Project - Cardio Good Fitness

Model Report

Table of Contents

1. Project Objective

The objective of the report is to explore the cardio data set ("CardioGoodFitness") in R and generate insights about the data set. This exploration report will consists of the following:

- Importing the dataset in R
- Understanding the structure of dataset
- Graphical exploration
- Descriptive statistics
- Insights from the dataset

2. Assumptions

We Assume that the products sells is impacted based on more than variable (Income, Gender, Education, Fitness and Miles).

In addition, we will try to analysis the relation between (Income, Age, Education, Miles, and Usage) to see if there is a possible promotion targeted group.

3. Exploratory Data Analysis – Step by step approach

A Typical Data exploration activity consists of the following steps:

1. Environment Set up and Data Import
2. Variable Identification
3. Univariate Analysis
4. Bi-Variate Analysis
5. Missing Value Treatment (Not in scope for our project)
6. Outlier Treatment (Not in scope for our project)
7. Variable Transformation / Feature Creation
8. Feature Exploration

We shall follow these steps in exploring the provided dataset.

Although Steps 5 and 6 are not in scope for this project, a brief about these steps (and other steps as well) is given, as these are important steps for Data Exploration journey.

3.1 Environment Set up and Data Import

3.1.1 Install necessary Packages and Invoke Libraries

These are the packages we need:

- readr : to read file of type "csv".
- ggplot2: for protting.
- plyr: the split-apply-combine paradigm for R
- scales: Graphical scales map data to aesthetics, and provide methods for automatically determining breaks and labels for axes and legends

Please refer Appendix A for Source Code.

3.1.2 Set up working Directory

Setting a working directory on starting of the R session makes importing and exporting data files and code files easier. Working directory is the location/ folder on the PC where you have the data, codes etc. related to the project.

Please refer Appendix A for Source Code.

3.1.3 Import and Read the Dataset

The given dataset is in .csv format. Hence, the command 'read.csv' is used for importing the file.

Please refer Appendix A for Source Code.

3.2 Variable Identification

- attach: Used for objects in the dataset can be accessed by simply giving their names.
- dim: Retrieve or set the dimension of an object.
- names: Get or set the names of an object.
- str: Display the internal structure of a dataset.
- head: Returns the first 6 rows of the dataset.
- tail: Returns the last 6 rows of the dataset.
- summary: Return a summary of the dataset variables.
- anyNA: NA is a logical constant of length 1 which contains a missing value indicator.
-

Please refer Appendix A for Source Code.

3.2.1 Variable Identification – Inferences

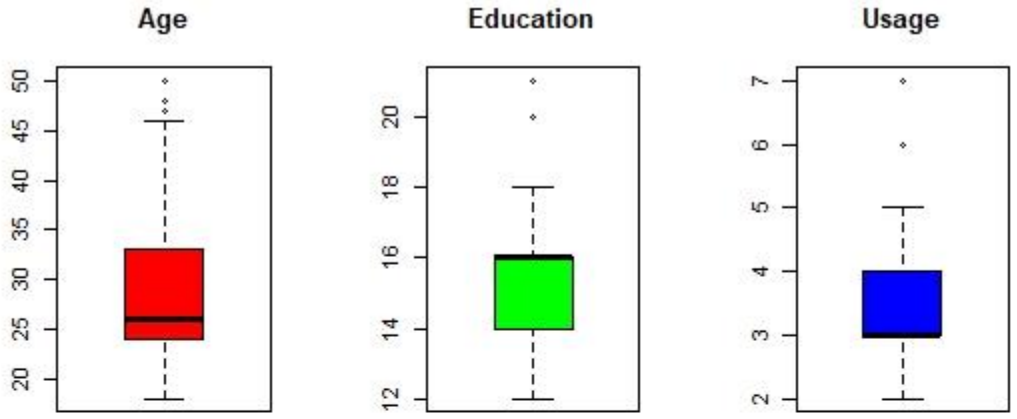
- dim: we see that we have 180 different observation in 9 variables.
- names: we see that all the names looking good and straightforward to work with.
- str: we identifying that:
    - Product is a Factor of 3 levels.
    - Age is int.
    - Gender is a Factor of 2 levels.
    - Education is int.
    - MaritalStatus is a Factor of 2 levels.
    - Usage is int.
    - Fitness is int.
    - Income is int.
    - Miles is int.
- head & tail: shows that we are lucky we have quite bet a clear data.
- summary: we see that we have 3 Qualitative variable which is Product, Gender and MaritalStatus, while the fourth one which is Fitness, is considered as Quantitative, and we think that we need to change it to be Qualitative as its mentioned in the question and this is make a sense.
  It will be like "very unfit", "unfit", "normal", "fit", "very fit".
- anyNA: we see that we don't have missing value at whole dataset.

