NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

Department of Computer Science and Engineering

(B.E. Computer Science & Engineering Program Accredited by NBA from 2018-19 to 2021-25)

Report on Mini Project

Retail Marketing Analysis

Course Code: 21CSA31

Course Name: R Programming

Semester: III SEM Section: C

Submitted To:

Mr. Sunil Kumar Aithal
Assistant Professor [GD-II]
Department of Computer Science
and Engineering

Submitted By:

Pranitha P Shetty (4NM21CS111)
Priyal Mariam Cornelio (4NM21CS120)
Riya Dsa (4NM21CS126)
Rupali Behera (4NM21CS131)

Date of submission:
21 January 2023

Signature of Course Instructor

ABSTRACT

The project includes various types of data visualization and data analysis on Meesho order data and list of best-selling mobile phones. Data visualization is a technique used for graphical representation of data. Visualization of any data through charts and graphs gives a clear image and idea of the data set. When there is a large data set it becomes a difficult task to understand the data. Data visualization and analysis makes it easy to view and comprehend the patterns in data. It is more understandable and effective. Hence, we have come up with the following data visualization and data analysis techniques in this project.

- read.csv (): to import a CSV file into a Data Frame.
- Lollipop Plot: A correlation matrix is a simple table which displays the correlation coefficients for different variables.
- Calendar Plot: Calendar plot graphs are generated from a dive on data that is formatted as a date. They display all calendar days, including days that have no data associated with them. The calendar view with bar plots shows bars (or points) representing the relative size of summary data on each day of the calendar.
- Donut Plot: Donut Charts or Doughnut Charts are a special kind of Pie chart with the difference that it has a Blank Circle at the centre. The whole ring represents the data series taken into consideration.
- Histogram: Histogram is similar to a bar chart but the difference is that it groups the values into continuous ranges. Each bar in histogram represents the height of the number of values present in that range.
- Circular Packaged Plot: Circular packing or circular tree map allows to visualize a hierarchic organization. It is an equivalent of a tree map or a dendrogram, where each node of the tree is represented as a circle and its sub-nodes are represented as circles inside of it.
- Word Cloud: A word cloud (or tag cloud) is a visual representation of text data. Tags are usually single words, and the importance of each tag is shown with font size or colour. In R, two packages allow the creation of word clouds.
- Circular Bar Plot: A circular bar plot is a plot where each bar is displayed along a circle instead of a line. Here no Y scale is displayed since exact values are written on each bar which could also be represented on interaction with the chart.
- Bubble Plot: A lollipop plot is the combination of a line and a dot. It shows the relationship between a numeric and a categorical variable just like a bar plot.
- Stacked Bar Plot: The stacked bar chart (aka stacked bar graph) extends the standard bar chart from looking at numeric values across one categorical variable to two. Each bar in a standard bar chart is divided into a number of sub-bars stacked end to end, each one corresponding to a level of the second categorical variable.

TABLE OF CONTENTS

INTRODUCTION

R is a data science-oriented programming language that offers more than 18,000 data science packages. R allows us to perform feature selection, to perform all type machine learning and deep learning tasks, to apply various machine learning methods, such as classification, regression, clustering. They can be used to quickly identify outliers, compare different groups or variables, and communicate findings to others. Additionally, R has a wide variety of graphical packages and functions, such as ggplot2, lattice, and base R graphics, that makes it easy to create high-quality, customizable visualizations. R is widely used for data analysis, data visualization, and statistical modeling. R is open-source software, which means that it is free to use and distribute. It has a large and active community of users and developers, which contributes to its ongoing development and improvement. R is particularly useful for working with large and complex data sets, as well as for creating advanced visualizations and statistical models.

R contains functionality for many plot types including graphic maps, mosaic plots, biplots, and the list goes on. Probability distribution plays a vital role in statistics and by using R we can easily handle various types of probability distribution such as Binomial Distribution, Normal Distribution, Chi-squared Distribution and many more. It provides a large, coherent and integrated collection of tools for data analysis.

Retail analytics is a great tool to anticipate future demand and optimize the supply chain of vendors to meet the forthcoming surge. With the use of historical customer data, seasonality effects, market trends, and customer behaviour, one can build models that predict the stock needs for the near and far future.

In this project we are using two datasets called Meesho order data and bestselling mobile phones. The former is a dataset which includes attributes such as Reason for credit entry, Order date, Customer State, Product name, Size and Supplier Discounted Price and the latter is a collection of data that typically includes a number of different features that describe each mobile phone, such as Manufacturer, Model, Form Factor and Year.

In R, graphs are used to visualize data and make it easier to understand patterns, trends, and relationships within the data. They can be used to quickly identify outliers, compare different groups or variables, and communicate findings to others. Data visualization is an effective tool for understanding and analysing the complex relationships between these factors. In this analysis, we will explore different visualizations to classify different attributes and identify patterns in the data.

PROBLEM STATEMENT

Aim of this project is to perform data analysis on 'Retail data' using R programming and also to use various visualizing concepts to represent the contents of the dataset to help the manufacturer understand the needs of customer and their choices.

