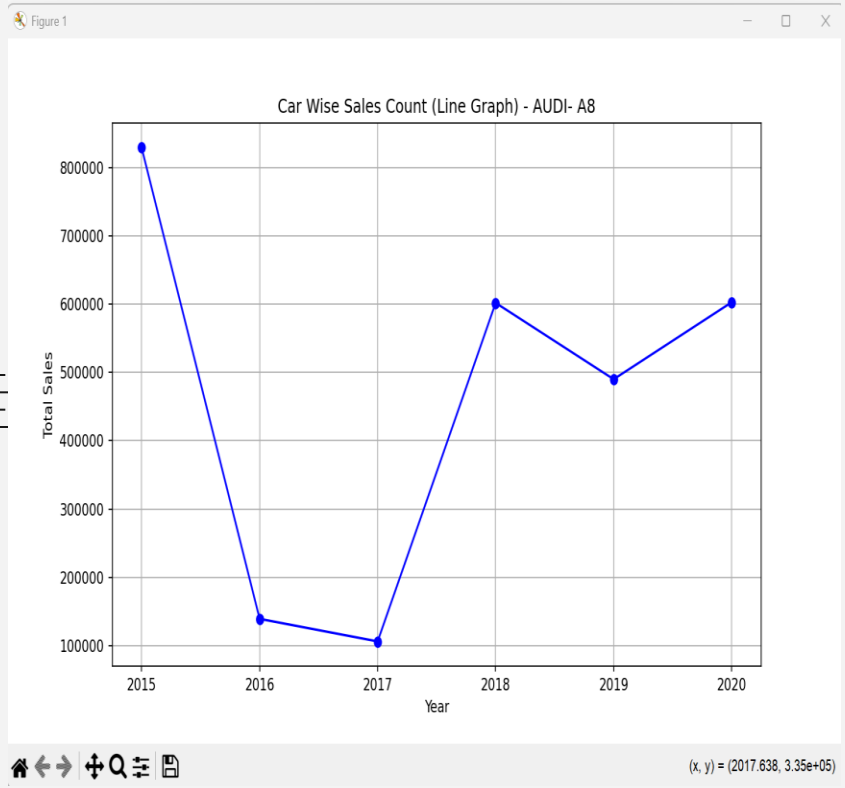


Figure 1
Car Wise Sales Count (Line Graph) - AUDI- A8
Total Sales
800000
700000
600000
500000
400000
300000
200000
100000
2015
2016
2017
2018
2019
2020
Year
(x, y) = (2017.638, 3.35e+05)