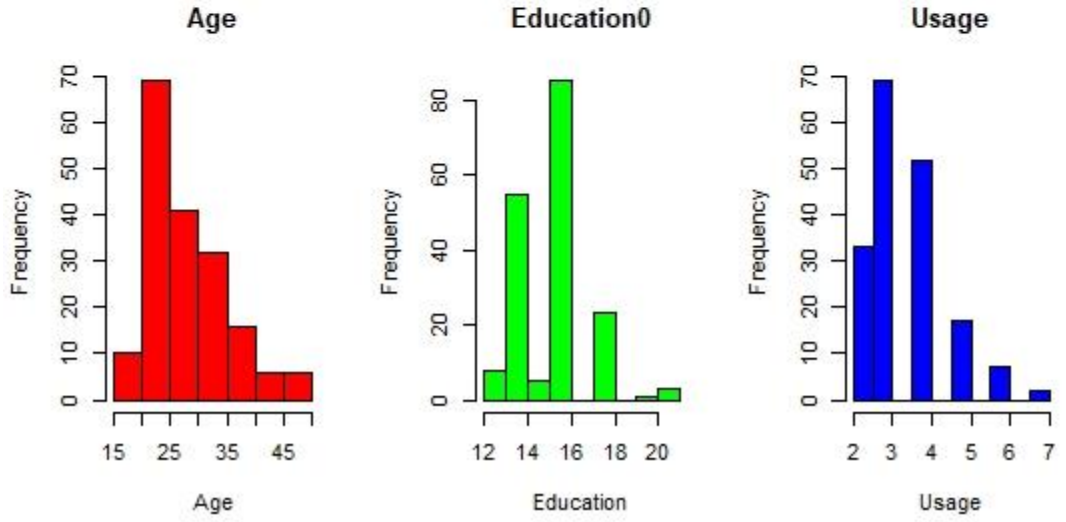
Please refer Appendix A for Source Code.

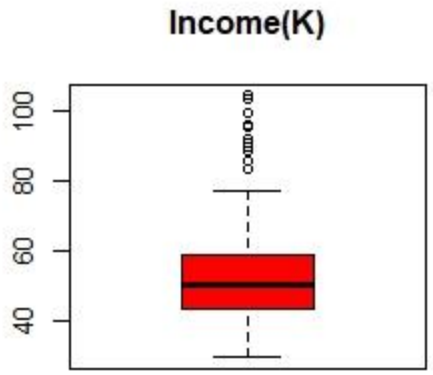
3.3 Univariate Analysis

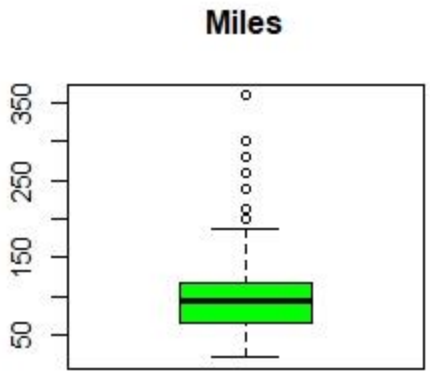
This table shows Univariate Analysis for all variables

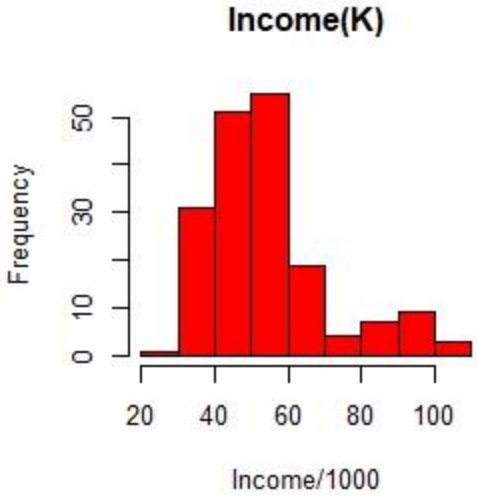
Variable	Mean	Mode	Median	Q1	Q3	IQR	Min	Max
Product	NA	TM195	NA	NA	NA	NA	NA	NA
Age	28.78889	25	26	24	33	9	18	50
Gender	NA	Male	NA	NA	NA	NA	NA	NA
Education	15.57222	16	16	14	16	2	12	21
MaritalStatus	NA	Partnered	NA	NA	NA	NA	NA	NA
Usage	3.455556	3	3	3	4	1	2	7
Fitness	NA	normal	NA	NA	NA	NA	NA	NA
Income	53719.58	45480	50596.5	43206	58516	15310	29562	104581
Miles	103.1944	85	94	66	133	67	21	360