OBJECTIVES

- The objective of the Retail Marketing Analysis is to enable companies to create customer recommendations based on their purchase history, resulting in personalized shopping experiences and improved customer service.
- These super-sized data sets also help with forecasting trends and making strategic decisions based on market analysis.
- The aim of this data analysis is to identify patterns, trends, and relationships in the data, such as correlations between variables or clustering of observations.
- Large datasets can create problems in understanding the data. Also, the process of understanding the large dataset is time consuming and may not be accurate. So, to overcome these complications we have come up with different types of data analysis and data visualization.
- The goal of the project is to plot varieties of graphs along with its animations so as to make analysing of data simple and easy and also to create interactive visualizations that clearly communicates the key factors affecting the prices of the mobile phones and to identify any patterns or trends in the data.
- This involves using various tools and techniques such as circular packages, histograms, scatter plots, lollipop plots, etc. to create visual representations of the data, which can help identify patterns and trends that might not be immediately found from the raw data

METHODOLOGY

Following Packages are used in the analysis:-

1.ggplot: -The 'ggplot2' package in R Programming Language also termed as Grammar of Graphics is a free, open-source, and easy-to-use visualization package widely used in R. It includes several layers on which it is governed. The layers are as follows:

➢ Data: Dataset used to plot the graph.

➢ Aesthetics: The data is to map onto the Aesthetics attributes such as x-axis, y-axis, color, fill, size, labels, alpha, shape, line width, line type.

➢ Geometrics: How our data being displayed using point, line, histogram, bar plot, boxplot.

➢ Facets: It displays the subset of the data using Columns and rows.

➢ Statistics: Binning, smoothing, descriptive, intermediate.

➢ Coordinates: The space between data and display using Cartesian, fixed, polar, limits.

➢ Themes: Non-data link used to customize the appearance of the various elements of a plot, such as the background, axis labels, and legend.

The basic syntax of ggplot is:

ggplot(data, mapping=aes()) + geometric object

Here a geometric object is the type of plot you want to show.

The most common objects are: -

- Point: `geom_point () `
- Bar: `geom_bar () `
- Line: `geom_line () `
- Histogram: `geom_histogram () `

2.wordcloud: -Wordcloud package in R Programming Language helps us to know the frequency of a word in textual content using visualization. A word cloud is a visual representation of information or data. It shows the popularity of words or phrases by making the most frequently used words appear larger or bolder compared with the other words around them.

3.CalendR: -The calendR package in R Programming Language allows creating fully customizable ggplot2 calendar plots with a single function. In addition, the package provides arguments to create calendar heatmaps. It creates monthly and yearly calendars based on the 'ggplot2' package.

4.dplyr: -The dplyr package in R Programming Language is a structure of data manipulation that provides a uniform set of verbs, helping to resolve the most frequent data manipulation hurdles.
count () lets you quickly count the unique values of one or more variables:
syntax: -*count(dataframe_name,attribute_name)*
3.RColorBrewer: -RColorBrewer is an R package that contains ready-to-use color palettes for creating beautiful graphics. RColorBrewer can be used to create colourful graphs with premade palettes that consist of 8 to 12 colours.
4.tidyverse: -Tidyverse is an R programming package that helps to transform and better present data. It assists with data import, tidying, manipulation, and data visualization. The tidyverse package is open source, meaning that it is freely available to use and is constantly being modified and improved.
5.packcircles: -This package provides functions to find non-overlapping arrangements of circles.
The code snippet to load the packages is as follows: -

```
library(ggplot2)
library(wordcloud)
library(calendR)
library(dplyr)
library(RColorBrewer)
library(tidyverse)
library(packcircles)
```

Before starting with the analysis, set the working directory using the *getwd()* and *setwd()* functions so that we can access the file from the directory that contains the dataset.

Code Snippet: -

Reading the csv file: The code snippet used to read the csv file is : -

```
meesho2=read.csv("meesho2.csv")
mobile=read.csv("mobile.csv")
```

- The read.csv () function in R is used to read a csv file and create a data frame from it. The code reads a csv file named "meesho2.csv" and “mobile.csv”and creates data frames named "meesho2" and “mobile” with the data contained in the file.
- The png() function is used to create a png image of the given file name
- The dev.off() function shuts down the specified (by default the current) device.

1.Lollipop Plot: -

```
s=count(meesho2,`Customer.State`)
png(file="RLollipop_chart.jpg")
ggplot(s,aes(x=`Customer.State`,y=n))+geom_segment(aes(x=`Customer.S
tate`,xend=`Customer.State`,y=0,yend=n))+
```

```
geom_point(color=rainbow(length(s$n)),size=4)+
theme(axis.text.x=element_text(angle=90))
s=count(meesho2,`Customer.State`)
```

- The `count()` function is used to create a frequency table of the values in data frame.The above code gives the table that contains the frequency of each value in the column "Customer_State" of the data frame "meesho2". The result is stored in the variable ‘s’.

```
ggplot(s,aes(x=`Customer.State`,y=n))+geom_segment(aes(x=`Customer.S
tate`,xend=`Customer.State`,y=0,yend=n))+
geom_point(color=rainbow(length(s$n)),size=4)+
theme(axis.text.x=element_text(angle=90))
```

- The `ggplot()` function is used for creating plots using the ‘ggplot2’ R package. The first input argument defines the input data that is stored in the variable ‘s’. The `aes()` function allows aesthetic mapping of the input variables by defining the internal Customer_State to be displayed on the X axis its frequency to be displayed on the Y axis.
- The `geom_segment()` function is used to draw a straight line between points (x, y)
- The `geom_point()` function adds a layer of points to your plot
- `rainbow()` function in R Language is a built in color palettes which can be used to quickly generate color vectors of desired length taken as the parameter. Length(s$n) to plot s$n no. of different colours
- The `theme()` function is used to customize the appearance of a plot. The `theme()` function can be added to a ggplot plot to change various elements of the plot such as the background color, axis labels, and text size. Here the `theme()` is used to to rotate the x-axis text by an angle 90 degree.