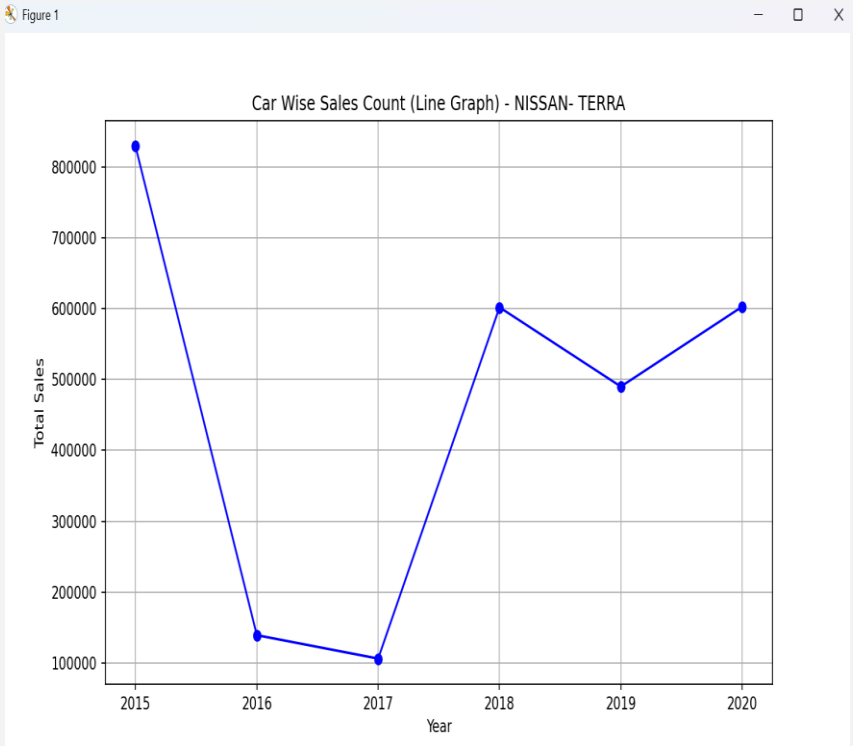


Figure 1
Car Wise Sales Count (Line Graph) - NISSAN- TERRA
Total Sales
800000
700000
600000
500000
400000
300000
200000
100000
2015
2016
2017
2018
2019
2020
Year

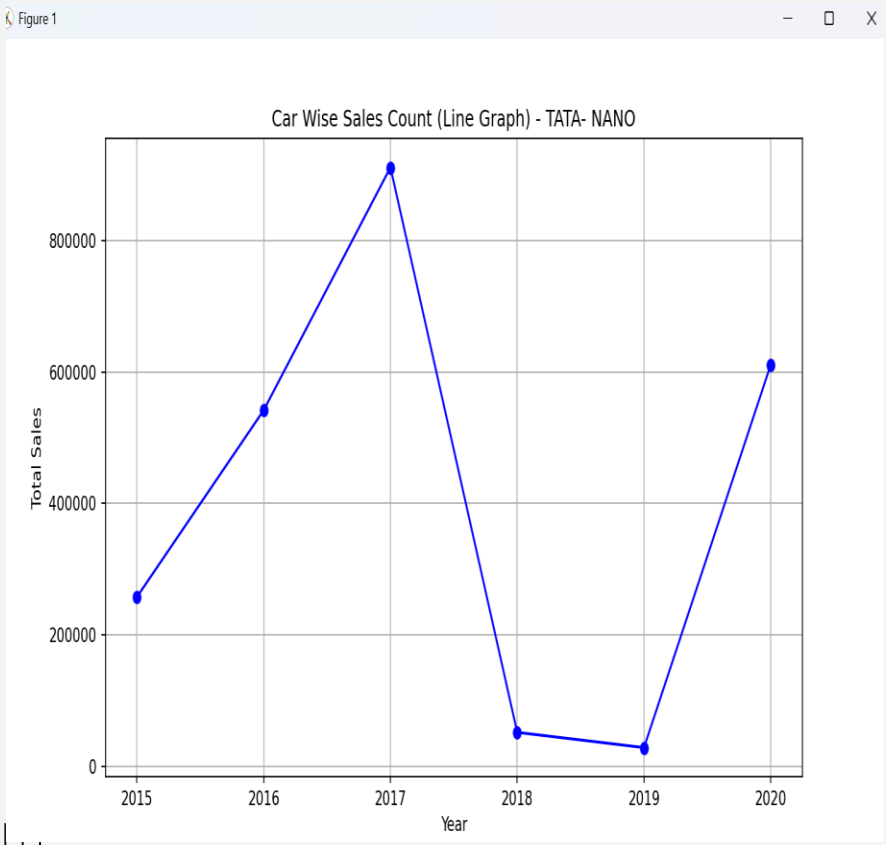


Figure 1
Car Wise Sales Count (Line Graph) - TATA- NANO
Total Sales
Year
0
200000
400000
600000
800000
2015
2016
2017
2018
2019
2020