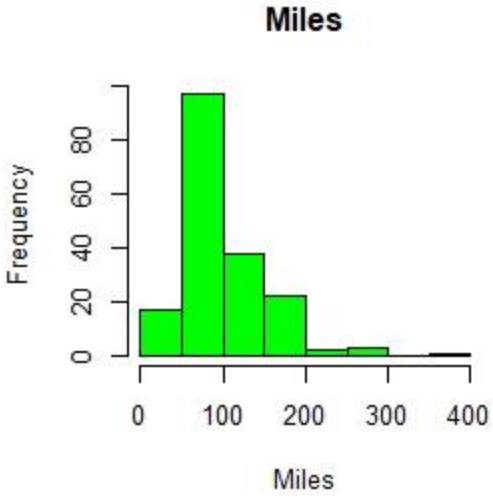




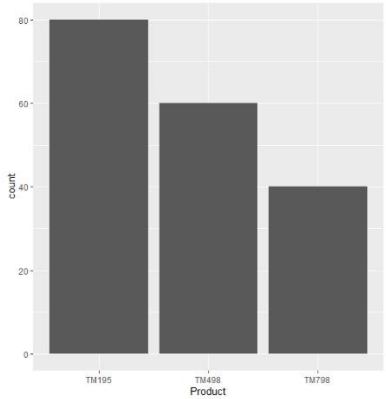


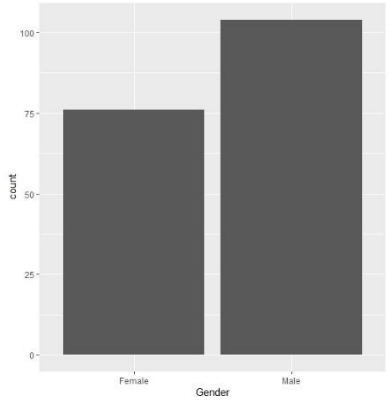


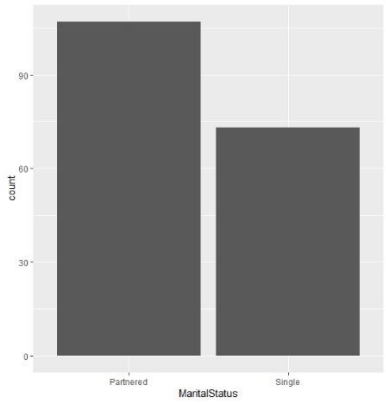


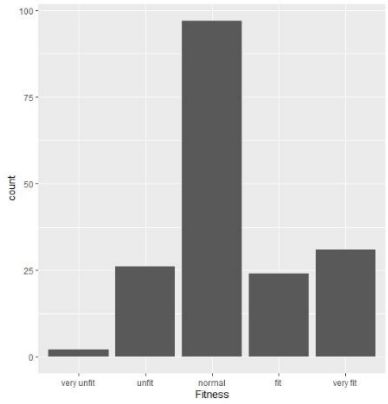


Histogram and Box Plot for Quantitative Variables









Categorical Variables

We see from above data the following:

- Most of the customers are between 24 – 33 years old.
- Most of the customer are Males.
- Most of the customer are between 14 – 16 years of education.
- Most of the customer are partnered.
- Between 3 – 4 is the most Avg. # time's customer wants to use the treadmill every week.
- Normal (3) is the most self-rated fitness score of the customer.
- Between 43 K – 58 K is the most Income of the customers.
- Most of the customer expect to run between 66 – 133 Miles.

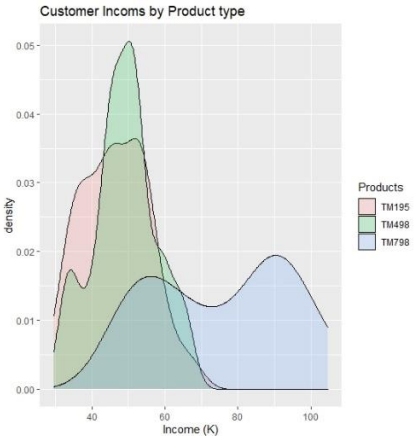
Please refer Appendix A for Source Code.

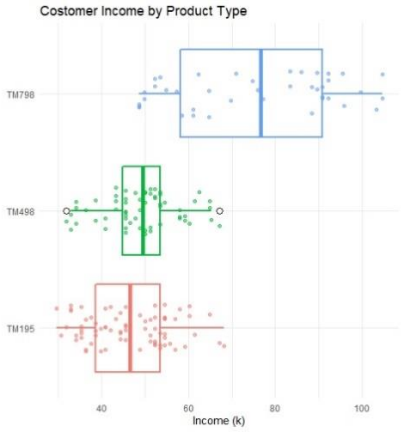
3.4 Bi-Variate Analysis

As we assume that Products sells effected by these variables (Income, Gender, Education, Fitness and Miles), so we are going to test that through geom_density and geom_boxplot for Quantitative variable and geom_bar for Categorical variable.