2.Calendar Plot:-

```
c=count(meesho2,`Order.Date`)
png(file="RCalender.jpg")
calendR(year=2022,month=8,special.days=c$n,weeknames=c("Sun","Mon","
Tue","Wed","Thu","Sat"),special.col="purple",gradient=TRUE,
legend.pos="right",legend.title="Delivery Frequency")
dev.off()
```

```
c=count(meesho2,`Order.Date`)
```

- The above code gives the table that contains the frequency of each value in the

column "Oder.date" of the data frame "meesho2". The result is stored in the variable ‘c’.

```
calendR(year=2022,month=8,special.days=c$n,weeknames=c("Sun","Mon","
Tue","Wed","Thu","Sat"),special.col="purple",gradient=TRUE,
legend.pos="right",legend.title="Delivery Frequency")
```

- year :-Calendar year. By default uses the current year.
- month :-Month of the year or NULL (default) for the yearly calendar.
- special.days :-Numeric vector indicating the days to color or "weekend" for coloring all the weekends.
- Weeknames:- Character vector with the names of the days of the week starting on Monday. By default they will be in the system locale.
- special.col:- Color for the days indicated in special.days. If gradient = TRUE, is the higher color of the gradient.
- gradient:-Boolean. If special.days is a numeric vector of the length of the displayed days, gradient = TRUE creates a gradient of the special.col on the calendar.
- legend.pos If gradient = TRUE, is the position of the legend. It can be set to "none" (default), "top", "bottom", "left" and "right".
- legend.title If legend.pos != "none" and gradient = TRUE, is the title of the legend.

3.Donut Plot: -

```
del=count(meesho2,`Reason.for.Credit.Entry`)
df=data.frame(var1=c(del$`Reason.for.Credit.Entry`),Freq=c(del$n))
df$fraction=df$Freq/sum(df$Freq) df$ymax=cumsum(df$fraction)
df$ymin=c(0,head(df$ymax,n=-1))
ggplot(df,aes(ymax=ymax,ymin=ymin,xmax=4,xmin=3,fill=var1))
+geom_rect()+ coord_polar("y") + xlim(c(2,4))
png(file="RDonut_plot.jpg")
dev.off()

del=count(meesho2,`Reason.for.Credit.Entry`)
```

- The above code gives the table that contains the frequency of each value in the column "Reason.for.Credit.Entry" of the data frame "meesho2". The result is stored in the variable ‘del’.

```
df=data.frame(var1=c(del$`Reason.for.Credit.Entry`),Freq=c(del$n))
```

```
df$fraction=df$Freq/sum(df$Freq)
```

- Calculates percentages of column.

```
df$ymax=cumsum(df$fraction)
```

- Calculates cumulative percentages (top of each rectangle) column

```
df$ymin=c(0,head(df$ymax,n=-1))
```

- Calculates the bottom of each rectangle

```
ggplot(df,aes(ymax=ymax,ymin=ymin,xmax=4,xmin=3,fill=var1))
+geom_rect()+ coord_polar("y") + xlim(c(2,4))
```

- `geom_rect()` to plot each group as a rectangle and then converting it to a ring using `coord_polar()`.
- `coord_polar(theta="y")` is used for converting to polar coordinates (stacked rectangle to pie-chart)
- `xlim(c(2,4))` is used to convert Pie-chart to a ring

4.Histogram

```
h=count(meesho2,`Supplier.Discounted.Price`)
price=h$`Supplier.Discounted.Price`
png(file="RHistogram_plot.jpg")
hist(price,col=brewer.pal(n=length(h),name="RdPu"),main="Price
Histogram")
dev.off()

h=count(meesho2,`Supplier.Discounted.Price`)
```

- The above code gives the table that contains the frequency of each value in the column "Supplier.Discounted.Price" of the data frame "meesho2". The result is stored in the variable ‘h’.

```
price=h$`Supplier.Discounted.Price`
```

- The column of data frame ‘h’ is stored in variable ‘price’.

```
hist(price,col=brewer.pal(n=length(h),name="RdPu"),main="Price
Histogram")
```

- `hist()` is the generic function hist **computes a histogram of the given data values**.Here the given data value is ‘price’.
- `brewer.pal()` function displays the selected palette in a graphics window.

  Here ‘RdPu’ is one of the diverging palettes that are available in variations from 3 different values up to 11 different values.

- `main()` is used to main title to the graph

5.Circular Packaged Plot:-

```
png(file="RCircular_packaging.jpg")
size=count(meesho2,Size)
x = runif(min(size$n), max(size$n))
y = runif(min(size$n), max(size$n))
packing=circleProgressiveLayout(size$n, sizetype='area')
size$packing
dat.gg= circleLayoutVertices(packing, npoints=50) ggplot() +
geom_polygon(data = dat.gg, aes(x,y,group=id, fill=factor(id)) )+
geom_text(data = size, aes(x,y,label = Size)) + theme(legend.title =
element_blank(),legend.position = "right") +
scale_fill_discrete(labels=c(size$Size))+ coord_equal()
  dev.off()
```

```
size=count(meesho2,Size)
```

- The above code gives the table that contains the frequency of each value in the column "Size" of the data frame "meesho2". The result is stored in the variable ‘size’.

```
x = runif(min(size$n), max(size$n))
y = runif(min(size$n), max(size$n))
```

- The `runif()` function generates random deviates of the uniform distribution.

```
packing=circleProgressiveLayout(size$n, sizetype='area')
```

- Generates the layout.This function returns a data frame with one line per bubble.

```
dat.gg= circleLayoutVertices(packing, npoints=50)
```

- It generates a set of circle vertices suitable for plotting.

```
ggplot() + geom_polygon(data = dat.gg, aes(x,y,group=id,
fill=factor(id)) )+ geom_text(data = size, aes(x,y,label = Size)) +
theme(legend.title = element_blank(),legend.position = "right") +
scale_fill_discrete(labels=c(size$Size))+ coord_equal()
```

- `geom_polygon()` draws polygons, which are filled paths. Each vertex of the polygon requires a separate row in the data. It is often useful to merge a data frame of polygon coordinates with the data just prior to plotting.

