SRI RAMACHANDRA
INSTITUTE OF HIGHER EDUCATION AND RESEARCH
(Category - I Deemed to be University) Porur, Chennai
SRI RAMACHANDRA FACULTY OF ENGINEERING AND TECHNOLOGY

ANALYSIS OF BOOK SALES DATASET USING R PROGRAMMING

PROJECT REPORT

Submitted by

JOTHIPRIYAN M A - E5223013

BACHELOR OF SCIENCE

In

ARTIFICIAL INTELLIGENCE AND DATAANALYTICS

Sri Ramachandra Faculty of Engineering and Technology

Sri Ramachandra Institute of Higher Education and Research,

Porur, Chennai – 600116

APRIL, 2024

SRI RAMACHANDRA
INSTITUTE OF HIGHER EDUCATION AND RESEARCH
(Category - I Deemed to be University) Porur, Chennai
SRI RAMACHANDRA FACULTY OF ENGINEERING AND TECHNOLOGY

BONAFIDE CERTIFICATE

Certified that this project report is the bonafide record of work done by

"JOTHIPRIYAN M A - E5223013".

Signature of the Supervisor

Dr.Arundhati Mahesh

Assistant Professor,

Department of Computer Science

Signature of the Coordinator

Dr. A. Christoper Tamilmathi

Assistant Professor,

Department of Computer Science

Sri Ramachandra Faculty of Engineering and Technology,

SRIHER, Porur, Chennai-600116

Examination Date:

Internal **External**

ACKNOWLEDGEMENT

I express my sincere gratitude to our Programme Coordinator **Dr. A. Christoper Tamilmathi** for her support and for providing the required facilities for carrying out this study.

I wish to thank my faculty supervisor(s), **Dr .Arundhati Mahesh** partment of Computer Science, Sri Ramachandra Faculty of Engineering and Technology for extending help and encouragement throughout the project. Without his/her continuous guidance and persistent help, this project would not have been a success for me.

I am grateful to all the members of Sri Ramachandra Faculty of Engineering and Technology, my beloved parents and friends for extending the support, who helped us to overcome obstacles in the study.

INTRODUCTION TO R PROGRAMMING

R is a powerful and versatile programming language primarily used for statistical analysis and data visualization. It was created by statisticians and is widely adopted by researchers, data analysts, and professionals in various fields. One of the main reasons for R's popularity is its extensive collection of packages, which are collections of functions and datasets contributed by users from around the world. These packages cover a wide range of tasks, from basic data manipulation to advanced machine learning algorithms. R is known for its readability and ease of use, especially for statistical operations. It has a syntax that's relatively straightforward to learn, making it accessible to beginners while still offering advanced capabilities for experienced users.

In addition to its statistical prowess, R excels in creating high-quality graphics and visualizations. With packages like ggplot2, users can easily generate a wide variety of plots to explore and communicate their data effectively. Overall, R is a valuable tool for anyone working with data, whether you're analyzing survey results, conducting experiments, or building predictive models. Its open-source nature, vast community support, and rich ecosystem of packages make it an essential language in the field of data science and statistics.

R STUDIO:

RStudio is a popular integrated development environment (IDE) for the R programming language. Think of it as a software tool that provides a convenient workspace for writing and running R code. It offers a user friendly interface with features like syntax highlighting, code completion, and built-in tools for data visualization. This makes it easier for users to write, debug, and organize their R code efficiently. One of the key benefits of RStudio is its seamless integration with R. It simplifies the process of working with R by providing a centralized platform where you can write code, view plots, manage files, and access help documentation all in one place. Additionally, RStudio supports project management, which allows you to organize your work into separate projects, making it easier to keep your files and data organized.

What is a dataset in R?