In addition, we will use scatter plot to try to analysis the relation between (Income, Age, Education, Miles, Usage, Fitness) to see if there is a possible promotion targeted group.

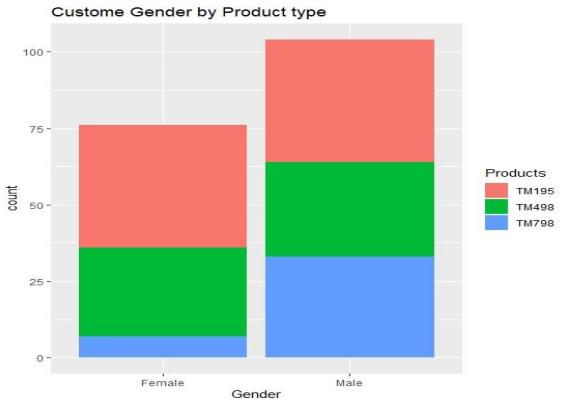
- Product vs Income





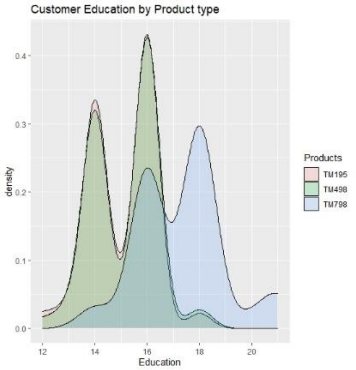
We can see clearly her customer with highest income are interested to bay the TM798 Product.

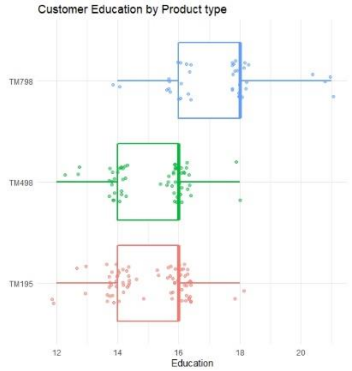
- Product vs Gender



From this, we see that the most of TM798 customer are Males.

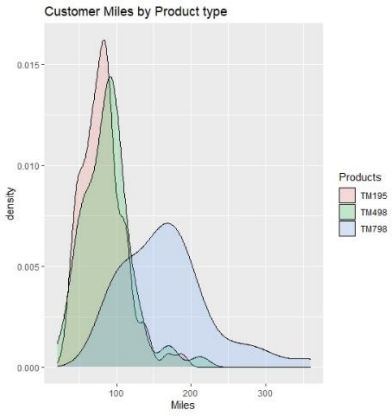
- Product vs Education

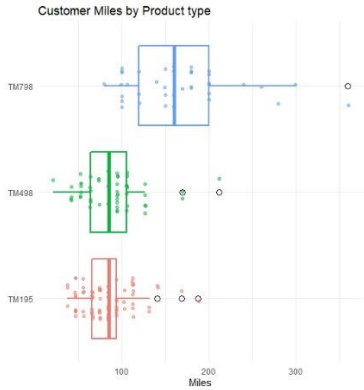




We can see the customer with higher education are interested in TM798 product

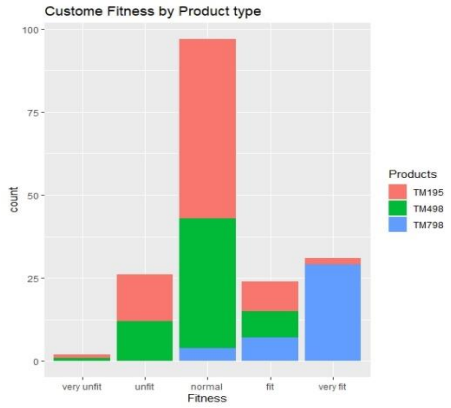
- Product vs Miles





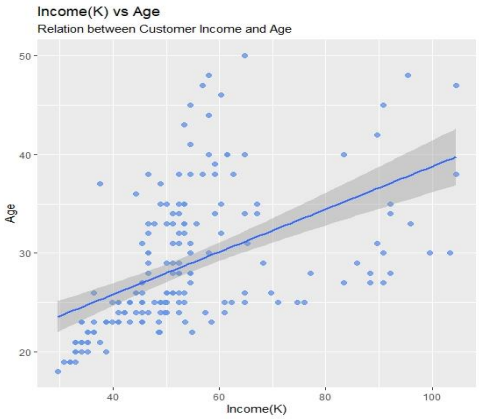
We can see the customer who has bought TM798 they are expect to use it more.