- `geom_text()` are useful for labeling plots..
- Scale fill discrete A function that returns a discrete color/fill scale.
- `coord_equal()` function ensures the units are equally scaled on the x-axis and on the y-axis.

6.Word Cloud

```
m2=mobile$Manufacturer
png(file="RWordcloud.jpg")
wordcloud(words=m2,min.freq=1,random.order=FALSE,random.color=TRUE,c
olors=brewer.pal(8,"D ark2"))
dev.off()
```

```
m2=mobile$Manufacturer
```

- The above code is used to store the names of manufacturers of mobiles in the variable ‘m2’.

```
wordcloud(words=m2,min.freq=1,random.order=FALSE,random.color=TRUE,
colors=brewer.pal(8,"Dark2"))
```

- `word` gives the words.
- `min.freq` -words with frequency below min.freq will not be plotted.
- `random.order` plot words in random order. If false, they will be plotted in decreasing frequency.
- `random.color` choose colors randomly from the colors. If false, the color is chosen based on the frequency.
- `Colors-` color words from least to most frequent.

7. Circular Bar Plot

```
m3=count(mobile,Manufacturer)
png(file="RCircular_bar.jpg")
ggplot(m3,aes(x=Manufacturer,y=n,fill=Manufacturer))+geom_bar(stat="
identity",fill="darkred")+co ord_polar(start=0)+ theme_minimal()
dev.off( )
```

```
m3=count(mobile,Manufacturer)
```

- The above code gives the data frame that contains the frequency of each value in the column "Manufacturer" of the data frame "mobile". The result is stored in the variable ‘m3’.

```
ggplot(m3,aes(x=Manufacturer,y=n,fill=Manufacturer))+geom_bar(stat="
```

```
identity",fill="darkred")+coord_polar(start=0)+ theme_minimal()
```

- The `aes()` function allows aesthetic mapping of the input variables by defining the 'Manufacturer' to be displayed on the X axis its frequency to be displayed on the Y axis.
- `geom_bar()` makes the height of the bar proportional to the number of cases in each group (or if the weight aesthetic is supplied, the sum of the weights).
- `Sate=identity`-this indicates that R should use the y-value given in the ggplot() function.
- `coord_polar(start=0)` –to make axis circular.
- `theme_minimal()` : A minimalistic theme with no background annotations.

8.Bubble Plot:-

```
m=count(meesho2,Size)
a=m$`Size`
b=m$`n`
color <- brewer.pal(n = length(a), name = "PuOr")
size=c(m$n)
png(file="RBubble_chart.jpg")
ggplot(data=m,aes(a,b,size=size))+geom_point(alpha=0.7,color=
rainbow(length(b)))+ theme(axis.text.x=element_text(angle=90)) +
scale_size_continuous(range = c(8, 13))
dev.off()

m=count(mobile,Year)
```

- The above code gives the data frame that contains the frequency of each value in the column "Year" of the data frame "mobile". The result is stored in the variable 'm'.

```
a=m$`Year`
b=m$`n`
```

- The values of column "Year" and column "n" of data frame 'm' are stored in variables 'a' and 'b' respectively.

```
color <- brewer.pal(n = length(a), name = "PuOr")
```

- `brewer.pal()` function displays the selected palette in a graphics window.

  Here 'PuOr' is one of the diverging palettes that are available in variations from 3 different values up to 11 different values.

```
size=c(m$n)
```

- The above code is used to store a vector of frequency numbers in variable 'size'.

```
ggplot(data=m,aes(a,b,size=size))+geom_point(alpha=0.7,color=
rainbow(length(b)))+ theme(axis.text.x=element_text(angle=90)) +
scale_size_continuous(range = c(8, 13))
```

- The `geom_point()` function adds a layer of points to your plot .
- `rainbow()` function in R Language is a built in color palettes which can be used to quickly generate color vectors of desired length taken as the parameter.
- The `theme`() function is used to customize the appearance of a plot. The `theme`() function can be added to a ggplot plot to change various elements of the plot such as the background color, axis labels, and text size. Here the `theme`() is used to rotate the x-axis text by an angle 90 degree.
- `scale_size()` continuous scales size.The size aesthetic is most commonly used for points and text, and humans perceive the area of points (not their radius), so this provides for optimal perception. range() -a numeric vector of length 2 that specifies the minimum and maximum size of the plotting symbol after transformation.

9.Stacked Bar Plot

```
png(file="RStacked_plot.jpg")
ggplot(mobile,aes(x = Year, fill = Form.factor)) + geom_bar(width =
0.4) + theme_classic() + theme( plot.title = element_text(family =
"Times New Roman", hjust ="bold"), axis.text = element_text(family =
"Times New Roman",face ="bold"), axis.title = element_text(family =
"Times New Roman", face = "bold"), legend.title = element_blank(),
legend.text = element_text(family = "Times New Roman") ) +
labs(title = "Types of Phones", x = NULL, y = "Mobile count")
dev.off()
```

- The `aes()` function allows aesthetic mapping of the input variables by defining the internal Customer_State to be displayed on the X axis its frequency to be displayed on the Y axis.
- `geom_bar()` makes the height of the bar proportional to the number of cases in each group (or if the weight aesthetic is supplied, the sum of the weights).
- `theme_classic()` - A classic-looking theme, with x and y axis lines and no gridlines.
- `plot.title`–to add title to the plot

- In conjunction with the theme system, the element_ functions specify the display of how non-data components of the plot are drawn.

element_text()-to add text.