Datasets are structured collections of data, typically organized into rows and columns, similar to tables in a spreadsheet. In R programming, datasets are useful for organizing and storing data in a format that can be easily manipulated and analysed. Datasets play a crucial role in data analysis and statistical programming with R because they provide a standardized way to represent and work with data. By loading datasets into R, users can perform various operations such as data manipulation, statistical analysis, and visualization. Datasets can come in various forms, including CSV files, Excel spreadsheets, databases, or built-in datasets provided by R packages. They may contain different types of data, such as numerical, categorical, or textual, allowing users to work with diverse datasets for different analytical tasks. In R programming, datasets can be loaded into memory using functions like read.csv() for CSV files or read.table() for tabular data. Once loaded, users can perform operations like filtering, sorting, summarizing, and modeling on the dataset using R's extensive library of functions and packages. Overall, datasets serve as the foundation for data analysis in R programming, enabling users to organize, explore, and derive insights from their data efficiently.

DOWNLOADING A DATASET

1. Sign in or create an account on Kaggle.

2. Search for the desired dataset and access its page.

3. Choose the CSV format for download.

4. Click the download button and wait for the CSV file(s) to download.

5. If downloaded as a ZIP file, extract its contents to access the CSV file(s).

6. The CSV file(s) are now ready for use in your data analysis projects.

ABOUT MY DATASET

1.Government Records: The government keeps track of things like vehicle registrations and sales, so they have data on which cars are being sold and where.

2. Industry Reports: Companies that study the car market, like JD Power or Kelley Blue Book, share reports and data about things like how many cars are being sold and what kinds of cars people are buying.

3. Websites: Places where people buy and sell cars online, like AutoTrader or eBay Motors, might have data on the sales of used cars.

4. Car Manufacturers: The companies that make cars also keep track of how many they sell and which models are popular.

5. Research Studies: Sometimes, researchers or analysts study the car market and publish their findings, which can include data on things like sales predictions or trends.

6. Social Media: People often talk about buying and selling cars on social media platforms like Twitter or Reddit, so there might be useful information there too.

Basically, these datasets give us information about what cars people are buying, how many are being sold, and where they're being sold. They're helpful for understanding trends in the car market and making decisions about things like pricing or advertising.

IMPORTING A DATASET INTO R STUDIO CONSOLE

To import a dataset into our R programming environment, we must perform the following steps:
1) Set the working directory of your data set file by copying its path.

```
setwd("C:/Users/Leela/OneDrive/Desktop")
```

2) Create a vector of your own and use the read.csv function to read your dataset file.

```
cars<-data.frame(read.csv("cars.csv")
```

3)Now use the head() function to retrieve the first few rows of your dataset to work on it.

```
cars<-head(cars)
```

OUTPUT:

```
> cars<-head(cars)
> cars
  Manufacturer   Model Sales_in_thousands X__year_resale_value Vehicle_type Price_in_thousands Engine_size
1        Acura Integra             16.919              16.360    Passenger              21.50         1.8
2        Acura      TL             39.384              19.875    Passenger              28.40         3.2
3        Acura      CL             14.114              18.225    Passenger                 NA         3.2
4        Acura      RL              8.588              29.725    Passenger              42.00         3.5
5         Audi      A4             20.397              22.255    Passenger              23.99         1.8
6         Audi      A6             18.780              23.555    Passenger              33.95         2.8
  Horsepower Wheelbase Width Length Curb_weight Fuel_capacity Fuel_efficiency Latest_Launch Power_perf_factor
1        140     101.2  67.3  172.4       2.639          13.2              28      2/2/2012          58.28015
2        225     108.1  70.3  192.9       3.517          17.2              25      6/3/2011          91.37078
3        225     106.9  70.6  192.0       3.470          17.2              26      1/4/2012                NA
4        210     114.6  71.4  196.6       3.850          18.0              22     3/10/2011          91.38978
5        150     102.6  68.2  178.0       2.998          16.4              27     10/8/2011          62.77764
6        200     108.7  76.1  192.0       3.561          18.5              22      8/9/2011          84.56511
```

PERFORMING DIFFERENT FUNCTIONS ON OUR DATASET:

DATA CLEANING

1) Working on missing data:

- is.na( ) is a function which returns a logical vector indicating whether each element in a vector or column of a data frame is missing.

