Mini 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

Given the nature of the data provided in the dataset, it is assumed that the data is related to the customers who have purchased the treadmill products in the last quarter (April – June 2017). The 180 rows of the dataset correspond to 180 unique customers of the treadmill products. The features in the dataset are linked to the demographics and treadmill usage characteristics of the customers.

Also, the following data dictionary is considered for the 9 features in the dataset:

Sl. No.	Feature Name	Feature Code	Feature Description
1	Product	Product	Model of treadmill product (TM195/ TM498/ TM 798)
2	Age	Age	Age of the customer (Years)
3	Gender	Gender	Gender of the customer (Male & Female)
4	Education	Education	Education of the customer (Years)
5	Marital Status	MaritalStatus	Marital status of the customer (Single & Partnered/ Married)
6	Usage	Usage	Weekly average number of times the customer plans to use the treadmill (No. of times/ Week)
7	Fitness Level	Fitness	Weekly average number of miles the customer expects to walk/run on the treadmill (Miles/ Week)
8	Household Income	Income	Annual household income of the customer ($)
9	Total Distance Covered	Miles	Total distance covered on the treadmill (Miles)

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

Use this section to install necessary packages and invoke associated libraries. Having all the packages at the same places increases code readability.

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. Basically, 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

The dataset is analysed for basic understanding of features and data. It is usually an activity by which data is explored and organized in order that information it contains is made clear.

Following R functions used during this step:

- dim(): See dimensions (# of rows/ # of columns) of the data frame.

- names(): See Feature names of the dataset.
- str(): Display internal structure of an R object, to identify classes of the features.

Individual Feature classes can be obtained by using **class()** function, by passing appropriate Feature name. This is an optional function.

- head(): Obtain the first six rows of the data frame.
- tail(): Obtain the last six rows of the data frame.

3.2.1 Variable Identification – Inferences

- **No. of Rows and Columns:**

No. of Rows	No. of Columns (Features)
180	9

  - The number of rows in the dataset is 180 and
  - The number of columns (Features) is 9.

- **Features and their Types:**
  - Various **Continuous** and **Categorical** Features of the dataset are as follows:

Sr. No.	Feature Name	Type	Continuous / Categorical
1	Product	Factor	Categorical
2	Age	Integer	Continuous
3	Gender	Factor	Categorical
4	Education	Integer	Continuous
5	Marital Status	Factor	Categorical
6	Usage	Integer	Continuous
7	Fitness	Integer	Continuous
8	Income	Integer	Continuous
9	Miles	Integer	Continuous

3.3 Univariate Analysis

In this step, we will explore variables one by one. The Method to perform uni-variate analysis depends on whether the variable type is categorical or continuous. Let's look at these methods and statistical measures for categorical and continuous variables individually:

Continuous Variables:- In case of continuous variables, we need to understand the central tendency and spread of the variable. These are measured using various statistical metrics visualization methods as shown below:

Central Tendency	Measure of Dispersion	Visualization Method
Mean	Range	Histogram
Median	Quartile	Boxplot
Mode	IQR	
Min	Variance	
Max	Standard Deviation	
	Skewness and Kurtosis	

Identification of Missing Values and Outliers:- The next activity for Continuous Variables is to identify the Missing Values and Outliers. There are methods to handle Missing Values and Outliers, but since it is out of scope for this exercise, we shall concentrate only on the identification part.

Categorical Variables:- For categorical variables, we'll use frequency table to understand distribution of each category. We can also read as percentage of values under each category. It can be measured using two metrics, **Count and Count%** against each category, which are part of **plyr** package. **Bar chart** and **Pie Chart** can be used as visualization.

3.3.1 Continuous Variables Analysis

summary(): Provides summary of the dataset.

- **Summary Statistics of the Continuous variables is as follows:**

Feature	Min. Val	1st Q	Median	Mean	3rd Q	Max
Age	18	24	26	28.79	33	50
Education	12	14	16	15.57	16	21
Usage	2	3	3	3.45	4	7
Fitness	1	3	3	3.31	4	5
Income	29,560	44,060	50,600	53,720	58,670	1,04,600
Miles	21	66	94	103.2	114.8	360

- colSums(is.na()): Check missing values.
- **Missing Values:**

Missing Values
None of the Features contain any missing values.

3.3.2 Categorical Variables Analysis

table(): List all values of a variable with frequencies

- **Features Summary:**
  - Population distribution of various Products, Gender and Marital Status is provided in the following tables:

Product	Quantity
TM195	80
TM498	60
TM798	40

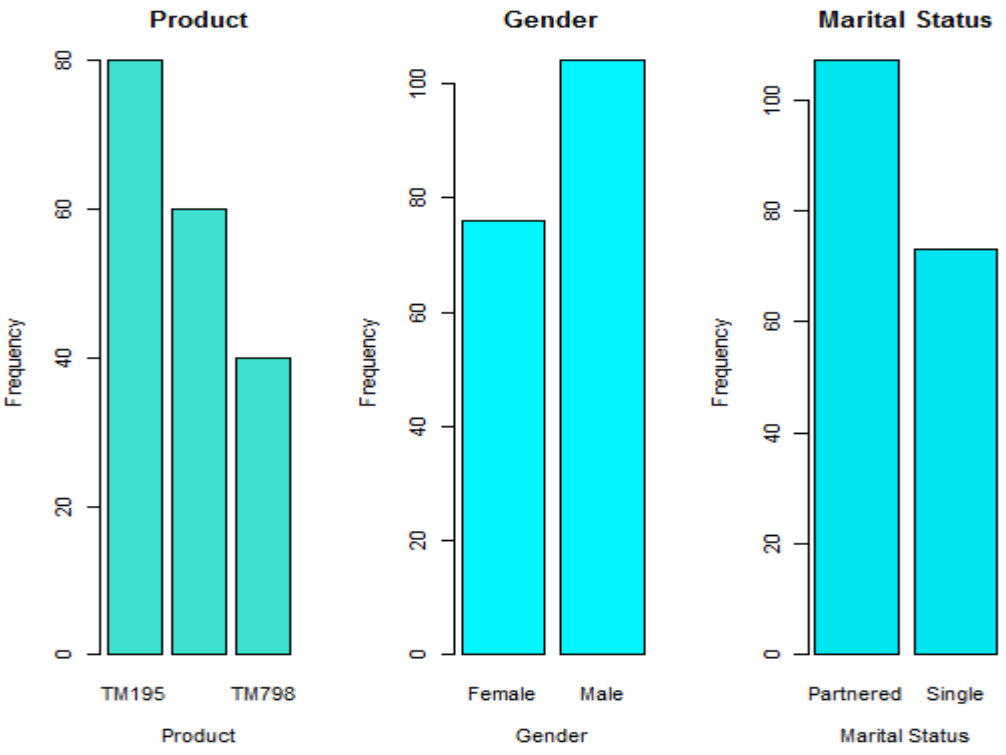
Gender	Count
Female	76
Male	104

Marital Status	Count
Partnered	107
Single	73

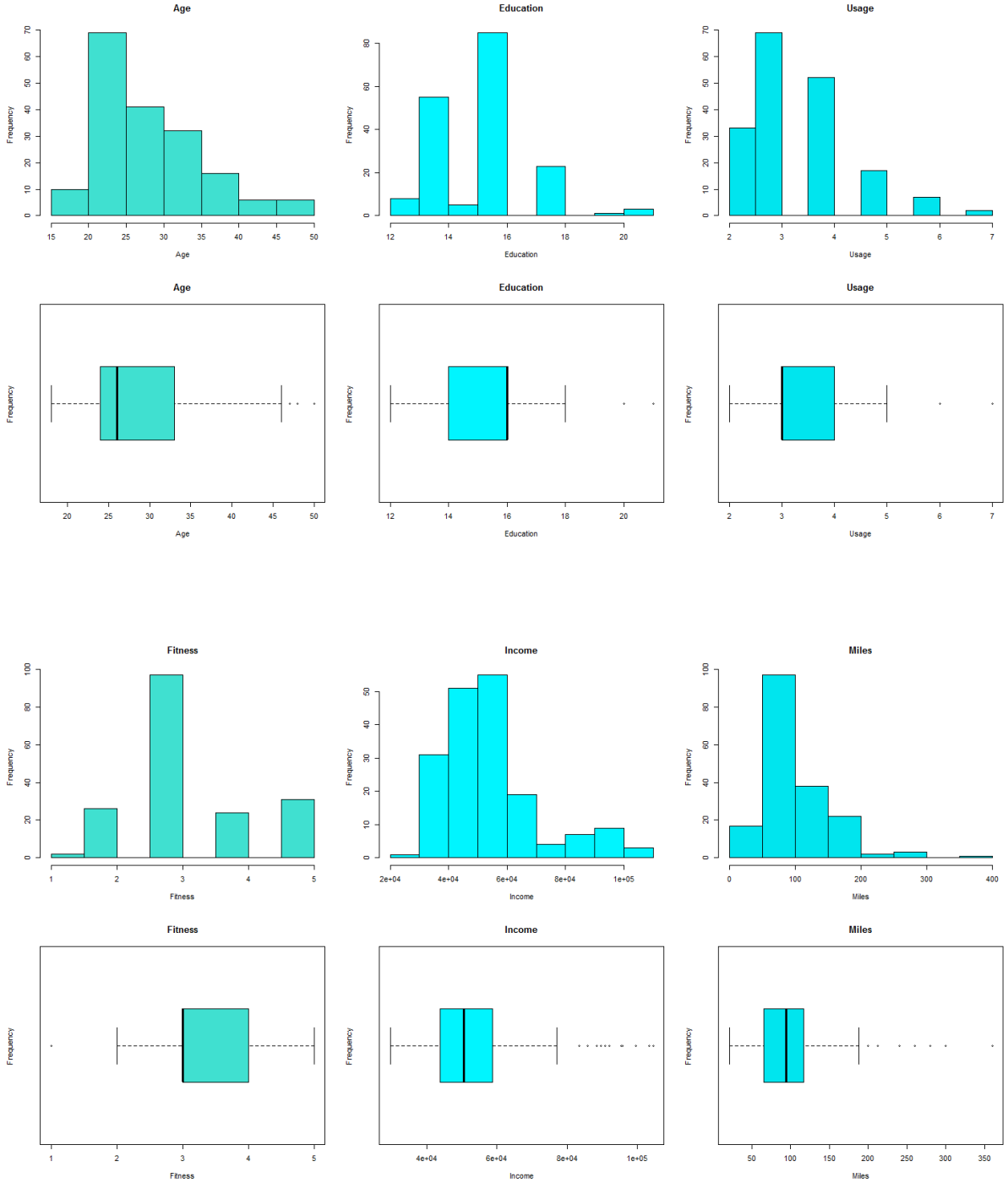
3.3.2.1 Categorical Variables Analysis – Visualization