- family-to specify font style of each text.
- axis.text-to add axis text.
- axis.title-to give title to the axis.
- legend.title-to give title to the legend.

IMPLEMENTATION

```
#Required library
library(ggplot2)
library(wordcloud)
library(calendR)
library(dplyr)
library(RColorBrewer)
library(tidyverse)
library(packcircles)

#Reading files
meesho2=read.csv("meesho2.csv")
mobile=read.csv("mobile.csv")

#Lollipop chart for meesho data
s=count(meesho2,`Customer.State`)
png(file="RLollipop_chart.jpg")
ggplot(s,aes(x=`Customer.State`,y=n))+geom_segment(aes(x=`Customer.S
tate`,xend=`Customer.State`,y=0,yend=n))+
geom_point(color=rainbow(length(s$n)),size=4)+
theme(axis.text.x=element_text(angle=90))
dev.off()

#Calendar plot for items bought on meesho in the month August 2022
c=count(meesho2,`Order.Date`)
png(file="RCalender.jpg")
calendR(year=2022,month=8,special.days=c$n,weeknames=c("Sun","Mon",
"Tue","Wed","Thu","Sat"),special.col="purple",gradient=TRUE,
legend.pos="right",legend.title="Delivery Frequency")
dev.off()
#Donut plot for delivery status
del=count(meesho2,`Reason.for.Credit.Entry`)
df=data.frame(var1=c(del$`Reason.for.Credit.Entry`),Freq=c(del$n))
df$fraction=df$Freq/sum(df$Freq) df$ymax=cumsum(df$fraction)
df$ymin=c(0,head(df$ymax,n=-1))
png(file="RDonut_plot.jpg")
ggplot(df,aes(ymax=ymax,ymin=ymin,xmax=4,xmin=3,fill=var1))
+geom_rect()+ coord_polar("y") + xlim(c(2,4))
dev.off()
```

#Histogram plot for prices of dresses

```
h=count(meesho2,`Supplier.Discounted.Price`)
price=h$`Supplier.Discounted.Price`
png(file="RHistogram_plot.jpg")
hist(price,col=brewer.pal(n=length(h),name="RdPu"),main="Price
Histogram")
dev.off()
```

#Circular packaged plot for different sizes of dresses bought

```
png(file="RCircular_packaging.jpg")
size=count(meesho2,Size)
x = runif(min(size$n), max(size$n))
y = runif(min(size$n), max(size$n))
packing=circleProgressiveLayout(size$n, sizetype='area')
size$packing
dat.gg= circleLayoutVertices(packing, npoints=50) ggplot() +
geom_polygon(data = dat.gg, aes(x,y,group=id, fill=factor(id)) )+
geom_text(data = size, aes(x,y,label = Size)) + theme(legend.title
= element_blank(),legend.position = "right") +
scale_fill_discrete(labels=c(size$Size))+ coord_equal()
dev.off()
```

#Word cloud for companies of mobile

```
m2=mobile$Manufacturer
png(file="RWordcloud.jpg")
wordcloud(words=m2,min.freq=1,random.order=FALSE,random.color=TRUE,
colors=brewer.pal(8,"D ark2"))
dev.off()
```

#Circular barplot for Companies of mobile

```
m3=count(mobile,Manufacturer)
png(file="RCircular_bar.jpg")
ggplot(m3,aes(x=Manufacturer,y=n,fill=Manufacturer))+geom_bar(stat=
"identity",fill="darkred")+co ord_polar(start=0)+ theme_minimal()
dev.off( )
```

#Bubble plot for sizes of clothes bought

```
m=count(mobile,Year)
a=m$`Year`
b=m$`n`
color <- brewer.pal(n = length(a), name = "PuOr")
```

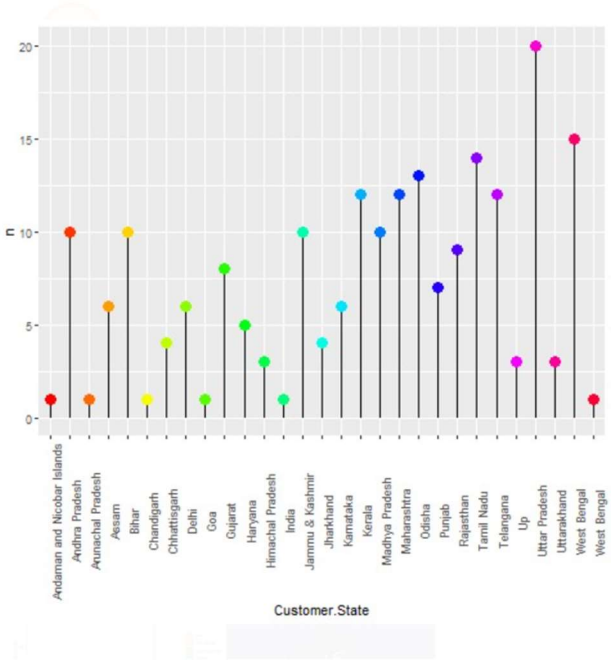
```
size=c(m$n)
png(file="RBubble_chart.jpg")
ggplot(data=m,aes(a,b,size=size))+geom_point(alpha=0.7,color=
rainbow(length(b)))+ theme(axis.text.x=element_text(angle=90)) +
scale_size_continuous(range = c(8, 13))
dev.off()
```