- Product vs Fitness



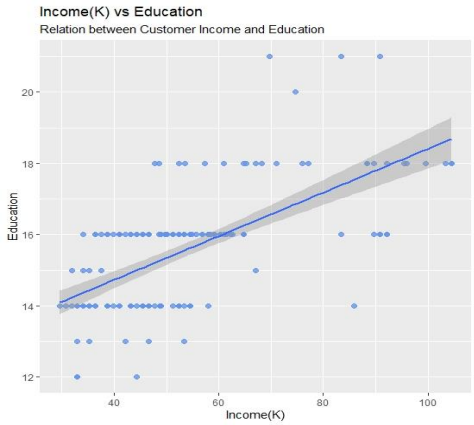
We can see the customers who has related themselves as very fit they are the most customers who has bought TM798.

- Income vs Age



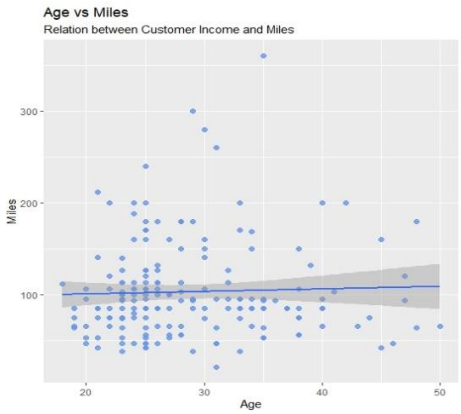
We can see relationship between two variables it goes up means when customer Age becomes older his Income increase.

- Income vs Education



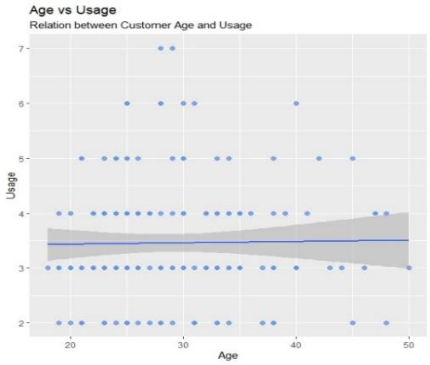
We can see relationship between two variables it goes up means when customer Education becomes Higher his Income increase.

- Age vs Miles



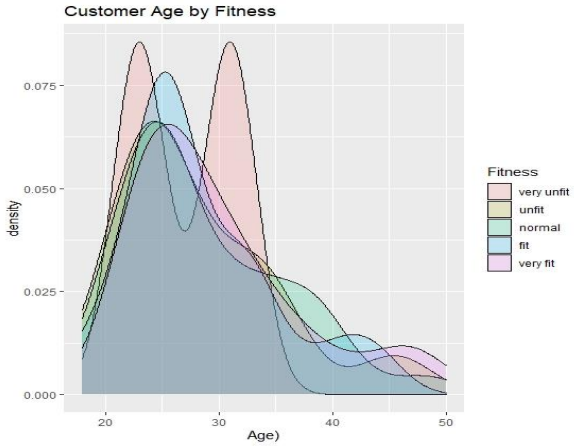
We can see relationship between two variables it goes its goes straightforward, nothing effect on each other.

- Age vs Usage



We can see relationship between two variables it goes its goes straightforward, nothing effect on each other.

- Age vs Fitness



We can see relationship between two variables (Age vs Fitness), up to 75% of customer between age (20 -30) rated themselves as fit.

3.5 Missing Value Identification

We don't have missing data hear in our dataset

3.6 Outlier Identification

In the following table, we will present the outlier for each Quantitative variable

Income	Age	Usage	Education	Miles
83416, 88396, 90886, 92131, 85906, 103336, 99601, 89641, 95866, 92131, 104581, 95508	47, 50, 48	6, 7	20, 21	200, 212, 240, 260, 280, 300, 360

3.7 Variable Transformation / Feature Creation

We do not see that we need to create new feature

4. Conclusion

As we assume that the product sells distinguished be varies of variable it's clear that it's really impact by the customer Income, Gender, Education, Fitness and Miles, we see that customer who's., where we can see customer with highest Income, Fitness, Males and Miles they are interested in TM798 product.

Therefore, that it is better to target that type of customer with this kind of products.

Second we notice that there are 75% of customers between age (20 -30) rated themselves as fit while there incomes is not high and Most of the customers Age are between 24 – 33. so we can say that they are a possible targeted audience for TM798 product with facilitates in the payment method as installment.