is.na(cars)

OUTPUT:

```
> is.na(cars)
  Manufacturer Model Sales_in_thousands X__year_resale_value Vehicle_type Price_in_thousands Engine_size
1        FALSE FALSE              FALSE                FALSE        FALSE              FALSE       FALSE
2        FALSE FALSE              FALSE                FALSE        FALSE              FALSE       FALSE
3        FALSE FALSE              FALSE                FALSE        FALSE               TRUE       FALSE
4        FALSE FALSE              FALSE                FALSE        FALSE              FALSE       FALSE
5        FALSE FALSE              FALSE                FALSE        FALSE              FALSE       FALSE
6        FALSE FALSE              FALSE                FALSE        FALSE              FALSE       FALSE
  Horsepower Wheelbase Width Length Curb_weight Fuel_capacity Fuel_efficiency Latest_Launch Power_perf_factor
1      FALSE     FALSE FALSE  FALSE       FALSE         FALSE          FALSE         FALSE             FALSE
2      FALSE     FALSE FALSE  FALSE       FALSE         FALSE          FALSE         FALSE             FALSE
3      FALSE     FALSE FALSE  FALSE       FALSE         FALSE          FALSE         FALSE              TRUE
4      FALSE     FALSE FALSE  FALSE       FALSE         FALSE          FALSE         FALSE             FALSE
5      FALSE     FALSE FALSE  FALSE       FALSE         FALSE          FALSE         FALSE             FALSE
6      FALSE     FALSE FALSE  FALSE       FALSE         FALSE          FALSE         FALSE             FALSE
```

- complete.cases( ) function returns a logical vector indicating which rows have no missing values across all columns

complete.cases(cars)

OUTPUT:

```
> complete.cases(cars)
[1]  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
```

- We can also remove rows or columns containing missing values using functions like na.omit ( ) or complete.cases( ) .

clean<-na.omit(cars)
clean

OUTPUT:

```
> clean<-na.omit(cars)
> clean
  Manufacturer   Model Sales_in_thousands X__year_resale_value Vehicle_type Price_in_thousands Engine_size
1        Acura Integra              16.919               16.360    Passenger              21.50         1.8
2        Acura      TL              39.384               19.875    Passenger              28.40         3.2
4        Acura      RL               8.588               29.725    Passenger              42.00         3.5
5         Audi      A4              20.397               22.255    Passenger              23.99         1.8
6         Audi      A6              18.780               23.555    Passenger              33.95         2.8
  Horsepower Wheelbase Width Length Curb_weight Fuel_capacity Fuel_efficiency Latest_Launch Power_perf_factor
1        140     101.2  67.3  172.4       2.639          13.2              28      2/2/2012          58.28015
2        225     108.1  70.3  192.9       3.517          17.2              25      6/3/2011          91.37078
4        210     114.6  71.4  196.6       3.850          18.0              22     3/10/2011          91.38978
5        150     102.6  68.2  178.0       2.998          16.4              27     10/8/2011          62.77764
6        200     108.7  76.1  192.0       3.561          18.5              22      8/9/2011          84.56511
```

2) Imputing missing values:

In R, imputation refers to the process of filling in missing values within a dataset with estimated or predicted values. Missing data is a common issue in real-world datasets and can adversely affect the results of statistical analysis or machine learning models. Imputation methods in R help address this issue by replacing missing values with plausible substitutes based on the available data. R offers various imputation techniques, including mean imputation, median imputation, mode imputation, regression imputation, and more, each suitable for different types of data and scenarios. Imputation methods can be implemented using built-in functions from packages like 'mice', 'missForest', or 'imputeTS'. By imputing missing values, analysts ensure the completeness and reliability of their datasets, enabling more robust and accurate analysis and modeling in R

- Replacing missing values using the mean of the column

```
cars$PC2[is.na(cars$Horsepower)]<-mean(cars$Horsepower,na.rm=TRUE)

cars
```