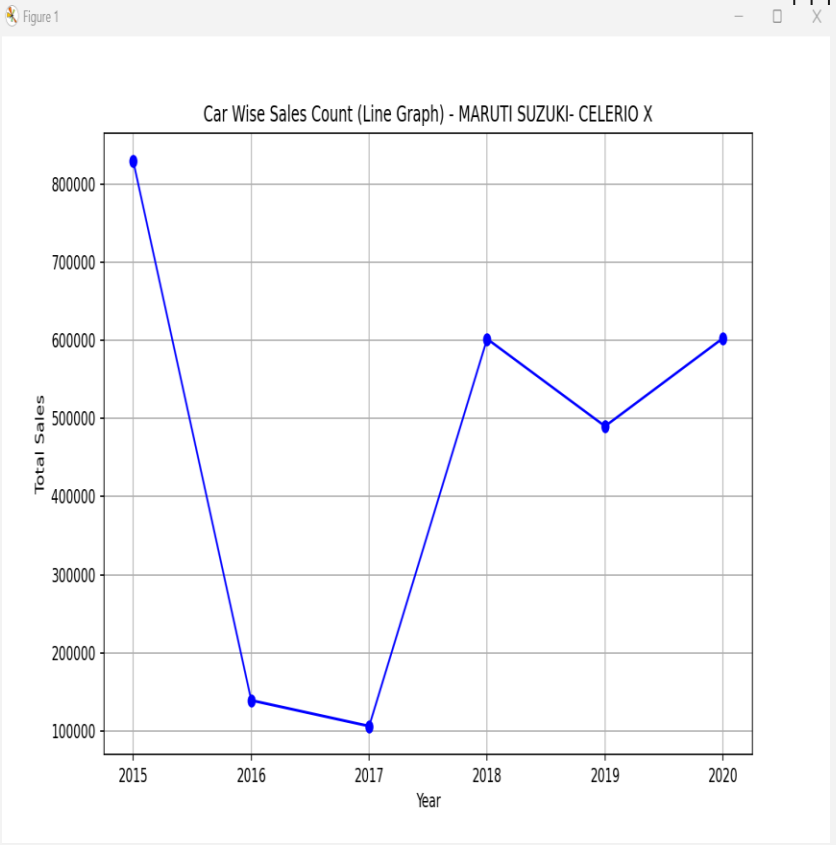


Figure 1
Car Wise Sales Count (Line Graph) - MARUTI SUZUKI- CELERIO X
Total Sales
Year
100000
200000
300000
400000
500000
600000
700000
800000
2015
2016
2017
2018
2019
2020

BAR GRAPH

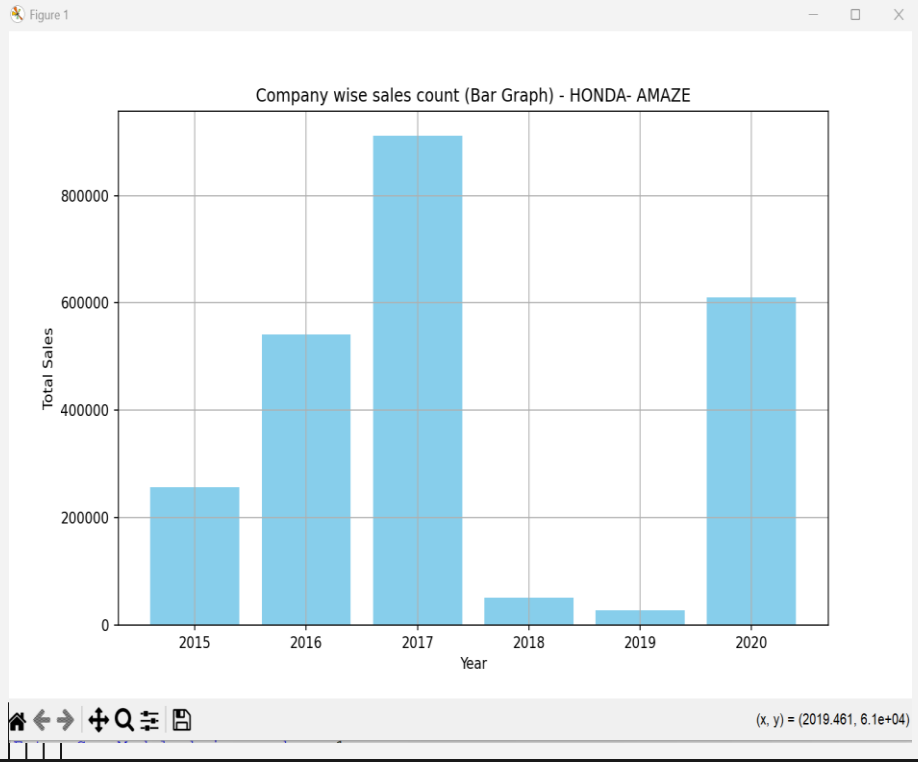


Figure 1
Company wise sales count (Bar Graph) - HONDA- AMAZE
Total Sales
800000
600000
400000
200000
0
2015
2016
2017
2018
2019
2020
Year
(x, y) = (2019.461, 6.1e+04)