5. Appendix A – Source Code

```
#=====================================================================
#
# Exploratory Data Analysis - CardioFitness
#
#=====================================================================
#calling all libraries that we are going to use
library(readr)
library(ggplot2)
library(plyr)
library(scales)

#setting up working directory
setwd("……../PGP DSBA/DSBA/Data")

#reading data from csv file to CardioGoodFitness variable
CardioGoodFitness <- read.csv("CardioGoodFitness.csv")
```

```r
#Retrieve the dimension of an object.
dim(CardioGoodFitness)

#Get the names of an object.
names(CardioGoodFitness)

#Display the internal structure of an dataset.
str(CardioGoodFitness)

#Returns the first 10 rows of the dataset.
head(CardioGoodFitness, 10)

#Returns the last 10 rows of the dataset.
tail(CardioGoodFitness, 10)

#Return a summary of the dataset variables.
summary(CardioGoodFitness)

#check if ther is any NA value in dataset
anyNA(CardioGoodFitness)

#convert from quantitative to qualitative
CardioGoodFitness$Fitness <- factor(CardioGoodFitness$Fitness, order = F,
levels =c("1", "2", "3", "4", "5"))
CardioGoodFitness$Fitness <- factor(mapvalues(Fitness, from = c("1", "2",
"3", "4", "5"), to = c("very unfit", "unfit", "normal", "fit", "very fit")))
summary(CardioGoodFitness)

#objects in the dataset can be accessed by simply giving their names
attach(CardioGoodFitness)

#get Mode function
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

# univariate Analysis
#product
#Mode
getmode(Product)

#Age
#mean
mean(Age)
#median
median(Age)
#mode
getmode(Age)
#get Q1 Q3
quantile(Age, c(0.25, 0.75), type = 1)
#IQR
unname(quantile(Age, c(0.25, 0.75), type = 1)[2] - quantile(Age, c(0.25,
0.75), type = 1)[1])
#min value
```

```r
min(Age)
#max value
max(Age)

#Gender
#mode
getmode(Gender)

#Education
#mean
mean(Education)
#median
median(Education)
#mode
getmode(Education)
#get Q1 Q3
quantile(Education, c(0.25, 0.75), Education = 1)
#IQR
unname(quantile(Education, c(0.25, 0.75), type = 1)[2] - quantile(Education,
c(0.25, 0.75), type = 1)[1])
#min value
min(Education)
#max value
max(Education)

#MaritalStatus
#mode
getmode(MaritalStatus)

#Usage
#mean
mean(Usage)
#median
median(Usage)
#mode
getmode(Usage)
#get Q1 Q3
quantile(Usage, c(0.25, 0.75), Usage = 1)
#IQR
unname(quantile(Usage, c(0.25, 0.75), type = 1)[2] - quantile(Usage, c(0.25,
0.75), type = 1)[1])
#min value
min(Usage)
#max value
max(Usage)

#Fitness
#mode
getmode(Fitness)

#Income
#mean
mean(Income)
#median
median(Income)
#mode
```

```
getmode(Income)
#get Q1 Q3
quantile(Income, c(0.25, 0.75), Income = 1)
#IQR
unname(quantile(Income, c(0.25, 0.75), type = 1)[2] - quantile(Income,
c(0.25, 0.75), type = 1)[1])
#min value
min(Income)
#max value
max(Income)

#Miles
#mean
mean(Miles)
#median
median(Miles)
#mode
getmode(Miles)
#get Q1 Q3
quantile(Miles, c(0.25, 0.75), type = 1)
#IQR
unname(quantile(Miles, c(0.25, 0.75), type = 1)[2] - quantile(Miles, c(0.25,
0.75), type = 1)[1])
#min value
min(Miles)
#max value
max(Miles)

#graph for all variable variables
# Quantitative
par(mfrow=c(2,3))
boxplot(Age, main = "Age")
boxplot(Education, main = "Education")
boxplot(Usage, main = "Usage")
hist(Age, main = "Age")
hist(Education, main = "Education")
hist(Usage, main = "Usage")

par(mfrow=c(2,2))
boxplot(Income/1000,main = "Income(K)")
boxplot(Miles, main = "Miles")
hist(Income/1000, main = "Income(K)")
hist(Miles, main = "Miles")

# catagorical
par(mfrow=c(2,2))
ggplot(CardioGoodFitness) + geom_bar(aes(x = Product))
ggplot(CardioGoodFitness) + geom_bar(aes(x = Gender))
ggplot(CardioGoodFitness) + geom_bar(aes(x = MaritalStatus))
ggplot(CardioGoodFitness) + geom_bar(aes(x = Fitness))


#Bi-Variate Analysis
# Customer Age by Fitness
#kernel density plots
ggplot(CardioGoodFitness,
```

```
       aes(x = Age, #quantitative variable
           fill = factor(Fitness, #defining x axis a categorical
                         levels = c("very unfit", "unfit", "normal", "fit",
"very fit"),
                         labels = c("very unfit", "unfit", "normal", "fit",
"very fit")))) +
  geom_density(alpha = 0.2) + #setting transparency of graph to keep overlaps
visible
  labs(fill = "Fitness", # setting title of legend
       x = "Age)",
       title = "Customer Age by Fitness")

# Customer Incoms by Product type
#kernel density plots
ggplot(CardioGoodFitness,
       aes(x = Income/1000, #quantitative variable
           fill = factor(Product, #defining x axis a categorical
                         levels = c("TM195", "TM498", "TM798"),
                         labels = c("TM195", "TM498", "TM798")))) +
  geom_density(alpha = 0.2) + #setting transparency of graph to keep overlaps
visible
  labs(fill = "Products", # setting title of legend
     x = "Income (K)",