#Stacked bar plot for types of Mobile Phones

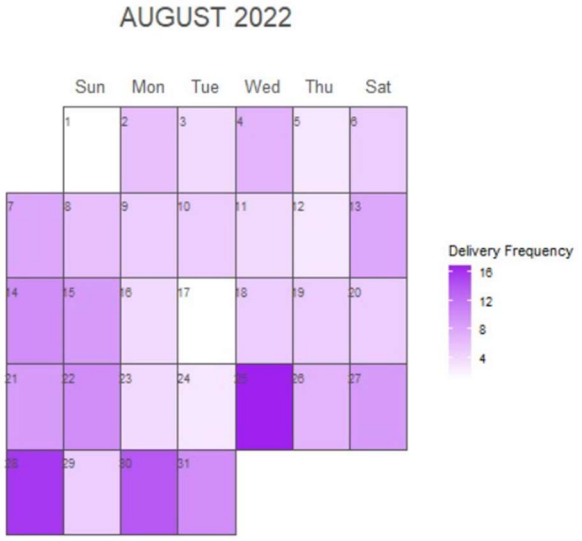
```
png(file="RStacked_plot.jpg")
ggplot(mobile,aes(x = Year, fill = Form.factor)) + geom_bar(width =
0.4) + theme_classic() + theme( plot.title = element_text(family =
"Times New Roman", hjust ="bold"), axis.text = element_text(family =
"Times New Roman",face ="bold"), axis.title = element_text(family =
"Times New Roman", face = "bold"), legend.title = element_blank(),
legend.text = element_text(family = "Times New Roman") ) +
labs(title = "Types of Phones", x = NULL, y = "Mobile count")
dev.off()
```

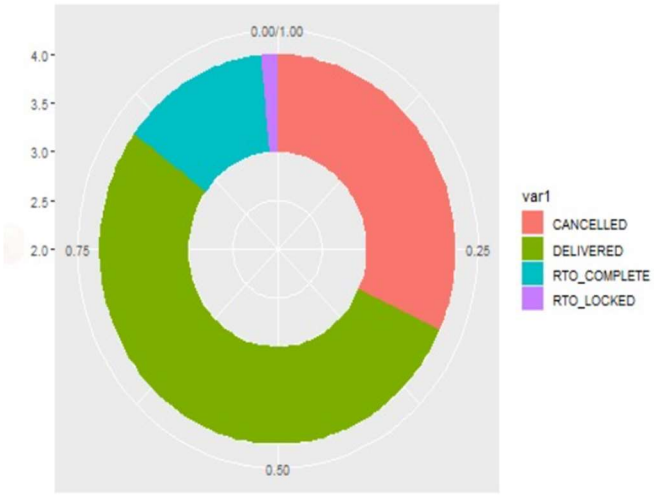
RESULTS AND DISCUSSIONS

Lollipop chart: In lollipop chart here we show the quantity of clothes each state has bought. 'geom_segment' is used to give the dimensions of line segment and 'geom_point' is used to give the dimensions of the point on top of the line segment which makes it look like a lollipop. Here from the figure we can easily deduce that the state of Uttar Pradesh buys clothes from Meesho the most in the month of August 2022.

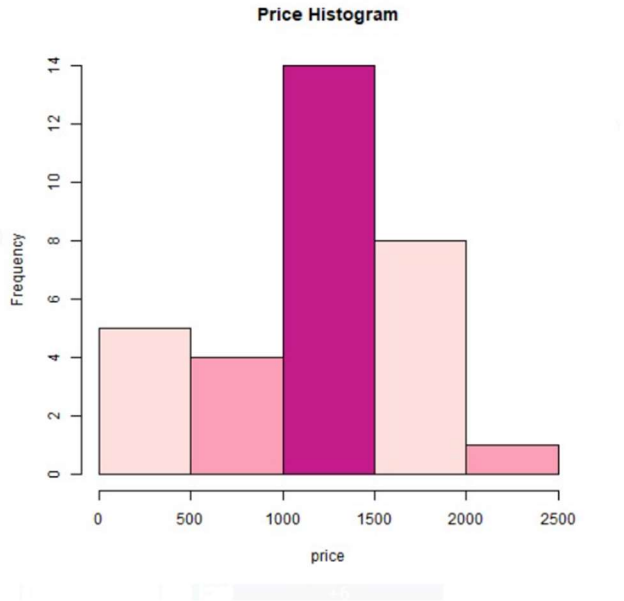


Calendar Plot: In calendar plot we use the calendR() function, where we enter the year and month of calendar we want to create, we then enter the data. Here, we have used it figure out in which day of the month in August 2022 most number of orders are ordered in Meesho. We find that it is mostly on the 25th, followed by 28th and 30th .



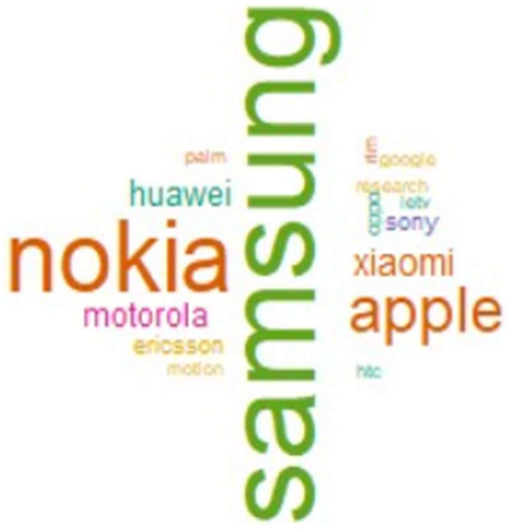


Donut plot: We have used donut plot here to show the state of delivery of the order. Here there are 4 possibilities order is cancelled, delivered, RTO_Completed(Given address for the order is wrong) and RTO_Locked(Receiver is not present to receive package). Here we first covert the total of each state of delivery to its fraction and store it in data frame df. We then convert 4 of these states into a single stacked bar, whict is then converted into a donut by joining ends using coord_polar() function.This allows us to visualize the states of delivery.