OUTPUT:

```
> cars$PC2[is.na(cars$Horsepower)]<-mean(cars$Horsepower,na.rm=TRUE)
> cars
  Manufacturer   Model Sales_in_thousands X__year_resale_value Vehicle_type Price_in_thousands Engine_size
1        Acura Integra             16.919               16.360    Passenger              21.50         1.8
2        Acura      TL             39.384               19.875    Passenger              28.40         3.2
3        Acura      CL             14.114               18.225    Passenger                 NA         3.2
4        Acura      RL              8.588               29.725    Passenger              42.00         3.5
5         Audi      A4             20.397               22.255    Passenger              23.99         1.8
6         Audi      A6             18.780               23.555    Passenger              33.95         2.8
  Horsepower Wheelbase Width Length Curb_weight Fuel_capacity Fuel_efficiency Latest_Launch Power_perf_factor
1        140     101.2  67.3  172.4       2.639          13.2              28      2/2/2012          58.28015
2        225     108.1  70.3  192.9       3.517          17.2              25      6/3/2011          91.37078
3        225     106.9  70.6  192.0       3.470          17.2              26      1/4/2012                NA
4        210     114.6  71.4  196.6       3.850          18.0              22     3/10/2011          91.38978
5        150     102.6  68.2  178.0       2.998          16.4              27     10/8/2011          62.77764
6        200     108.7  76.1  192.0       3.561          18.5              22      8/9/2011          84.56511
  PC2
```

REMOVE DUPILICATE VALUE

```
unique_cars <-unique(cars)
```

OUTPUT:

```
unique_cars
Manufacturer   Model Sales_in_thousands X__year_resale_value Vehicle_type Price_in_thousands Engine_size
       Acura Integra             16.919               16.360    Passenger              21.50         1.8
       Acura      TL             39.384               19.875    Passenger              28.40         3.2
       Acura      CL             14.114               18.225    Passenger                 NA         3.2
       Acura      RL              8.588               29.725    Passenger              42.00         3.5
        Audi      A4             20.397               22.255    Passenger              23.99         1.8
        Audi      A6             18.780               23.555    Passenger              33.95         2.8
Horsepower Wheelbase Width Length Curb_weight Fuel_capacity Fuel_efficiency Latest_Launch Power_perf_factor
       140     101.2  67.3  172.4       2.639          13.2              28      2/2/2012          58.28015
       225     108.1  70.3  192.9       3.517          17.2              25      6/3/2011          91.37078
       225     106.9  70.6  192.0       3.470          17.2              26      1/4/2012                NA
       210     114.6  71.4  196.6       3.850          18.0              22     3/10/2011          91.38978
       150     102.6  68.2  178.0       2.998          16.4              27     10/8/2011          62.77764
       200     108.7  76.1  192.0       3.561          18.5              22      8/9/2011          84.56511
PC2
 NA
 NA
 NA
 NA
 NA
 NA
```

1) **DETECT AND REMOVE THE OUTLIERS**

cars$Horsepower[cars$Horsepower%in%boxplot.stats(car

s$Horsepower)$out]

OUTPUT:

```
> cars$Horsepower[cars$Horsepower%in%boxplot.stats(cars$Horsepower)$out]
integer(0)
```

2) **MODIFY COLUMN NAMES**

colnames(cars)[colnames(cars)=="price_in_thousands"]<-"price"
colnames(cars)

OUTPUT:

```
> colnames(cars)[colnames(cars)=="price_in_thousands"]<-"price"
> colnames(cars)
 [1] "Manufacturer"         "Model"                "Sales_in_thousands"   "X__year_resale_value"
 [5] "Vehicle_type"         "Price_in_thousands"   "Engine_size"          "Horsepower"
 [9] "Wheelbase"            "Width"                "Length"               "Curb_weight"
[13] "Fuel_capacity"        "Fuel_efficiency"      "Latest_Launch"        "Power_perf_factor"
[17] "PC2"
```