     title = "Customer Incoms by Product type")

#jitter and box plots
ggplot(CardioGoodFitness,
       aes(x = factor(Product, #defining x axis a categorical
                      labels = c("TM195", "TM498", "TM798")),
           y = Income/1000,
           color = Product)) + #specifying that coloring is to be based on
drive type
  geom_boxplot(size=1, #makes the lines thicker
               outlier.shape = 1, #specifies circles for outliers
               outlier.color = "black", #makes outliers black
               outlier.size = 3) + #increases the size of the outlier symbol
  geom_jitter(alpha = 0.5, #setting transparency of graph
              width=.2) + #decreases the amount of jitter (.4 is the default)
  labs(title = "Costomer Income by Product Type",
       x = "",
       y = "Income (k)") +
  theme_minimal() + #setting minimal theme (no background color)
  theme(legend.position = "none") + #hiding legend
  coord_flip() #x and y axes are reversed

#Customer Education by Product type
#kernel density plots
ggplot(CardioGoodFitness,
       aes(x = Education, #quantitative variable
           fill = factor(Product, #defining x axis a categorical
                         levels = c("TM195", "TM498", "TM798"),
                         labels = c("TM195", "TM498", "TM798")))) +
  geom_density(alpha = 0.2) + #setting transparency of graph to keep overlaps
visible
  labs(fill = "Products", # setting title of legend
       x = "Education",
```

```
       title = "Customer Education by Product type")

#jitter and box plots
ggplot(CardioGoodFitness,
       aes(x = factor(Product, #defining x axis a categorical
                      labels = c("TM195", "TM498", "TM798")),
           y = Education,
           color = Product)) + #specifying that coloring is to be based on
drive type
  geom_boxplot(size=1, #makes the lines thicker
               outlier.shape = 1, #specifies circles for outliers
               outlier.color = "black", #makes outliers black
               outlier.size = 3) + #increases the size of the outlier symbol
  geom_jitter(alpha = 0.5, #setting transparency of graph
              width=.2) + #decreases the amount of jitter (.4 is the default)
  labs(title = "Customer Education by Product type",
       x = "",
       y = "Education") +
  theme_minimal() + #setting minimal theme (no background color)
  theme(legend.position = "none") + #hiding legend
  coord_flip() #x and y axes are reversed


#Customer Miles by Product type
#kernel density plots Customer Miles by Product type
ggplot(CardioGoodFitness,
       aes(x = Miles, #quantitative variable
           fill = factor(Product, #defining x axis a categorical
                         levels = c("TM195", "TM498", "TM798"),
                         labels = c("TM195", "TM498", "TM798")))) +
  geom_density(alpha = 0.2) + #setting transparency of graph to keep overlaps
visible
  labs(fill = "Products", # setting title of legend
       x = "Miles",
       title = "Customer Miles by Product type")

#jitter and box plots
ggplot(CardioGoodFitness,
       aes(x = factor(Product, #defining x axis a categorical
                      labels = c("TM195", "TM498", "TM798")),
           y = Miles,
           color = Product)) + #specifying that coloring is to be based on
drive type
  geom_boxplot(size=1, #makes the lines thicker
               outlier.shape = 1, #specifies circles for outliers
               outlier.color = "black", #makes outliers black
               outlier.size = 3) + #increases the size of the outlier symbol
  geom_jitter(alpha = 0.5, #setting transparency of graph
              width=.2) + #decreases the amount of jitter (.4 is the default)
  labs(title = "Customer Miles by Product type",
       x = "",
       y = "Miles") +
  theme_minimal() + #setting minimal theme (no background color)
  theme(legend.position = "none") + #hiding legend
  coord_flip() #x and y axes are reversed
```

```
#Customer Usage by Product type
#kernel density plots Customer Usage by Product type
ggplot(CardioGoodFitness,
       aes(x = Usage, #quantitative variable
           fill = factor(Product, #defining x axis a categorical
                         levels = c("TM195", "TM498", "TM798"),
                         labels = c("TM195", "TM498", "TM798")))) +
  geom_density(alpha = 0.2) + #setting transparency of graph to keep overlaps
visible
  labs(fill = "Products", # setting title of legend
       x = "Usage",
       title = "Customer avrage Usage by Product type")


ggplot(CardioGoodFitness,
       aes(x = factor(Product, #defining x axis a categorical
                      labels = c("TM195", "TM498", "TM798")),
           y = Usage,
           color = Product)) + #specifying that coloring is to be based on
drive type
  geom_boxplot(size=1, #makes the lines thicker