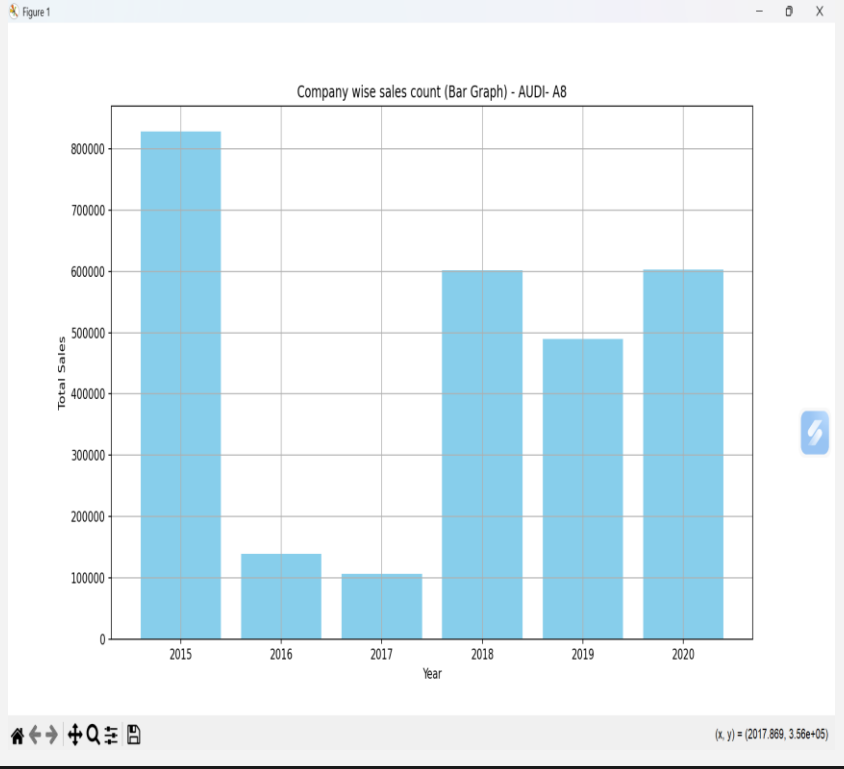


Figure 1
Company wise sales count (Bar Graph) - AUDI- A8
Total Sales
800000
700000
600000
500000
400000
300000
200000
100000
0
2015
2016
2017
2018
2019
2020
Year
(x, y) = (2017.869, 3.56e+05)