3) **Remove spaces in Character Strings**

Withoutspace<-gsub(" ","",cars)
Withoutspace

OUTPUT:

```
> Withoutspace<-gsub(" ","",cars)
> Withoutspace
 [1] "c(\"Acura\",\"Acura\",\"Acura\",\"Acura\",\"Audi\",\"Audi\")"
 [2] "c(\"Integra\",\"TL\",\"CL\",\"RL\",\"A4\",\"A6\")"
 [3] "c(16.919,39.384,14.114,8.588,20.397,18.78)"
 [4] "c(16.36,19.875,18.225,29.725,22.255,23.555)"
 [5] "c(\"Passenger\",\"Passenger\",\"Passenger\",\"Passenger\",\"Passenger\",\"Passenger\")"
 [6] "c(21.5,28.4,NA,42,23.99,33.95)"
 [7] "c(1.8,3.2,3.2,3.5,1.8,2.8)"
 [8] "c(140,225,225,210,150,200)"
 [9] "c(101.2,108.1,106.9,114.6,102.6,108.7)"
[10] "c(67.3,70.3,70.6,71.4,68.2,76.1)"
[11] "c(172.4,192.9,192,196.6,178,192)"
[12] "c(2.639,3.517,3.47,3.85,2.998,3.561)"
[13] "c(13.2,17.2,17.2,18,16.4,18.5)"
[14] "c(28,25,26,22,27,22)"
[15] "c(\"2/2/2012\",\"6/3/2011\",\"1/4/2012\",\"3/10/2011\",\"10/8/2011\",\"8/9/2011\")"
[16] "c(58.28014952,91.37077766,NA,91.38977933,62.7776392,84.56510502)"
[17] "c(NA,NA,NA,NA,NA,NA)"
```

DATA VISUALISATION

Data visualization techniques are essential tools used to represent complex data in a graphical or visual format. They serve to communicate insights, patterns, and trends within data sets, m aking them easier to understand and interpret. Some common data visualization techniques in clude:

1. **Scatter Plots**: Used to visualize the relationship between two variables, scatter plots ar e particularly useful for identifying correlations or patterns within the data.
2. **Bar Charts**: Display data using rectangular bars with lengths proportional to the value s they represent. They are effective in comparing the frequency, count, or distribution of c ategorical data.
3. **Histograms**: Represent the distribution of continuous data by dividing it into intervals (bins) and displaying the frequency of observations within each bin.
4. **Line Charts:** Ideal for visualizing trends over time or other ordered categories, line ch arts connect data points with lines to illustrate changes or patterns.
5. **Heatmaps:** Use color gradients to represent data values in a matrix format. They are va luable for visualizing correlations and patterns in large datasets.
6. **Box Plots:** Display the distribution of numerical data through quartiles, highlighting th e median, upper, and lower quartiles, and any potential outliers.

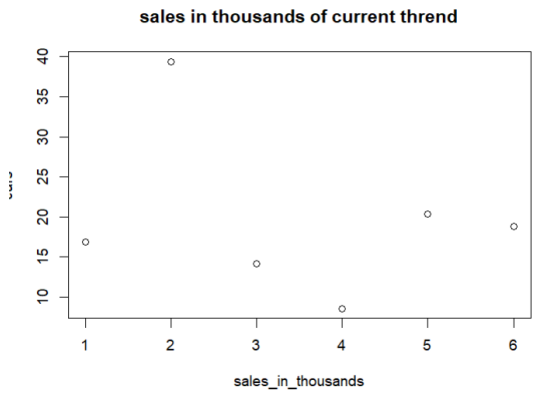
These visualization techniques help analysts, data scientists, and researchers explore, analyze, and communicate insights effectively from their datasets, facilitating informed decision-maki ng and understanding of complex data relationships. First install the package “ggplot2” using the install.packages( ) function.