The Categorical Variables: Product, Gender and Marital Status:

Create a Bar plot to visualise three categorical variables.



The Continuous Variables: using Histogram and Box Plot:



3.3.3 Continuous and Categorical Variables: Key Observations

- ***Product:*** The count by each product type (i.e. treadmill) is provided. TM195 has the highest count (80), followed by TM498 (60) and TM798 (40)
- ***Age:*** The minimum age of the customer is 18 years and the maximum age of the customer is 50 years. The average age of customers is 28.79 years. The Boxplot indicates some outliers beyond the age of 46.

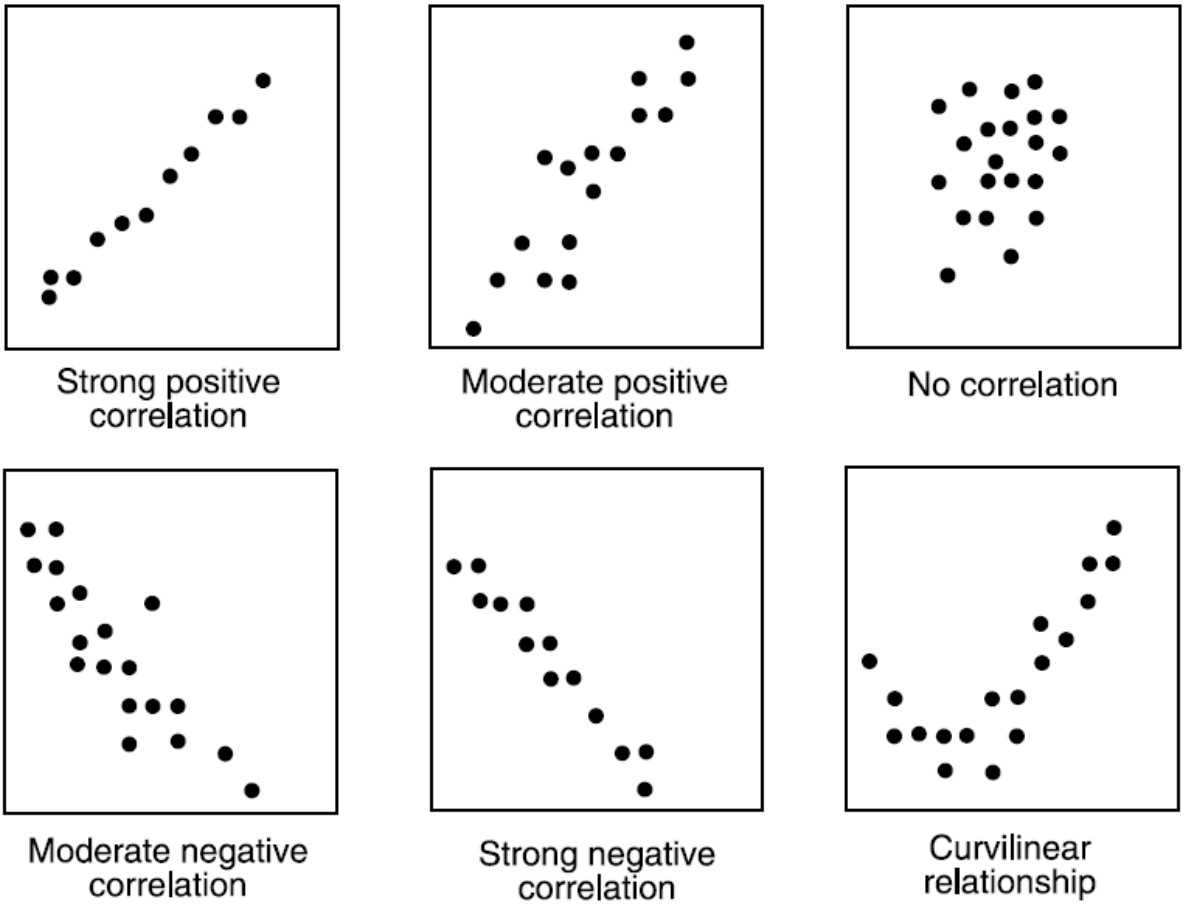
- ***Gender:*** Number of males (104) as customers is greater than number of females (76) as customers.
- ***Education:*** The minimum education year of the customer is 12 years and the maximum education year of the customer is 21 years. The average years of education of customers is 15.57 years. The Boxplot indicates few outliers on the higher side, 18 years.
- **MaritalStatus:** Partnered, i.e. married (107) as customers is greater than single (73) as customers
- ***Usage:*** The minimum number of times the customer wants to use treadmill in a week is 2 and the maximum number of times the customer wants to use treadmill in a week is 7, i.e. all days of the week. The weekly average of all the customers is 3.456. The Boxplot indicates few outliers on the higher side, beyond 5.
- ***Fitness:*** The minimum number of miles the customer wants to cover on the treadmill in a week is 1 and the maximum number of times the customer wants to cover on the treadmill in a week is 5. The weekly average of all the customers is 3.311 miles. The Boxplot indicates an outlier on the lower side, with fitness level 1.
- ***Income:*** The minimum average household income of the customer is $ 29,562 and the maximum average household income of the customer is $ 104,581. For all the customers, the average is $ 53,720. The Boxplot indicates many outliers on the higher side, with Income beyond $75000.
- ***Miles:*** The lowest total distance covered by the customer is 21 miles and the highest total distance covered by the customer is 360 miles. The average of all the customers is 103.2 miles. The Boxplot indicates many outliers on the higher side, beyond 180 miles.

3.4 Bi-Variate Analysis

Bi-variate Analysis finds out the relationship between two variables. Here, we look for association and disassociation between variables at a pre-defined significance level. We can perform bi-variate analysis for any combination of categorical and continuous variables. The combination can be: Categorical & Categorical, Categorical & Continuous and Continuous & Continuous. Different methods are used to tackle these combinations during analysis process.

Let's understand the possible combinations in detail:

Continuous & Continuous: While doing bi-variate analysis between two continuous variables, we should look at **scatter plot**. It is a nifty way to find out the relationship between two variables. The pattern of scatter plot indicates the relationship between variables. The relationship can be linear or non-linear.



Scatter plot shows the relationship between two variable, but does not indicates the strength of relationship amongst them. To find the strength of the relationship, we use Correlation. Correlation varies between -1 and +1.