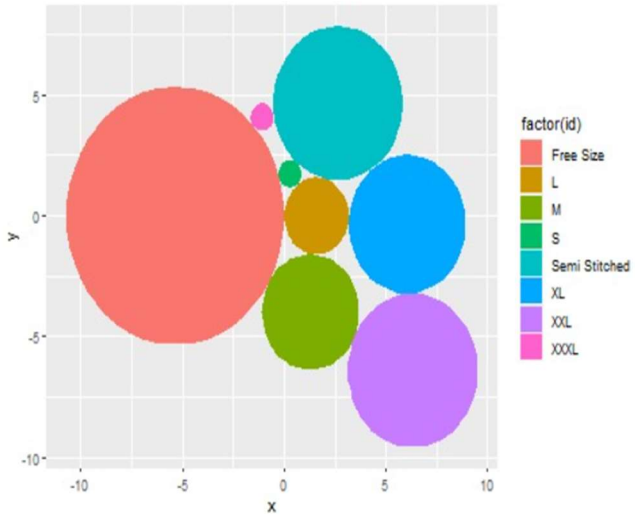


Histogram Graph: Here we have plotted a histogram graph based on the prices of the clothes people have bought on Meesho, using the function hist(). This graph is done for the manufacturers the figure out the price range of most customers and in from the visuals of this graph we can see that most orders are in the price range of Rs.1000-Rs.1500, which means that this range belongs to the price range which most customers prefer.

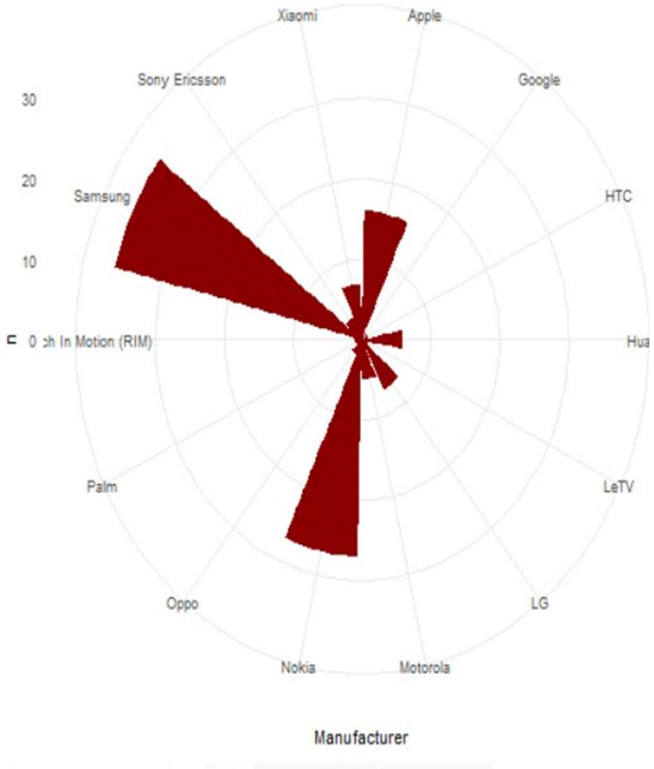
Word cloud: A word cloud is an visual representation of text data, where tags are usually single words and the importance of each tag is shown with font size or color. Here we have represented the various companies of mobile phone bought by people using word cloud.Each word in the word cloud has differentiating size which represents the frequency at which the mobile phone of each company is bought. Here we can see that Samsung is the most preferred company of customers

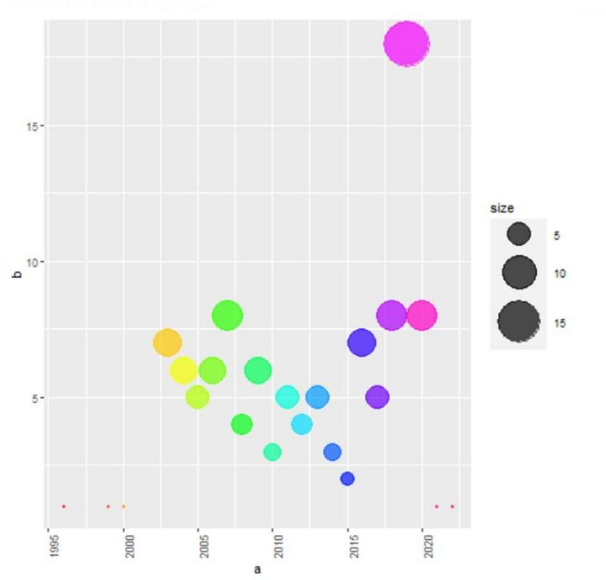


Circular Packaged Plot: This plot refers to a series of circles which are not interconnected with each other. Here we have given the size of circles as the frequency of the size of clothes people have ordered from Meesho. We use the function circleProgressiveLayout() to prevent the circles from interconnecting and coord_equal() makes the oval to have equal x and y co-ordinates to make it into a circle. This is an attractive way to display a visual which shows what size or type of clothes people prefer. Here we find that even though people buy all sizes of clothes available but customers prefer the clothes of 'Free size' rather than a customized size.