• USING SCATTER PLOTS

The following code will create a scatter plot with the sales_in_thousands on the x-axis and cars on the y-axis using base R plotting functions

```
#scatter plot
plot(cars$Sales_in_thousands,
    xlab="sales_in_thousands",
    ylab="cars",
    main="sales in thousands of current threnD")
```

OUTPUT:

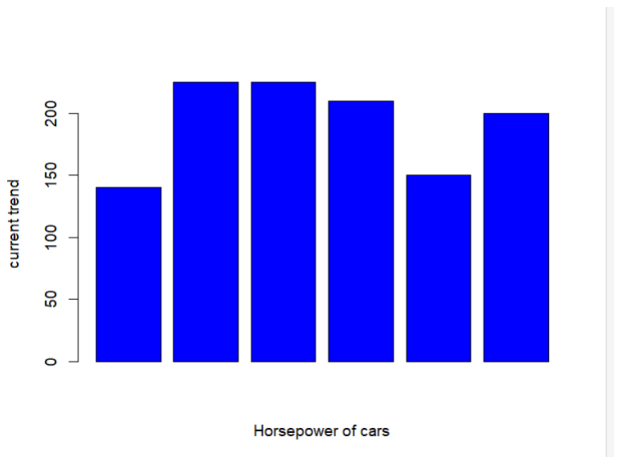


• USING BARCHARTS

The following code will generate a bar chart with thalach on the x-axis

```
#BAR PLOT
barplot(cars$Horsepower,xlab="Horsepower of cars",ylab="current trend",col="blue")
```

OUTPUT:



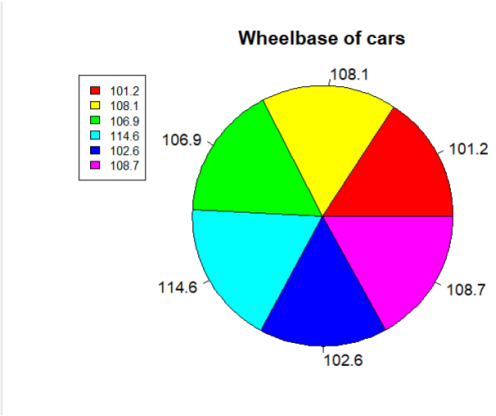
• USING PIECHART

The following code will generate a pie chat for the wheelbase based on the car

#piechart

```
pie(cars$Wheelbase,labels=cars$Wheelbase,radius = 1,main="Wheelbase of cars",col=rainbow(length(as.integer(cars$Wheelbase ))))
legend("topleft",legend=as.character(cars$Wheelbase),cex=0.8,fill=rainbow(length(as.integer(cars$Wheelbase))))
```

OUTPUT:

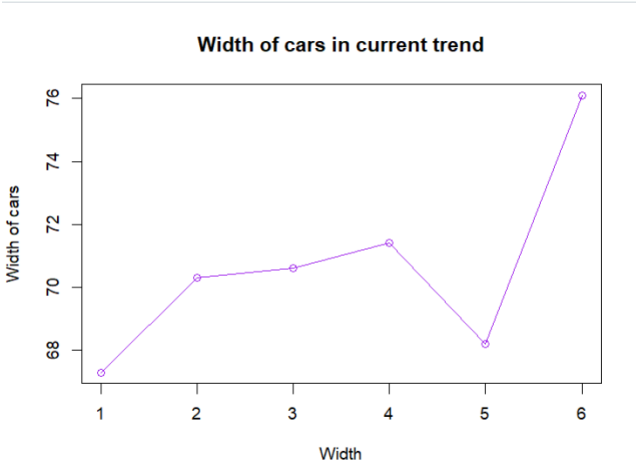


• USING LINECHART

The following code will generate a linechart for the width.

```
plot(cars$Width,type="o",xlab="Width",ylab="Width of cars",main="Width of cars in current trend",col="purple")
```

OUTPUT:



STATISTICALANALYSIS USING R

R offers a comprehensive suite of statistical analysis functions that empower users to conduct a wide range of analytical tasks. From basic descriptive statistics to advanced modelling techniques, R provides a rich ecosystem of packages and functions to perform statistical analysis efficiently. Users can compute measures such as mean, median, variance, and standard deviation using built-in functions like mean(), median(), var(), and sd(), respectively. Furthermore, R facilitates hypothesis testing with functions such as t.test() for comparing means between groups and chisq.test() for testing associations between categorical variables. Regression analysis, a cornerstone of statistical modelling, is made accessible through functions like lm() for linear regression and glm() for generalized linear models. Moreover, R supports exploratory data analysis with graphical functions like plot() and ggplot2, enabling users to visualize relationships, distributions, and trends in their data. Overall, R's statistical analysis functions empower users to explore, analyze, and interpret data effectively, making it a versatile tool for researchers, analysts, and data scientists.