- -1: perfect negative linear correlation
- +1:perfect positive linear correlation and
- 0: No correlation

Correlation can be derived using following formula:

Correlation = Covariance(X,Y) / SQRT(Var(X)* Var(Y))

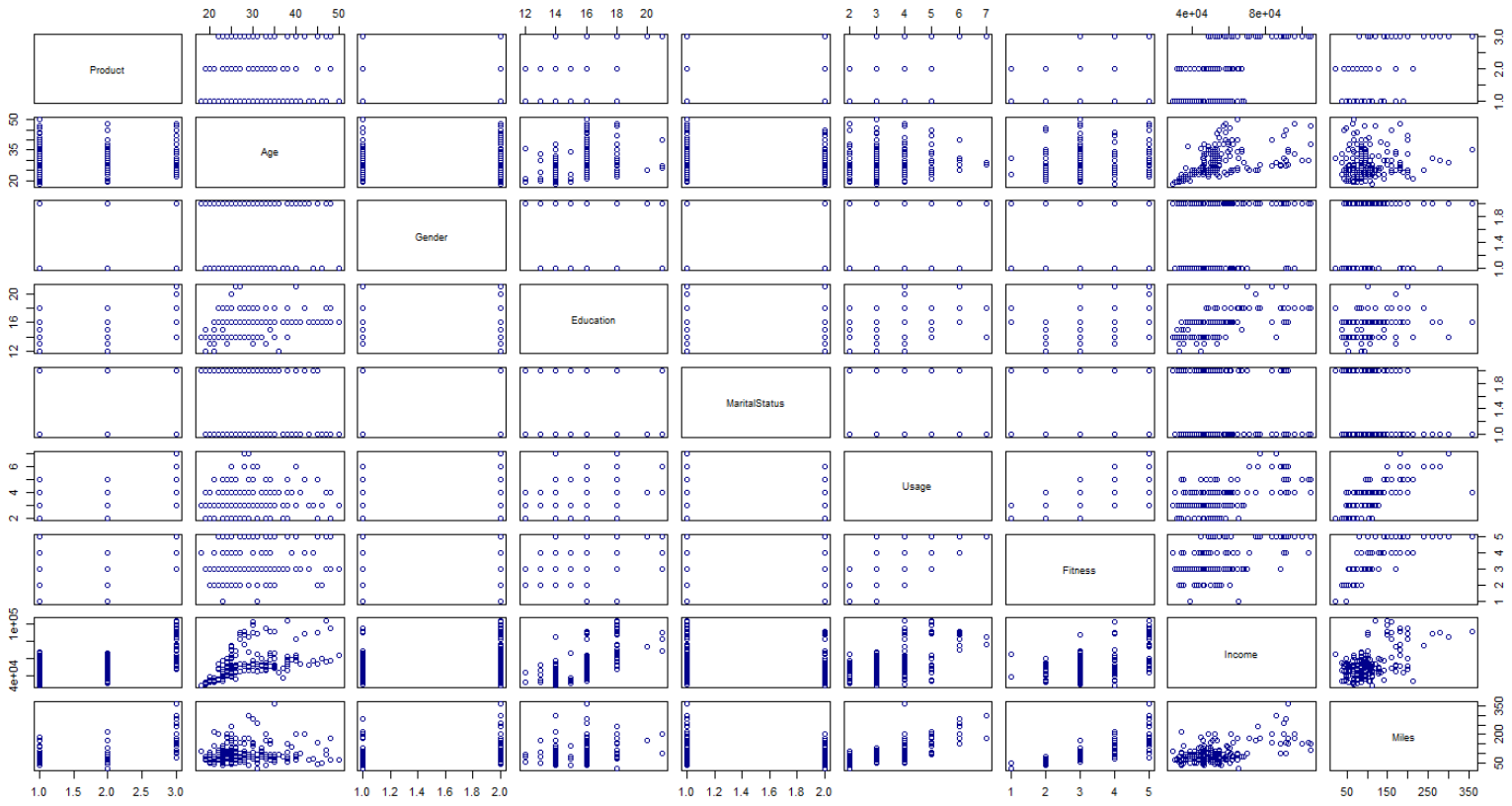
Categorical & Categorical: To find the relationship between two categorical variables, we can use following methods:

- **Two-way table:** We can start analysing the relationship by creating a two-way table of count. The rows represents the category of one variable and the columns represent the categories of the other variable.
- **Stacked Column Chart:** This method is more of a visual form of Two-way table.

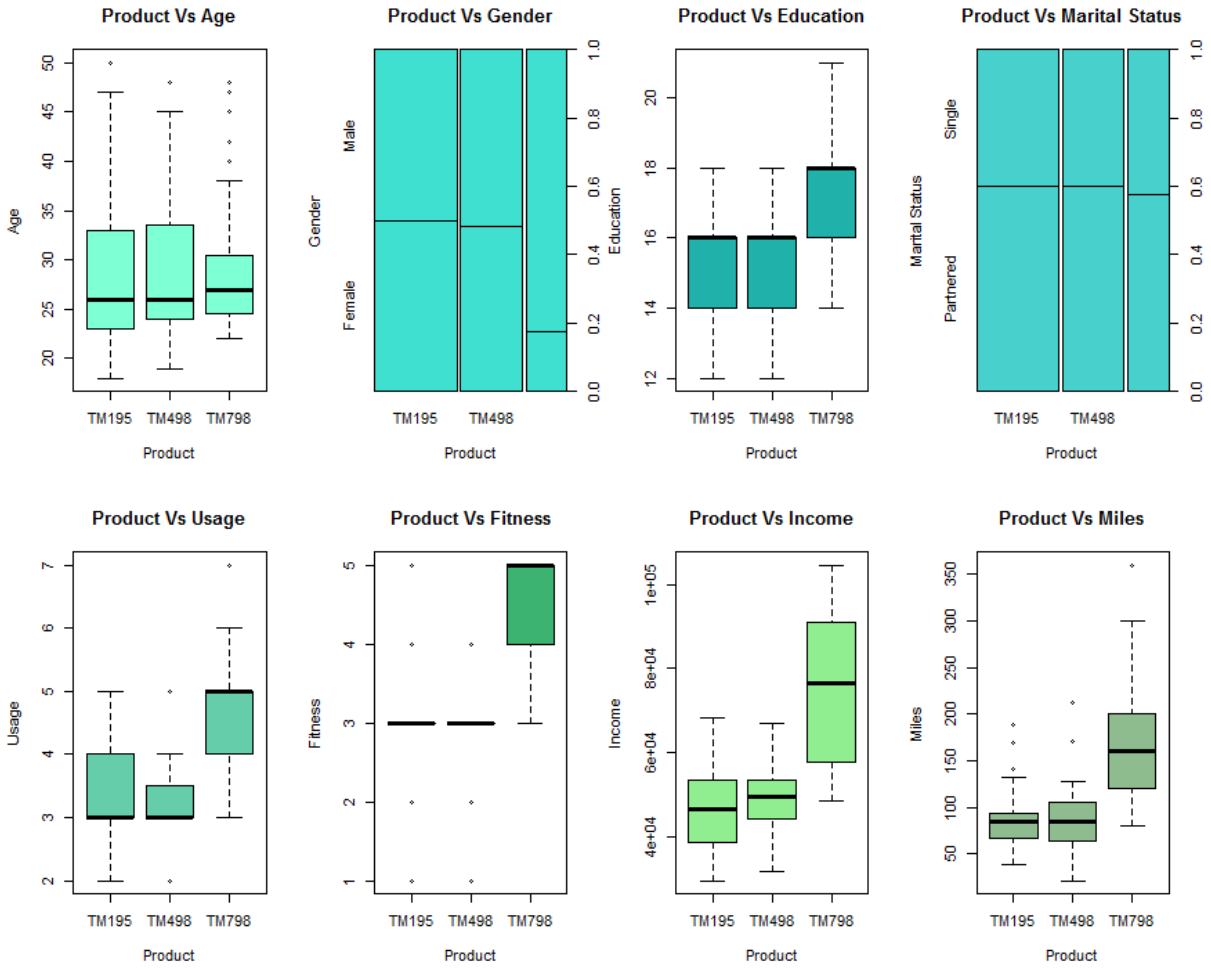
Categorical & Continuous: While exploring relation between categorical and continuous variables, we can draw box plots for each level of categorical variables. If levels are small in number, it will not show the statistical significance. To look at the statistical significance we can perform Z-test, T-test or ANOVA, however, this is beyond scope of our project.

3.4.1 Bivariate Analysis – Cardio Good Fitness

Let's analyse the relationship of the 9 features with each other. The visualization can be obtained either by using pairs() function, or plot() function, giving one pair variables at a time. Following graph is created using the pairs() function.



Individual pairwise plots:



Interpretation:

- **Product Vs Age:** TM195 is popular among all age groups. TM798 is popular among 22-47 age group.
- **Product Vs Gender:** TM798 is popular among Male.
- **Product Vs Education:** TM798 is more popular in highly educated people.
- **Product Vs Marital Status:** No Correlation found.
- **Product Vs Usage**: TM798 has more usage level.
- **Product Vs Fitness:** TM195 and TM498 is mostly used by people with fitness level of 3, whereas TM798 is popular among 4 and 5 fitness levels.
- **Product Vs Income:** TM195 and TM498 are popular among low and medium income group, whereas TM798 is popular among medium to high income group.
- **Product Vs Miles:** Positive correlation, as the Product range increases from TM195 to TM498 and TM798, there is an increase in Total Distance covered.

