**Report**

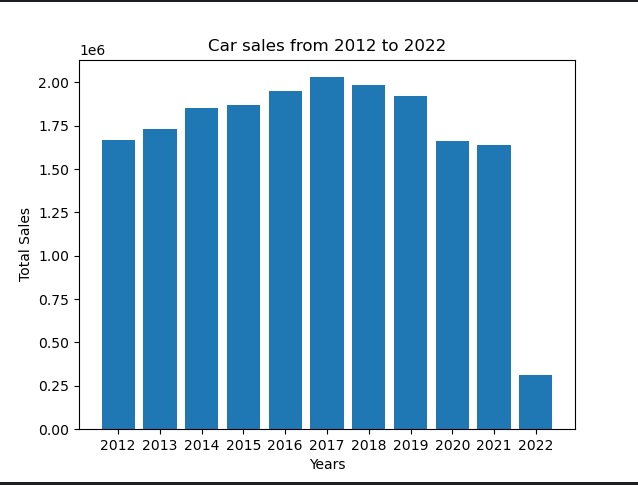
**Part one**

import csv  
import matplotlib.pyplot as plt  
  
with open('Data.csv', mode='r') as file:  
 data = csv.reader(file)  
 next(data) # Skip the header column  
 years = []  
 sales = []  
  
 for row in data:  
 year = row.pop(0) # Remove the year from the sales  
 years.append(year) # create a list containing the years  
  
 total\_sales = 0 # Initialize the total\_sales as 0  
 for item in row:  
 total\_sales += int(item) # add all the sales by converting them to integers  
 sales.append(total\_sales) # create a list containing the total sales per year  
  
 with open('stats.txt', mode='a') as stats: # Write the year and total using append  
 stats.write(f"{year}: {total\_sales}\n") # add newline after each  
  
# Plotting values for each year  
x = years  
y = sales  
  
plt.figure(1)  
plt.bar(x, y)  
  
plt.title("Car sales from 2012 to 2022") # Writing plot title  
plt.xlabel("Years") # Writing x-axis label  
plt.ylabel("Total Sales") # Writing y-axis label  
  
plt.show()

This Python code reads data from the data CSV file that contains information about car sales for different years. It then calculates the total sales for each year and stores them in two lists, years and sales. The data is also written to a text file named 'stats.txt'.

After processing the data, a bar graph is created using the years and sales lists. The graph shows the total sales for each year and is given a title ('Car sales from 2012 to 2022') and labels for the x and y axis ('Years' and 'Total Sales' respectively). Finally, the graph is displayed using the plt.show() function.

In summary, this code reads car sales data from a CSV file, calculates the total sales for each year, writes the data to a text file, and creates a bar graph to visualize the total sales for each year as shown below:



**Part two**

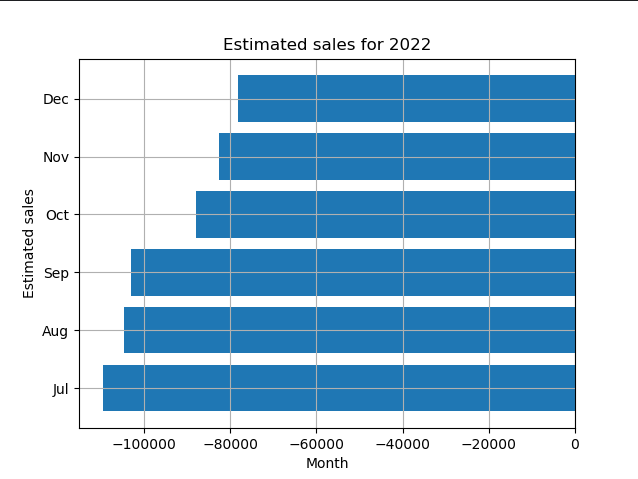
import csv  
import matplotlib.pyplot as plt  
  
# Read the file  
with open('Data.csv', mode='r') as file:  
 data = csv.reader(file)  
 data\_2021 = []  
 for row in data:  
 if row[0] == "Month":  
 last\_months = row[-6:] # Get the last 6-month names  
 elif row[0] == "2021":  
 data\_2021 = row # Add the data values for 2021 in a list  
 sales\_2021 = sum([int(x) for x in row[1:7]]) # Calculate the sum of the first 6 months 2021  
 elif row[0] == "2022":  
 sales\_2022 = sum([int(x) for x in row[1:7]]) # Calculate the sum of the first 6 months 2022  
  
 sales\_growth\_rate = (sales\_2022 - sales\_2021) / sales\_2022  
  
 estimated\_values = [] # Initialize a list for the estimated values  
  
 i = 0 # Counter for months  
 # Loop through values in the 2021 list for the last 6 months  
 for month\_value\_2021 in [int(x) for x in data\_2021[-6:]]:  
 month\_value\_2022 = month\_value\_2021 + month\_value\_2021 \* sales\_growth\_rate # Get the 2022 estimated values  
 estimated\_values.append(month\_value\_2022)  
  
 with open('stats.txt', mode='a') as stats:  
 stats.write(f"{last\_months[i]}: {month\_value\_2022}\n") # Append the values in the existing stats file  
 i += 1  
  
# Horizontal Bar Plot  
x = last\_months # x values to be months  
y = estimated\_values # y values to be the estimated values  
  
plt.figure(1)  
plt.barh(x, y)  
  
plt.title("Estimated sales for 2022")  
plt.xlabel("Month")  
plt.ylabel("Estimated sales")  
plt.grid() # Showing grids on the plot  
  
plt.show()

This code uses two Python modules to read data from the data CSV file and create a graph. It starts by opening the CSV file and reading its contents using the csv.reader() function. The code then extracts the sales data for 2021 and calculates the sales growth rate between 2021 and 2022.

The estimated sales data for the last 6 months of 2022 is then calculated using the sales growth rate and the sales data for the same period in 2021. These estimated sales data are stored in a list called estimated\_values, and are also written to a text file named 'stats.txt' for later use.

Finally, the code uses the matplotlib.pyplot module to create a horizontal bar graph showing the estimated sales data. The x-axis shows the last 6 months' names, while the y-axis shows the estimated sales data. The graph is given a title and axis labels before being displayed.

Overall, this code reads data, performs calculations and visualizes the results in a graph as shown below.



**Stats.txt**

2012: 1665063  
2013: 1728140  
2014: 1851645  
2015: 1867498  
2016: 1948375  
2017: 2029668  
2018: 1987373  
2019: 1921449  
2020: 1661560  
2021: 1638340  
2022: 312230  
  
Jul: -109563.49828651955  
Aug: -104612.8465554239  
Sep: -103008.19697018224  
Oct: -87980.17407039684  
Nov: -82480.56327066588  
Dec: -78068.85049482755

The first part of this text file is writing down the calculated sales of total vehicles sold in each year starting from 2012 to 2021.

The second part of the text is writing down the estimated sales for last six months in month M of 2022.