➢ The summary() function in R offers essential descriptive statistics for data frames and model outputs, aiding in understanding data distributions and model performance. It's a vital tool for quick insights and diagnostics in R.

summary(cars)

OUTPUT:

```
> #summary
> summary(cars)
 Manufacturer          Model           Sales_in_thousands X__year_resale_value Vehicle_type
 Length:6           Length:6           Min.   : 8.588     Min.   :16.36        Length:6
 Class :character   Class :character   1st Qu.:14.815     1st Qu.:18.64        Class :character
 Mode  :character   Mode  :character   Median :17.849     Median :21.07        Mode  :character
                                       Mean   :19.697     Mean   :21.67
                                       3rd Qu.:19.993     3rd Qu.:23.23
                                       Max.   :39.384     Max.   :29.73

 Price_in_thousands  Engine_size      Horsepower      Wheelbase        Width           Length
 Min.   :21.50      Min.   :1.800   Min.   :140.0   Min.   :101.2   Min.   :67.30   Min.   :172.4
 1st Qu.:23.99      1st Qu.:2.050   1st Qu.:162.5   1st Qu.:103.7   1st Qu.:68.72   1st Qu.:181.5
 Median :28.40      Median :3.000   Median :205.0   Median :107.5   Median :70.45   Median :192.0
 Mean   :29.97      Mean   :2.717   Mean   :191.7   Mean   :107.0   Mean   :70.65   Mean   :187.3
 3rd Qu.:33.95      3rd Qu.:3.200   3rd Qu.:221.2   3rd Qu.:108.5   3rd Qu.:71.20   3rd Qu.:192.7
 Max.   :42.00      Max.   :3.500   Max.   :225.0   Max.   :114.6   Max.   :76.10   Max.   :196.6
 NA's   :1
  Curb_weight    Fuel_capacity   Fuel_efficiency Latest_Launch      Power_perf_factor      PC2
 Min.   :2.639   Min.   :13.20   Min.   :22.00   Length:6           Min.   :58.28     Min.   : NA
 1st Qu.:3.116   1st Qu.:16.60   1st Qu.:22.75   Class :character   1st Qu.:62.78     1st Qu.: NA
 Median :3.494   Median :17.20   Median :25.50   Mode  :character   Median :84.57     Median : NA
 Mean   :3.339   Mean   :16.75   Mean   :25.00                      Mean   :77.68     Mean   :NaN
 3rd Qu.:3.550   3rd Qu.:17.80   3rd Qu.:26.75                      3rd Qu.:91.37     3rd Qu.: NA
 Max.   :3.850   Max.   :18.50   Max.   :28.00                      Max.   :91.39     Max.   : NA
                                                                    NA's   :1         NA's   :6
>
```

➢ Calculating the summary of the correlation model of the

dataset Finding correlation between the Horsepower and width of

cars.

```
correlation <- cor(cars$Horsepower, cars$Width)
correlation
```

OUTPUT:

```
> correlation <- cor(cars$Horsepower, cars$Width)
> correlation
[1] 0.5546928
```

FUTURE WORK

1. Predictive Modeling: Develop predictive models to forecast future car sales based on historical data. This could involve using techniques such as time series analysis, regression analysis, or machine learning algorithms.
2. Market Segmentation: Segment the market based on various factors such as demographics, geographic location, vehicle type, or purchasing behavior. Analyze sales trends within each segment to identify opportunities for targeted marketing strategies.
3. Customer Profiling: Create customer profiles based on purchasing history, preferences, and demographics. Use these profiles to tailor marketing campaigns, product offerings, and sales strategies to specific customer segments.
4. Inventory Management: Optimize inventory management processes by analyzing sales data to identify fast-moving and slow-moving vehicles. Use this information to adjust inventory levels, pricing strategies, and promotional efforts accordingly.
5. Competitive Analysis: Conduct a competitive analysis to compare sales performance across different car manufacturers, models, and dealerships. Identify strengths, weaknesses, opportunities, and threats within the market landscape.