- **Age Vs Gender:** No Correlation found.
- **Age Vs Education:** No Correlation found.
- **Age Vs Marital Status:** No Correlation found.
- **Age Vs Usage**: Usage levels 2,3,4 and 5 are popular among all age groups, whereas Usage Levels 6 and 7 are popular among 30-40 age group.
- **Age Vs Fitness:** No Correlation found.
- **Age Vs Income:** Positive Correlation. Income increases along with Age.
- **Age Vs Miles:** Random distribution. Probable candidates for Transformation.
- **Gender Vs Education:** No Correlation found.
- **Gender Vs Marital Status:** No Correlation found.
- **Gender Vs Usage**: Uniform Usage level up to 6. Level 7 is popular among Male.
- **Gender Vs Fitness:** No Correlation found.
- **Gender Vs Income:** Income is slightly more among Male.
- **Gender Vs Miles:** Male tend to burns more Miles than Female.
- **Education Vs Marital Status:** No Correlation found.
- **Education Vs Usage**: No Prominent correlation found.
- **Education Vs Fitness:** No Correlation found.
- **Education Vs Income:** Positive Correlation, Higher income for highly educated person.
- **Education Vs Miles:** Persons at education level 16 tend to burn more miles.
- **Marital Status Vs Usage**: Partnered have a tendency of more usage.
- **Marital Status Vs Fitness:** No Correlation found.
- **Marital Status Vs Income:** Partnered persons have slightly higher income level.
- **Marital Status Vs Miles:** Partnered person’s burn more miles.
- **Usage Vs Fitness:** Positive Correlation. More usage with increased fitness.
- **Usage Vs Income:** Higher usage is found at higher income level, however, average usage is found across income groups.
- **Usage Vs Miles:** Positive Correlation found. Miles burned increases with increase in Usage.
- **Fitness Vs Income:** Positive correlation found. High fitness level observed among high income group.
- **Fitness Vs Miles:** Positive Correlation found. Persons with High fitness level tend to burn more miles.
- **Income Vs Miles:** Positive Correlation found.

3.5 Missing Value Treatment

Missing value treatment is an important step in Exploratory Data Analysis, as missing data in the training data set can reduce the power / fit of a model or can lead to a biased model because we have not analysed the behaviour and relationship with other variables correctly. It can lead to wrong prediction or classification.

However, the scope of this project is just to explore the data and derive inferences, hence we are not going in details of how missing value treatment can be done.

3.6 Outlier Treatment

Outlier is a commonly used terminology by analysts and data scientists as it needs close attention else it can result in wildly wrong estimations. Simply speaking, Outlier is an observation that appears far away and diverges from an overall pattern in a sample.

Outliers can drastically change the results of the data analysis and statistical modeling. There are numerous unfavourable impacts of outliers in the data set:

- It increases the error variance and reduces the power of statistical tests
- If the outliers are non-randomly distributed, they can decrease normality
- They can bias or influence estimates that may be of substantive interest
- They can also impact the basic assumption of Regression, ANOVA and other statistical model assumptions.

Most commonly used method to detect outliers is visualization. We use various visualization methods, like **Box-plot, Histogram, and Scatter Plot.**

There are various ways of dealing with outliers. However, similar to Missing Value Treatment, dealing with Outliers too is out of scope for this Project.

3.7 Variable Transformation / Feature Creation

Variable Transformation and Feature Creation are part of Feature Engineering, and can be performed once we are done with the earlier steps of Variable Identification, Univariate / Bivariate Analysis, Missing Value imputation and Outlier detection.

These two techniques are vital in data exploration and have a remarkable impact on the power of prediction.

Variable Transformation:

In data modelling, transformation refers to the replacement of a variable by a function. For instance, replacing a variable x by the square / cube root or

logarithm x is a transformation. In other words, transformation is a process that changes the distribution or relationship of a variable with others.

Below are the situations where variable transformation is a requisite:

- When we want to **change the scale** of a variable or standardize the values of a variable for better understanding. While this transformation is a must if you have data in different scales, this transformation does not change the shape of the variable distribution.
- When we can **transform complex non-linear relationships into linear relationships.** Existence of a linear relationship between variables is easier to comprehend compared to a non-linear or curved relation. Transformation helps us to convert a non-linear relation into linear relation. Scatter plot can be used to find the relationship between two continuous variables. These transformations also improve the prediction. Log transformation is one of the commonly used transformation technique used in these situations.
- **Symmetric distribution is preferred over skewed distribution** as it is easier to interpret and generate inferences. Some modeling techniques requires normal distribution of variables. So, whenever we have a skewed distribution, we can use transformations which reduce skewness. For right skewed distribution, we take square / cube root or logarithm of variable and for left skewed, we take square / cube or exponential of variables.

Common Methods of Variable Transformation:

- **Logarithm:** Log of a variable is a common transformation method used to change the shape of distribution of the variable on a distribution plot. It is generally used for reducing right skewness of variables. Though, It can’t be applied to zero or negative values as well.
- **Square / Cube root:** The square and cube root of a variable has a sound effect on variable distribution. However, it is not as significant as logarithmic transformation. Cube root has its own advantage. It can be applied to negative values including zero. Square root can be applied to positive values including zero.
- **Binning:** It is used to categorize variables. It is performed on original values, percentile or frequency. Decision of categorization technique is based on business understanding. For example, we can categorize income in three categories, namely: High, Average and Low. We can also perform co-variate binning which depends on the value of more than one variables.

Feature Creation:

- Feature / Variable creation is a process to generate a new variables / features based on existing variable(s). This step is used to highlight the hidden relationship in a variable.
- Most common methods of creating new variables is to derive from existing variables, or creating dummy variables.

3.7.1 Categorization of Continuous Variables (Binning):

The following continuous variables are grouped into various slabs:

- ***Age (Variable Name: Age)***
  - Early Youth: 18 – 25 years
  - Late Youth: 26-35 years
  - Middle Aged Adults: 36+ years
- ***Education (Variable Name: Education)***
  - Higher Secondary: 12 years
  - Graduation: 13-16 years
  - Masters/ Post Graduation: 17+ years
- ***Usage (Variable Name: Usage)***
  - Fitness Amateurs: 2-3 times/ week
  - Fitness Regulars: 4-5 times/ week
  - Fitness Freaks: 6-7 times/ week
- ***Fitness Level (Variable Name: Fitness)***
  - Low Intensity: 1-2 miles/ week
  - Medium Intensity: 3-4 miles/ week
  - High Intensity: 5 miles/week
- ***Household Income (Variable Name: Income)***
  - Low Income Household: Upto $ 40,000
  - Medium Income Household: $ 40,001 - $ 60,000
  - High Income Household: $ 60,001+
- ***Total Distance Covered (Variable Name: Miles)***
  - Low Usage: Up to 60 miles
  - Medium Usage: 61-120 miles
  - High Usage: 121+ miles

The groupings of continuous variables were done using the command 'ifelse'. Also, the group slabs were defined by taking into considerations the summary parameters such as minimum value, mean value, quartile value etc. obtained through command 'summary'.

The R codes for the groupings of continuous variables by groupings of continuous variables are as follows:

Please refer Appendix A for Source Code.

After groupings of continuous variables, the data is appended to the dataset 'cardiodata' using the command 'cbind'.

The number of rows in the dataset is 180 and the number of columns is 15. In the new dataset, the number of columns has increased from 9 to 15.

3.7.2 Subset Creation:

Given the business objective is to identify different customer segments available in the market for the product (i.e., treadmill) by understanding whether there are differences across the product lines with respect to customer characteristics.

In such a scenario, the overall database (Variable="cgf_data_1") is divided into three subsets by product types:

- Product 'TM195': Variable="product1"
- Product 'TM498': Variable="product2"
- Product 'TM798': Variable="product3"

The subsets were created using the command 'which' as depicted in Source Code (Appendix A)

3.8 Feature Exploration

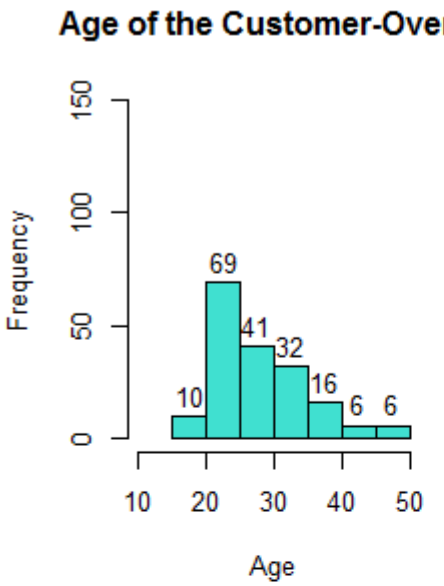
In this section, the features available in the new dataset 'cgf_data_1' will be explored in detail. The goal is to describe or summarize data in ways that are meaningful and useful for insights generation. It provides simple summaries about the sample and the measures. Together with simple graphics analysis, it forms the basis of virtually every quantitative analysis of data.