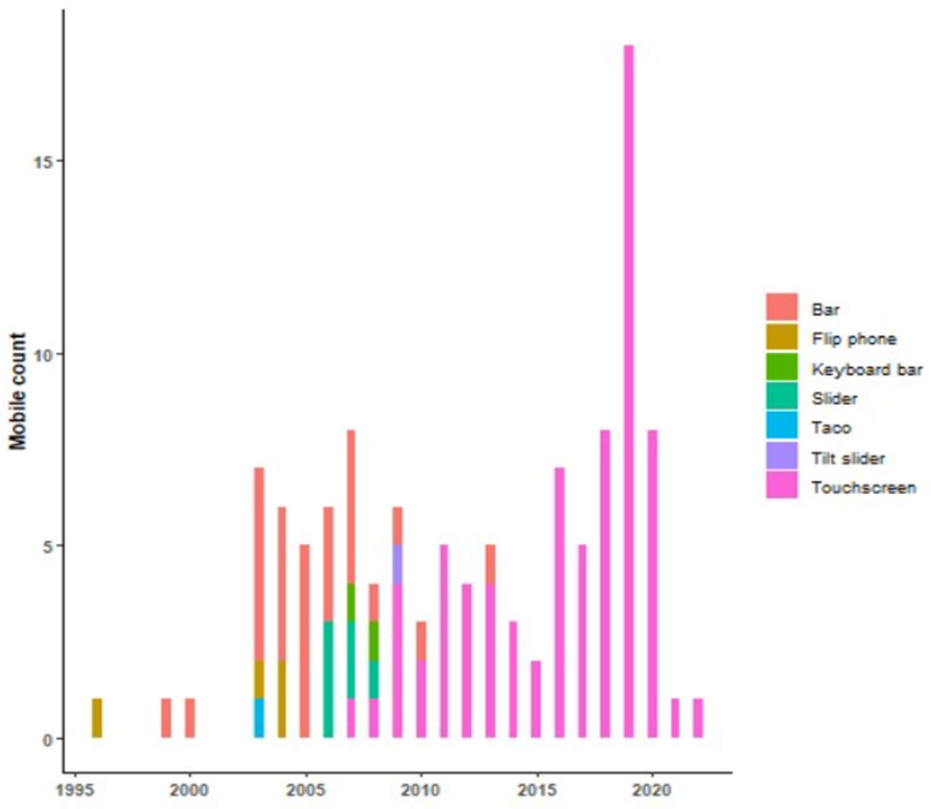


Circular bar chart: Circular bar chart is a typical bar chart displayed on a polar coordinate system, instead of a cartesian system using the function coord_polar(). Here we use the data of frequency of mobile phones bought belonging to a company. This circular bar chart displays the high order of 'Samsung' company mobile phone compared to other companies followed by 'Nokia'.





Bubble plot: A bubble plot is a data visualization that helps to displays multiple circles (bubbles) in a two-dimensional plot as same in a scatter plot. With ggplot2, bubble plots can be built using geom_point() function. Here we display the frequency of mobile phones bought in a year using this plot, where the bubble which is visually the largest has the corresponding year which sold the most phones. From the figure we can find that even though a lot of mobile phones have been bought over the years, in the year 2019 most number of mobile phones were sold.



Stacked bar chart: Each bar in a standard bar chart is divided into a number of sub-bars stacked end to end, each one corresponding to a level of the second categorical variable and is called stacked bar chart. Using stacked bar chart we have displayed the types of phones bought over the years and we compare the type of phones bought in recent years to that of before those years. From the chart we find that before the year of 2010 there was a wide variety of phones bought which include tilt slider, slider, keyboard bar etc. But after 2010 there's a huge increase in the rate of touchscreen phones bought. Then we see that from the year 2014 people have switched to buying only touchscreen phone.

CONCLUSION AND FUTURE SCOPE

Conclusion: -The different ways of data analysis and data visualization is done and we can conclude that graphs make the understanding easier and effective. Retail Analysis of dresses based on their sizes, discounted rate etc and mobile phones based on their company, characteristics and features make the customer better experience and can help companies more accurately analyse their data and costs, as well as recognize spending trends. After accurately identifying this information, businesses can extrapolate insights about potential future
costs
or cost-saving techniques to limit spending and cut waste based on which the graphs are plotted. These graphs have high accuracy and hence it gives clear and accurate information to the readers. The plotting of graphs has helped in following the trends and patterns in mobile price classification. Comparing various features becomes easier using the plots.

Future Work: - R is a powerful programming language that is widely used in many different fields. As the language continues to evolve and new tools and technologies become available, it's possible to analyze and visualize data in various ways to get an effective result. Some of them are mentioned below.

• Ensemble methods: These methods combine multiple models to improve the overall performance, for example, Random Forest, Gradient Boosting, and Ad boost are ensemble methods that can be used for mobile price classification.

• Transfer learning: This technique uses a pre-trained model to extract features from the mobile phone data, which can then be used to train a new model for mobile price classification. This can be useful when there is limited data available for training.

• Multi-task learning: This approach trains a model to perform multiple tasks simultaneously, such as mobile price classification and image recognition. This can be useful when the mobile phone data includes images of the devices.

• Anomaly detection: This type of analysis is used to identify and handle outliers in the mobile phone data, which can be useful when the dataset contains noise or inconsistencies.

• Time series analysis: This type of analysis can be used to analyse the price trends of mobile phones over time, which can be useful for forecasting future prices.

REFERENCES

[1] https://www.kaggle.com/datasets/sahilr05/meesho-orders
[2] https://www.kaggle.com/datasets/devrimtuner/top-100list-of-bestselling-mobile-phones
[3] https://towardsdatascience.com/how-to-create-animated-plots-in-r-adf53a775961
[4] https://rpubs.com/
[5] https://r-graph-gallery.com/