CONCLUSION

In conclusion, the analysis of the car sales dataset has provided valuable insights into the dynamics of the automotive market. Through thorough examination of sales trends, customer behavior, and market factors, several key findings have emerged:

Sales Trends: Overall, the dataset revealed fluctuations in car sales over time, influenced by various factors such as economic conditions, consumer preferences, and industry innovations. While some segments experienced growth, others faced challenges, highlighting the importance of adaptability and responsiveness to market changes.

Market Segmentation: The dataset enabled segmentation of the market based on factors such as vehicle type, geographic location, and demographic characteristics. Understanding these segments allowed for targeted marketing strategies and customized offerings to meet the diverse needs of consumers.

Customer Insights: By analyzing customer profiles and purchasing behavior, it became evident that consumer preferences varied widely across different segments. Factors such as brand loyalty, price sensitivity, and technological features played significant roles in influencing purchase decisions

Competitive Landscape: Through competitive analysis, we gained insights into the performance of various car manufacturers and models within the market. Identifying strengths and weaknesses relative to competitors provided opportunities for strategic positioning and differentiation.

Future Opportunities: Looking ahead, emerging trends such as electric vehicles, autonomous driving technology, and shared mobility services present both challenges and opportunities for the automotive industry. Leveraging data-driven insights will be crucial in adapting to these trends and capitalizing on new market opportunities.

Reference Material:

https://statisticsglobe.com/data-cleaning-í
https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R-Manual/R-Manual_píint.html
https://cían.í-píoject.oíg/web/packages/HSAUR/vignettes/Ch_intíoduction_to_R.pdf

APPENDIX

PROGRAM

```
install.packages("ggplot2")
library(ggplot2)
setwd("C:/Users/Leela/OneDrive/Desktop")
cars<-data.frame(read.csv("cars.csv"))
cars
#select first few rows
cars<-head(cars)
cars
#Data cleaning
is.na(cars)
complete.cases(cars)
clean<-na.omit(cars)
clean
#Imputing missing values
cars$PC2[is.na(cars$Horsepower)]<-mean(cars$Horsepower,na.rm=TRUE)
cars
#remove duplicate value
unique_cars<-unique(cars)
unique_cars
#detect and remove the outliers
cars$Horsepower[cars$Horsepower%in%boxplot.stats(cars$Horsepower)$out]
```

```
#Modify the column names
colnames(cars)[colnames(cars)=="price_in_thousands"]<-"price"
colnames(cars)
#remove spaces in character strings
Withoutspace<-gsub(" ","",cars)
Withoutspace
#data visualization
#Bar plot
barplot(cars$Horsepower,xlab="Horsepower of cars",ylab="current trend",col="blue")
 pie(cars$Wheelbase,labels=cars$Wheelbase,radius = 1,main="Wheelbase of
cars",col=rainbow(length(as.integer(cars$Wheelbase ))))
legend("topleft",legend=as.character(cars$Wheelbase),cex=0.8,fill=rainbow(length(as.integer(car
s$Wheelbase))))
#Scatter plot
plot(cars$Sales_in_thousands,
    xlab="sales_in_thousands",
    ylab="cars",
    main="sales in thousands of current threnD")
#Line chart
plot(cars$Width,type="o",xlab="Width",ylab="Width of cars",main="Width of cars in current
trend",col="purple")
#summary
summary(cars)
#reggresion
model<-lm(cars$Horsepower~cars$Width,cars=cars)
model
#correlation
correlation <- cor(cars$Horsepower, cars$Width)
correlation
```