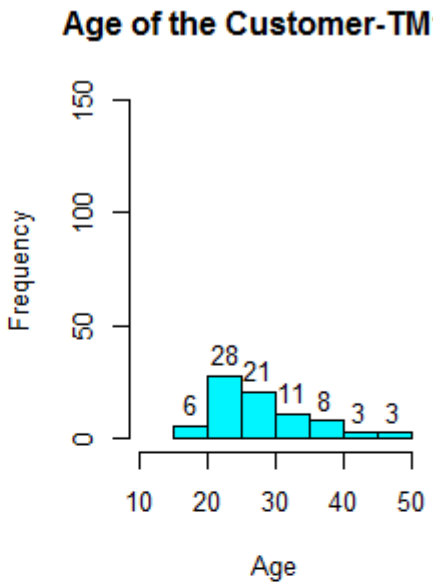
The feature exploration will be done by product types (i.e., treadmill).

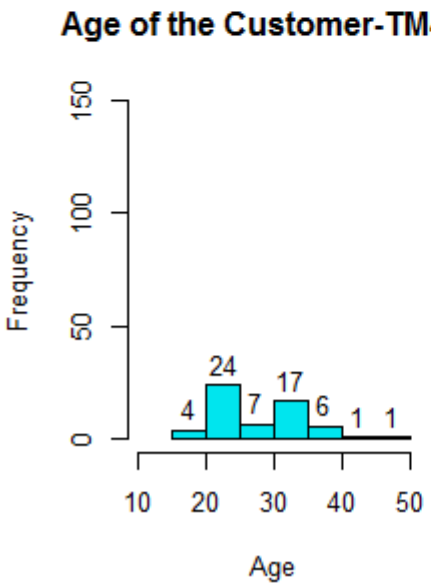
3.8.1 Age

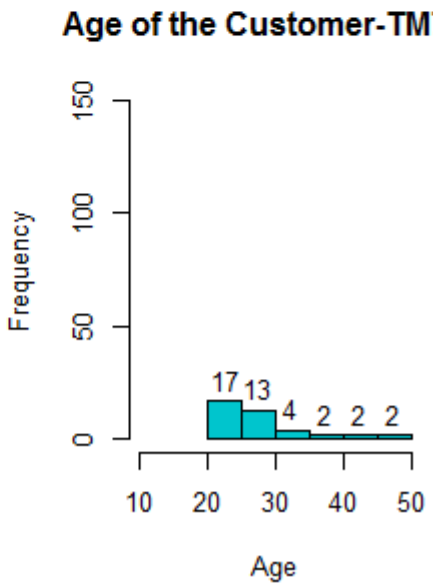
The variable 'Age' is a continuous varaible.

The histogram of Age for different Products is as follows:









Mean, Standard Deviation, and Variance:

Mean	Standard Deviation	Variance	Remark
28.79	6.94	48.21	Age Stats for All Products
28.55	7.22	52.15	Age Stats for TM195
28.9	6.65	44.16	Age Stats for TM498
28.1	6.97	48.61	Age Stats for TM798

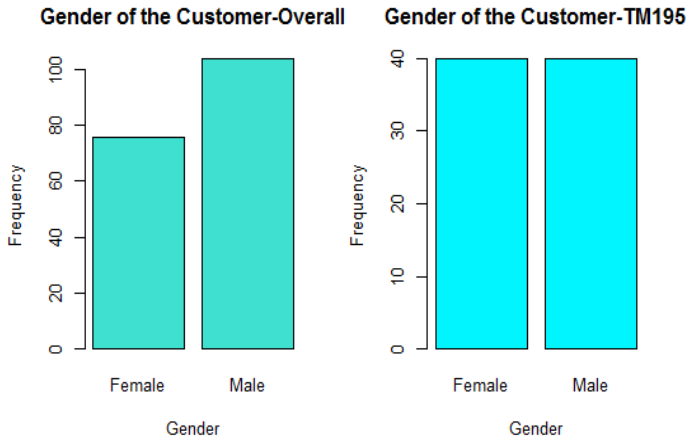
Interpretation:

- The majority of the treadmill customers had age between 21-30 years across product types – TM195, TM498 & TM798
- TM498 has relatively higher proportion of customers above 30 years of age as compared to TM195 & TM798
- The average customer age across product types (TM195, TM498 & TM798) is similar, approximately 29 years
- TM195 has relatively higher variation in the data distribution in comparison to TM498 & TM798

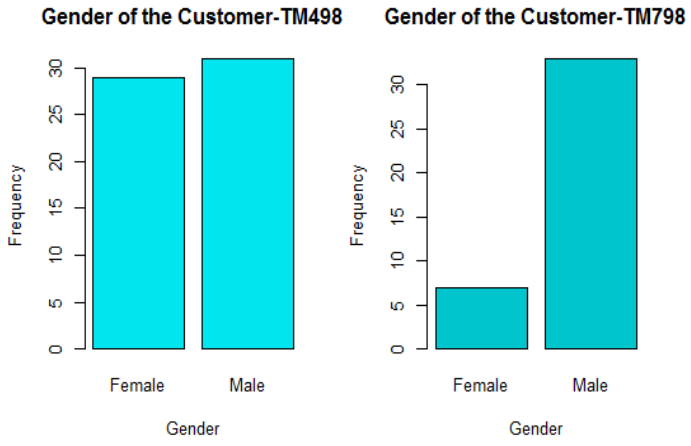
3.8.2 Gender

The variable 'Gender' is a categorical varaible.

Lets use barplot and Frequency Table to explore the details.



Product	Female	Male
Overall	76	104
TM195	40	40
TM498	29	31
TM798	7	33



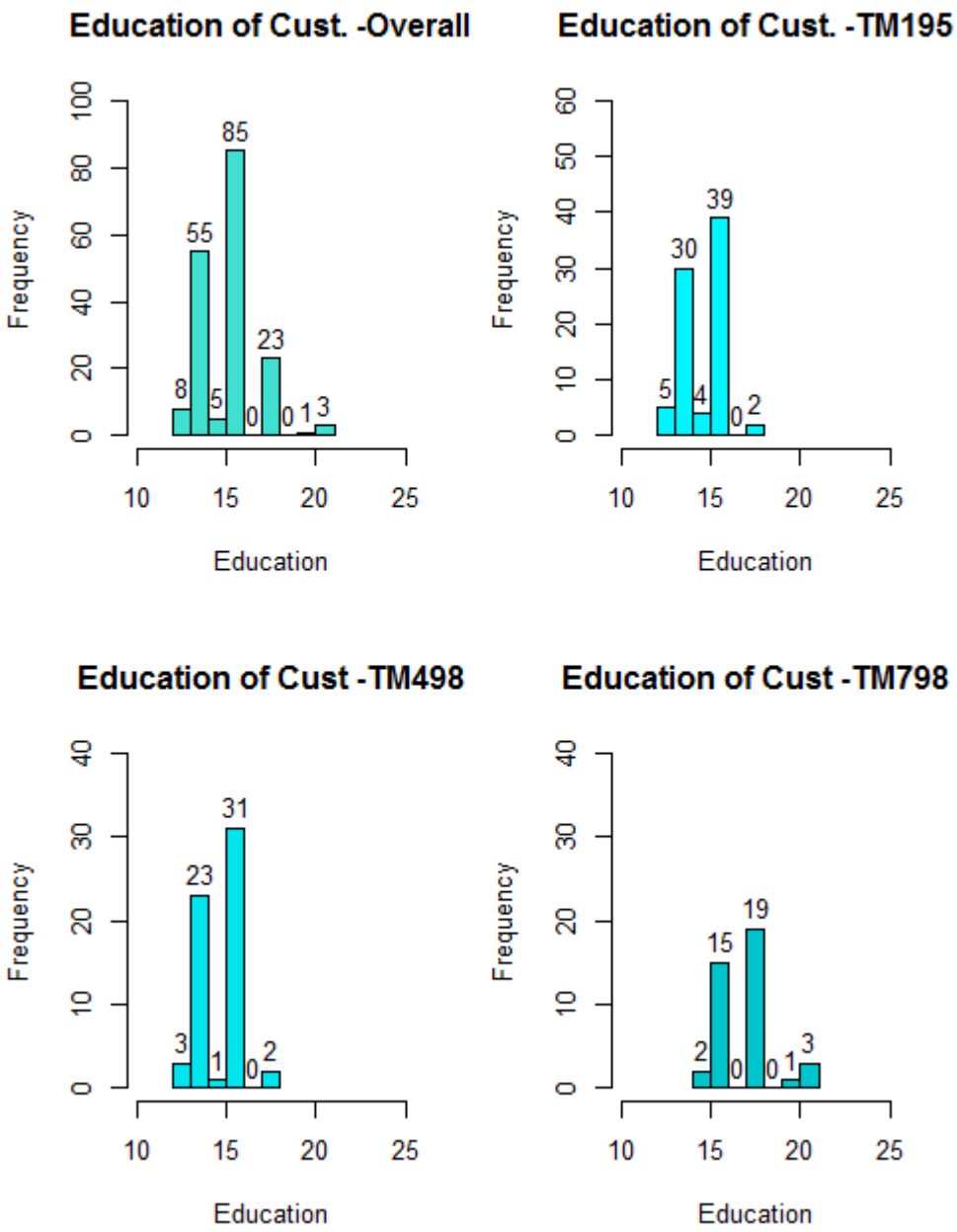
Interpretation:

- The gender profile of products – TM195 & TM498 seems to be almost equally balanced between males and females, i.e. equal proportion of males and females as customers.
- TM798 is dominated by male users.