               outlier.shape = 1, #specifies circles for outliers
               outlier.color = "black", #makes outliers black
               outlier.size = 3) + #increases the size of the outlier symbol
  geom_jitter(alpha = 0.5, #setting transparency of graph
              width=.2) + #decreases the amount of jitter (.4 is the default)
  labs(title = "Customer Avrage Usage by Product type",
       x = "",
       y = "Usage") +
  theme_minimal() + #setting minimal theme (no background color)
  theme(legend.position = "none") + #hiding legend
  coord_flip() #x and y axes are reversed



#Custome Gender by Product type
# stacked bar chart
ggplot(CardioGoodFitness,
       aes(x = Gender,
           fill = factor(Product,
                         levels = c("TM195", "TM498", "TM798"),
                         labels = c("TM195", "TM498", "TM798")))) +
  labs(fill = "Products", # setting title of legend
       x = "Gender",
       title = "Custome Gender by Product type") +
  geom_bar(position = "stack") #specifying the type of bar chart as stacked

#Custome Fitness by Product type
# stacked bar chart
ggplot(CardioGoodFitness,
       aes(x = Fitness,
           fill = factor(Product,
                         levels = c("TM195", "TM498", "TM798"),
                         labels = c("TM195", "TM498", "TM798")))) +
  labs(fill = "Products", # setting title of legend
```

```
       x = "Fitness",
       title = "Custome Fitness by Product type") +
  geom_bar(position = "stack") #specifying the type of bar chart as stacked

#Custome MaritalStatus by Product type
# stacked bar chart
ggplot(CardioGoodFitness,
       aes(x = MaritalStatus,
           fill = factor(Product,
                         levels = c("TM195", "TM498", "TM798"),
                         labels = c("TM195", "TM498", "TM798")))) +
  labs(fill = "Products", # setting title of legend
       x = "Marital Status",
       title = "Custome Marital Status by Product type") +
  geom_bar(position = "stack") #specifying the type of bar chart as stacked

#Relation between Customer Income and Age
#scatter plot
ggplot(CardioGoodFitness,aes(x = Income/1000,y = Age)) +
  geom_point(color="cornflowerblue", #setting the colour, size and
transparency(alpha) of the points
size = 2,
alpha=.8) +
  labs(x = "Income(K)", #specifying the labels of axes and title of plot
       y = "Age",
       title = "Income(K) vs Age",
       subtitle = "Relation between Customer Income and Age") +
  geom_smooth(method = "lm") # this adds a linear trend line which is useful
to summarize the relationship between the two variables


#Relation between Customer Income and Education
#scatter plot
ggplot(CardioGoodFitness,aes(x = Income/1000,y = Education)) +
  geom_point(color="cornflowerblue", #setting the colour, size and
transparency(alpha) of the points
             size = 2,
             alpha=.8) +
  labs(x = "Income(K)", #specifying the labels of axes and title of plot
       y = "Education",
       title = "Income(K) vs Education",
       subtitle = "Relation between Customer Income and Education") +
  geom_smooth(method = "lm") # this adds a linear trend line which is useful
to summarize the relationship between the two variables

#Relation between Customer Age and Miles
#scatter plot
ggplot(CardioGoodFitness,aes(x = Age,y = Miles)) +
  geom_point(color="cornflowerblue", #setting the colour, size and
transparency(alpha) of the points
             size = 2,
             alpha=.8) +
  labs(x = "Age", #specifying the labels of axes and title of plot
       y = "Miles",
       title = "Age vs Miles",
       subtitle = "Relation between Customer Income and Miles") +
```

```
  geom_smooth(method = "lm") # this adds a linear trend line which is useful
to summarize the relationship between the two variables

#Relation between Customer Age and Usage
#scatter plot
ggplot(CardioGoodFitness,aes(x = Age,y = Usage)) +
  geom_point(color="cornflowerblue", #setting the colour, size and
transparency(alpha) of the points
             size = 2,
             alpha=.8) +
  labs(x = "Age", #specifying the labels of axes and title of plot
       y = "Usage",
       title = "Age vs Usage",
       subtitle = "Relation between Customer Age and Usage") +
  geom_smooth(method = "lm") # this adds a linear trend line which is useful
to summarize the relationship between the two variables

#outlier identification
#Income
boxplot.stats(CardioGoodFitness$Income)$out

#Age
boxplot.stats(CardioGoodFitness$Age)$out

#Usage
boxplot.stats(CardioGoodFitness$Usage)$out

#Education
boxplot.stats(CardioGoodFitness$Education)$out

#Miles
boxplot.stats(CardioGoodFitness$Miles)$out

#=============================================================================
#
# T H E - E N D
#
#=============================================================================
```