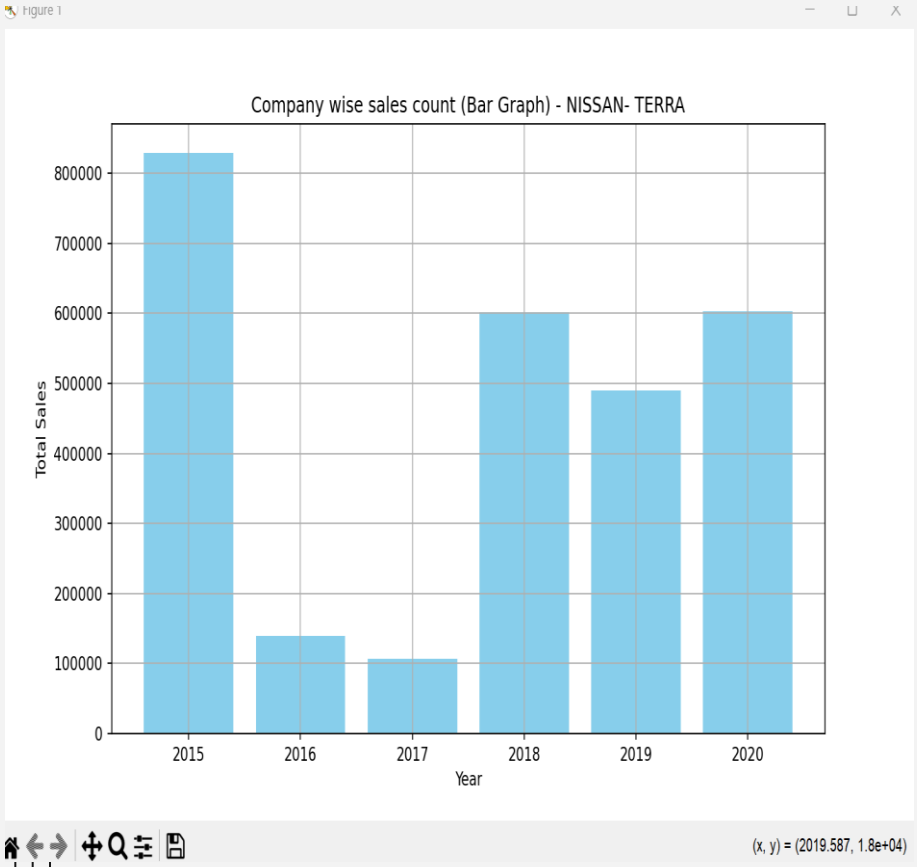


Figure 1
Company wise sales count (Bar Graph) - NISSAN- TERRA
Total Sales
0
100000
200000
300000
400000
500000
600000
700000
800000
2015
2016
2017
2018
2019
2020
Year
(x, y) = (2019.587, 1.8e+04)

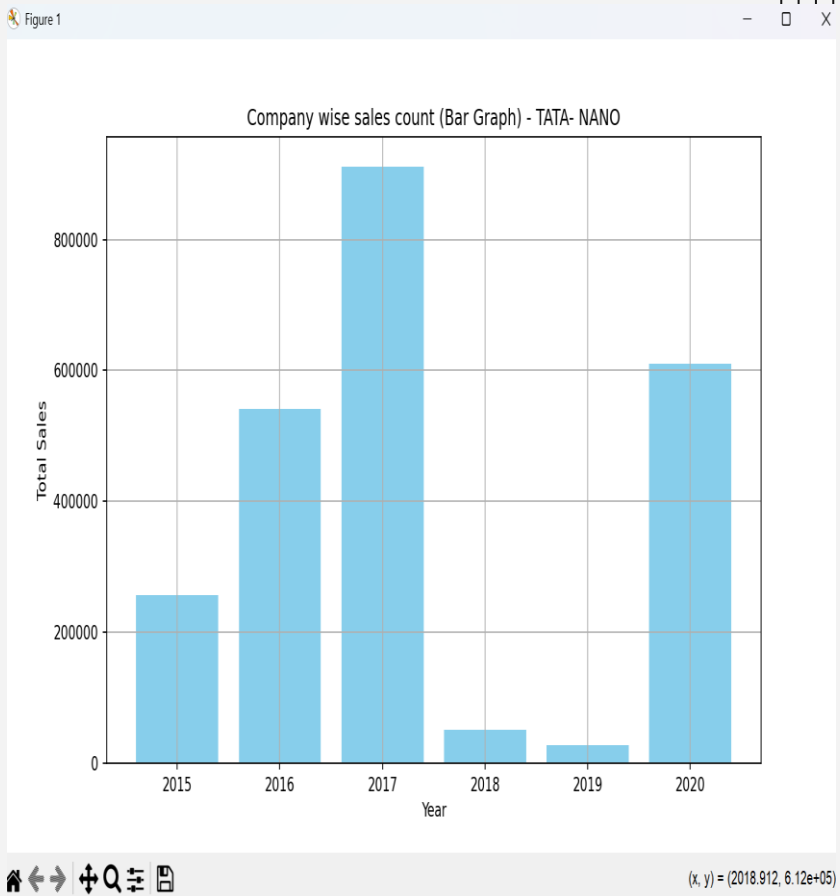


Figure 1
Company wise sales count (Bar Graph) - TATA- NANO
Total Sales
0
200000
400000
600000
800000
2015
2016
2017
2018
2019
2020
Year
(x, y) = (2018.912, 6.12e+05)

```
--------------------------------
MAIN MENU
1. Read CSV File
2. Data Analysis Menu
3. Graph Menu
4. Exit
--------------------------------
Enter your choice: 4

Exiting the application. Goodbye!
>>
```