3.8.3 Education

The variable 'Education' is a continuous varaible.

Lets use Histogram and the measures of dispersion (Mean, Standard Deviation and Variance) to explore this variable.



Mean	Standard Deviation	Variance	Remark
15.57	1.62	2.61	Education Stats for All Products
15.04	1.22	1.48	Education Stats for TM195
15.12	1.22	1.49	Education Stats for TM498
17.32	1.64	2.69	Education Stats for TM798

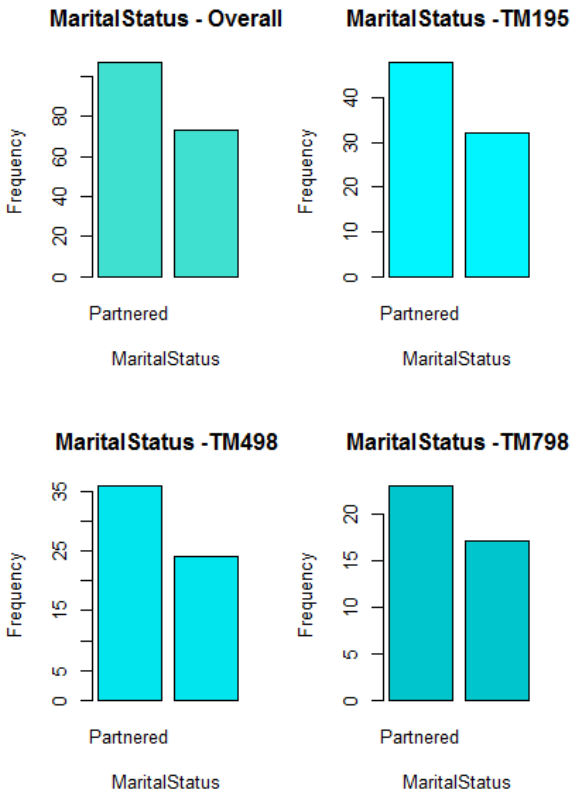
Interpretation:

- The majority of the treadmill customers had education in years above 15 years across product types – TM195, TM498 & TM798.
- TM195 & TM498 had 49% & 45% of their customers with up to 15 years of education respectively.
- TM798 had predominantly customers with more than 15 years of education (95%).
- TM798 had relatively higher average number of years in education by customers in comparison to TM195 & TM498.
    - Also, TM798 had relatively higher variation in the data distribution in comparison to TM195 & TM498.

3.8.4 Marital Status

The variable 'MaritalStatus' is a categorical varaible.

Lets use barplot and Frequency Table to explore the details.



Product	Partnered	Single
Overall	107	73
TM195	48	32
TM498	36	24
TM798	23	17

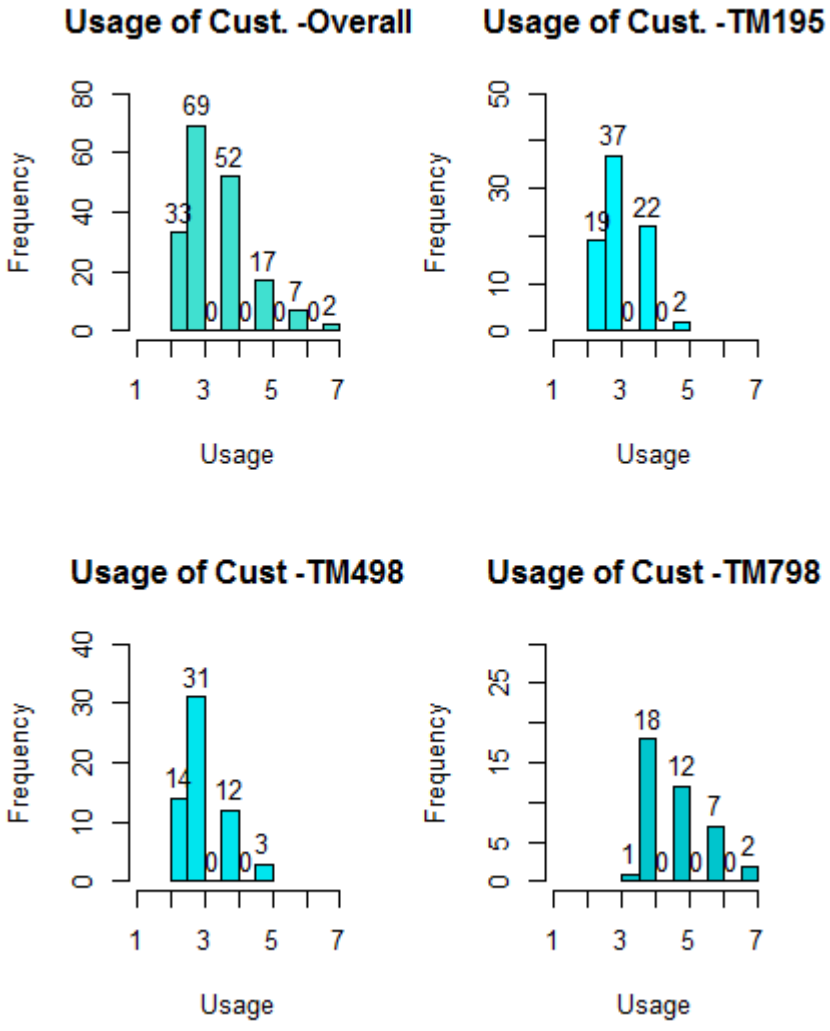
Interpretation:

- Majority of the customers had marital status as 'Partnered' across product types – TM195, TM498 & TM 798.

3.8.5 Usage

The variable 'Usage is a continuous varaible.

Lets use Histogram and the measures of dispersion (Mean, Standard Deviation and Variance) to explore this variable.



Mean	Standard Deviation	Variance	Remark
3.46	1.08	1.18	Usage Stats for All Products
3.09	0.78	0.61	Usage Stats for TM195
3.07	0.8	0.64	Usage Stats for TM498
4.78	0.95	0.9	Usage Stats for TM798

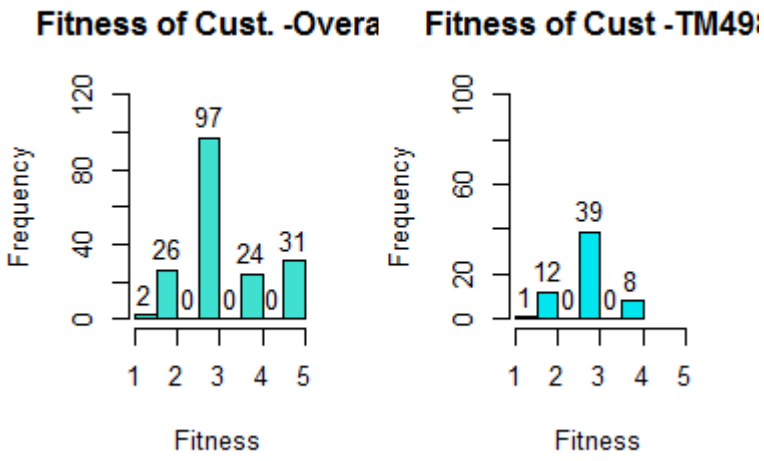
Interpretation:

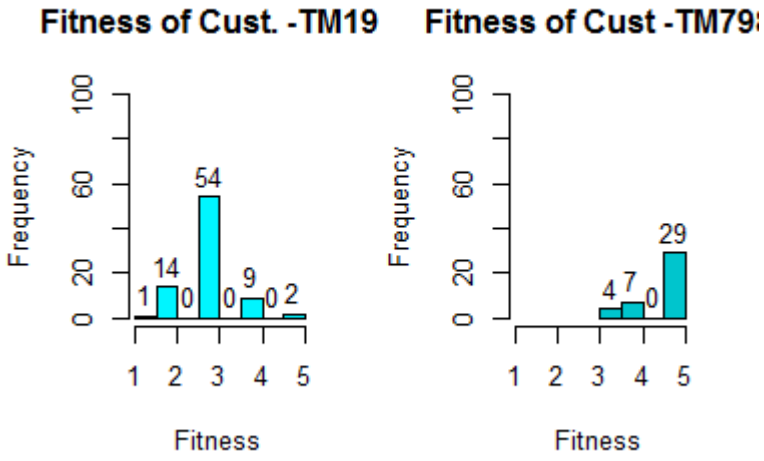
- Majority of TM195 & TM498 customers plan to use the treadmill upto 3 times a week on an average
- Almost cent percent of the TM798 customers plan to use the treadmill 4 times or more a week on an average
- TM798 customers had relatively higher intention to use treadmill (in terms of average number of times using the treadmill per week) in comparison to TM195 & TM498 customers
    - Also, TM798 had relatively higher variation in the data distribution in comparison to TM195 & TM498.

3.8.6 Fitness

The variable 'Fitness' is a continuous varaible.

Lets use Histogram and the measures of dispersion (Mean, Standard Deviation and Variance) to explore this variable.





Mean	Standard Deviation	Variance	Remark
3.31	0.96	0.92	Fitness Stats for All Products
2.96	0.66	0.44	Fitness Stats for TM195
2.9	0.63	0.4	Fitness Stats for TM498
4.62	0.67	0.45	Fitness Stats for TM798

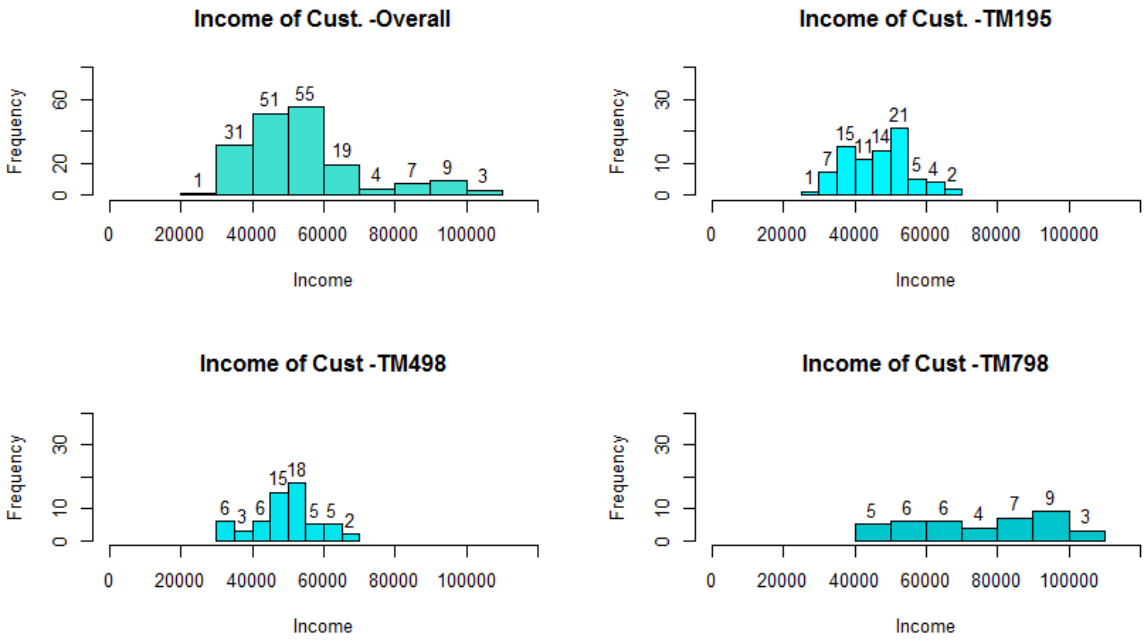
Interpretation:

- Majority of TM195 & TM498 customers plan to cover upto 3 miles on the treadmill per week on an average.
- Majority of TM798 customers plan to cover 4 miles or more on the treadmill per week on an average.
- TM798 customers had relatively higher fitness goals (in terms of average number of miles covered on treadmill per week) in comparison to TM195 & TM498 customers.

3.8.7 Income

The variable 'Income is a continuous varaible.

Lets use Histogram and the measures of dispersion (Mean, Standard Deviation and Variance) to explore this variable.



Mean	Standard Deviation	Variance	Remark
53719.58	16506.68	272470624.1	Income Stats for All Products
46418.03	9075.78	82369840.51	Income Stats for TM195
48973.65	8653.99	74891532.33	Income Stats for TM498
75441.57	18505.84	342465992.7	Income Stats for TM798

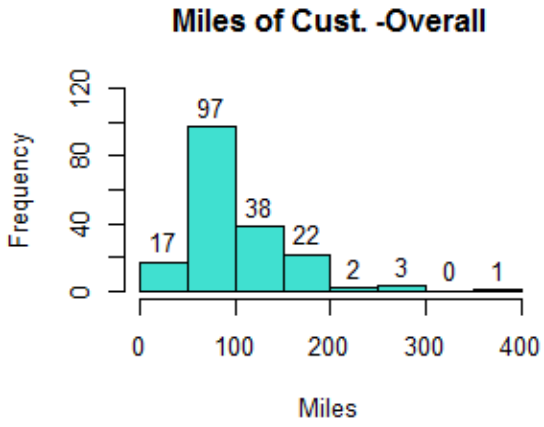
Interpretation:

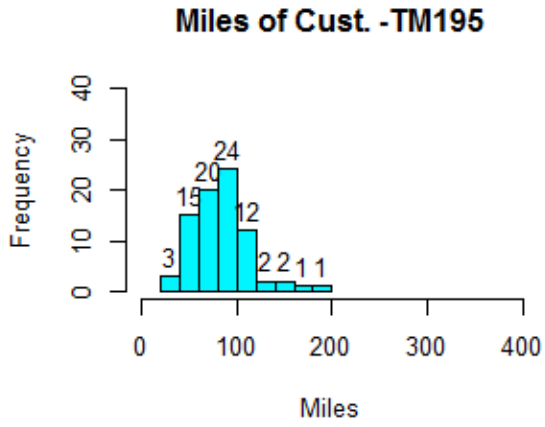
- Almost 3/4th customer of the products (TM195, TM498 & TM798) have annual household income in the following range:
  - 76% of the TM195 customers have annual household income between $35,000 - $55,000
  - 73% of the TM498 customers have annual household income between $40,000 - $60,000
  - 72% of the TM798 customers have annual household income of more than $60,000
- TM798 customers had relatively higher average annual household income in comparison to TM195 & TM498 customers
  - Also, TM798 had relatively higher variation in the data distribution in comparison to TM195 & TM498

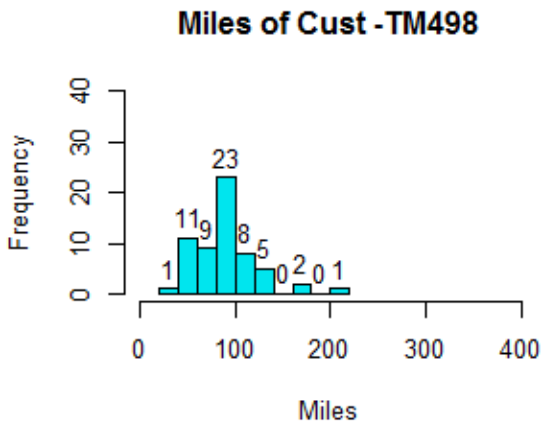
3.8.8 Total Distance Covered – Miles

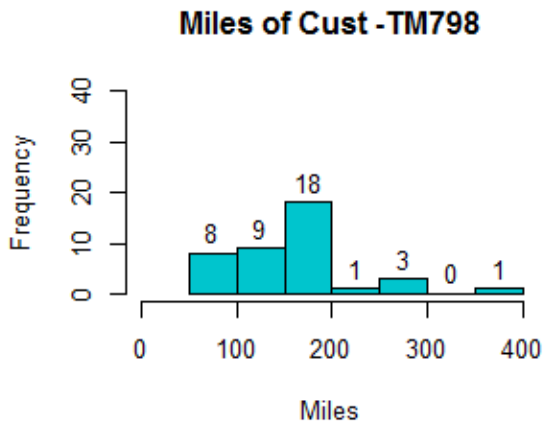
The variable 'Miles' is a continuous varaible.

Lets use Histogram and the measures of dispersion (Mean, Standard Deviation and Variance) to explore this variable.









Mean	Standard Deviation	Variance	Remark
103.19	51.86	2689.83	Miles Stats for All Products
82.79	28.87	833.71	Miles Stats for TM195
87.93	33.26	1106.44	Miles Stats for TM498
166.9	60.07	3607.99	Miles Stats for TM798

Interpretation:

- Almost 3 out of 4 TM195 & TM498 customers have covered upto 100 miles on treadmill.
- Nearly cent percent of TM798 customers have covered 100 miles or more on treadmill.
- TM798 customers had relatively higher average distance covered on treadmill in comparison to TM195 & TM498 customers.
- Also, TM798 had relatively higher variation in the data distribution in comparison to TM195 & TM498.

4 Conclusion

4.1 Outliers in the Data:

- Outliers found in the data for Age, Education, Usage, Income, and Miles.
- Further investigation required as how these outliers should be treated.

4.2 Gender:

- The collected data is Male dominated (104 Vs 76). Hence precaution should be taken while interpreting the results for Male Vs. Female.
- Usage of the Product: Level 7 is popular among Male.
- It is observed that Male candidate have slightly higher income than Female.
- Also the Miles burning rate is higher in Male than Female. Looking at the Product Features, we may conclude that Male candidates can be targeted for product M798.

4.3 Marital Status:

- The collected data is dominated with Partners (107 Vs 73). Hence precaution should be taken while interpreting the results for Partners Vs. Single.
- Partnered have shown more usage of the Fitness Centre compared to Singles. This is significant insight for target marketing.
- Partnered persons also tend to burn more miles. This suggests that the promotion of products TM498 and TM798 can be penetrated more on them.

4.4 Product:

- TM195 is popular among all age groups, whereas TM798 is popular among 22-47 age group. This is useful insight for target marketing of TM798.
- TM195 seems to be an entry level product, as it's been used across various Fitness levels whereas TM798 seems to be a specialized product, as it's being used for higher fitness levels. (3+ Levels)
- TM798 also should be targeted among highly educated people.
- A new comer to the fitness club may safely start using TM195.
- TM195 and TM498 are popular among low and medium income group, whereas TM798 is popular among medium to high income group. Another indicator for Target Marketing.
- If one needs to achieve more miles, then TM798 is the product for him/her.
- Significance of Usage Level need to understand to provide further inputs on this feature.

4.5 Education:

- Highly educated people tend to earn more income. This information can be used for cross selling.
- Persons at education level 16 tend to burn more miles. Such are the ideal candidates for product TM798.

4.6 Usage:

- Persons with higher fitness level tend to use the fitness centre more.
- Higher Income group has shown more usage. This information can be used for promotional activities / cross selling.
- Persons with increasing income has shown good fitness and inclination towards more usage of the fitness centre.

5 Appendix A – Source Code

```
#=============================================================================
#
#   Exploratory Data Analysis - CardioFitness
#
#=============================================================================
# Environment Set up and Data Import
#=============================================================================
# Install Packages
#=============================================================================
#
# Install the necessary packages in this section, including libraries.
# Having all packages and libraries at one place makes the code readable.
# For example, if ggplot2 is needed, following is the sample code:
#install.packages("ggplot2")
#library("ggplot2")
#
# Setup Working Directory
setwd("D:/BACP Mini Project")
getwd()

#
# Read Input File
cgf_data=read.csv("CardioGoodFitness.csv")
attach(cgf_data)
#
# Find out Total Number of Rows and Columns
dim(cgf_data)

## [1] 180   9

# Find out Names of the Columns (Features)
names(cgf_data)

## [1] "Product"       "Age"           "Gender"        "Education"
## [5] "MaritalStatus" "Usage"         "Fitness"       "Income"
## [9] "Miles"

# Find out Class of each Feature, along with internal structure
str(cgf_data)

## 'data.frame':    180 obs. of  9 variables:
##  $ Product      : Factor w/ 3 levels "TM195","TM498",..: 1 1 1 1 1 1 1
1 1 1 ...
##  $ Age          : int  18 19 19 19 20 20 21 21 21 21 ...
##  $ Gender       : Factor w/ 2 levels "Female","Male": 2 2 1 2 2 1 1 2 2
1 ...
##  $ Education    : int  14 15 14 12 13 14 14 13 15 15 ...
##  $ MaritalStatus: Factor w/ 2 levels "Partnered","Single": 2 2 1 2 1 1
1 2 2 1 ...
##  $ Usage        : int  3 2 4 3 4 3 3 3 5 2 ...
##  $ Fitness      : int  4 3 3 3 2 3 3 3 4 3 ...
##  $ Income       : int  29562 31836 30699 32973 35247 32973 35247 32973
```

```
35247 37521 ...
##  $ Miles        : int  112 75 66 85 47 66 75 85 141 85 ...

#
# Check top 6 and bottom 6 Rows of the Dataset
head(cgf_data)

##   Product Age Gender Education MaritalStatus Usage Fitness Income Miles
## 1   TM195  18   Male        14        Single     3       4  29562   112
## 2   TM195  19   Male        15        Single     2       3  31836    75
## 3   TM195  19 Female        14     Partnered     4       3  30699    66
## 4   TM195  19   Male        12        Single     3       3  32973    85
## 5   TM195  20   Male        13     Partnered     4       2  35247    47
## 6   TM195  20 Female        14     Partnered     3       3  32973    66

tail(cgf_data)

##     Product Age Gender Education MaritalStatus Usage Fitness Income
Miles
## 175   TM798  38   Male        18     Partnered     5       5 104581
150
## 176   TM798  40   Male        21        Single     6       5  83416
200
## 177   TM798  42   Male        18        Single     5       4  89641
200
## 178   TM798  45   Male        16        Single     5       5  90886
160
## 179   TM798  47   Male        18     Partnered     4       5 104581
120
## 180   TM798  48   Male        18     Partnered     4       5  95508
180

# head(cgf_data,10) # To obtain desired number of rows, here 10.

#
#Check for Missing Values
colSums(is.na(cgf_data))

##       Product           Age        Gender     Education MaritalStatus
##             0             0             0             0             0
##         Usage       Fitness        Income         Miles
##             0             0             0             0

#
# Provide Summary of a Dataset.
summary(cgf_data)

##   Product        Age           Gender      Education
MaritalStatus
##  TM195:80   Min.   :18.00   Female: 76   Min.   :12.00   Partnered:107
##  TM498:60   1st Qu.:24.00   Male  :104   1st Qu.:14.00   Single   : 73
##  TM798:40   Median :26.00                Median :16.00
##             Mean   :28.79                Mean   :15.57
##             3rd Qu.:33.00                3rd Qu.:16.00
##             Max.   :50.00                Max.   :21.00
```

```
##      Usage           Fitness          Income           Miles
##  Min.   :2.000   Min.   :1.000   Min.   : 29562   Min.   : 21.0
##  1st Qu.:3.000   1st Qu.:3.000   1st Qu.: 44059   1st Qu.: 66.0
##  Median :3.000   Median :3.000   Median : 50597   Median : 94.0
##  Mean   :3.456   Mean   :3.311   Mean   : 53720   Mean   :103.2
##  3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.: 58668   3rd Qu.:114.8
##  Max.   :7.000   Max.   :5.000   Max.   :104581   Max.   :360.0

#
# Check all values of a Feature with it's frequencies.
table(Product)

## Product
## TM195 TM498 TM798
##    80    60    40

table(Gender)

## Gender
## Female   Male
##     76    104

table(MaritalStatus)

## MaritalStatus
## Partnered    Single
##       107        73

table(Age)

## Age
## 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41
42
##  1  4  5  7  7 18 12 25 12  7  9  6  7  6  4  8  6  8  1  2  7  1  5  1
1
## 43 44 45 46 47 48 50
##  1  1  2  1  2  2  1

table(Education)

## Education
## 12 13 14 15 16 18 20 21
##  3  5 55  5 85 23  1  3

table(Usage)

## Usage
##  2  3  4  5  6  7
## 33 69 52 17  7  2

table(Fitness)

## Fitness
##  1  2  3  4  5
##  2 26 97 24 31

table(Income)
```

```
## Income
##  29562  30699  31836  32973  34110  35247  36384  37521  38658  39795
##      1      1      2      5      5      5      4      2      5      2
##  40932  42069  43206  44343  45480  46617  47754  48556  48658  48891
##      6      2      5      4     14      8      2      2      1      5
##  49801  50028  51165  52290  52291  52302  53439  53536  54576  54781
##      2      7      7      1      1      9      8      1      8      1
##  55713  56850  57271  57987  58516  59124  60261  61006  61398  62251
##      1      2      1      4      1      3      3      2      2      1
##  62535  64741  64809  65220  67083  68220  69721  70966  74701  75946
##      1      2      3      1      2      1      1      1      1      1
##  77191  83416  85906  88396  89641  90886  92131  95508  95866  99601
##      1      2      1      2      2      3      3      1      1      1
## 103336 104581
##      1      2

table(Miles)

## Miles
##  21  38  42  47  53  56  64  66  74  75  80  85  94  95 100 103 106 112
##   1   3   4   9   7   6   6  10   3  10   1  27   8  12   7   3   9   1
## 113 120 127 132 140 141 150 160 169 170 180 188 200 212 240 260 280 300
##   8   3   5   2   1   2   4   5   1   3   6   1   6   1   1   1   1   1
## 360
##   1

#
#=============================================================================
# Variable Transformation / Feature Creation
#=============================================================================
#
AgeGroup <- ifelse(Age<26, "Early Youth",
                   ifelse(Age>35, "Middle Aged Adults",
                          "Late Youth"))
EducationLevel <- ifelse(Education<13, "Higher Secondary",
                         ifelse(Education>16, "Masters/ Post Graduation",
                                "Graduation"))
Usagelevel <- ifelse(Usage<4, "Fitness Amateurs",
                     ifelse(Usage>5, "Fitness Freaks",
                            "Fitness Regulars"))
FitnessLevel <- ifelse(Fitness<3, "Low Intensity",
                       ifelse(Fitness>4, "High Intensity",
                              "Medium Intensity"))
HHIncome <- ifelse(Income<40001, "Low Income HH",
                   ifelse(Income>60000, "High Income HH",
                          "Medium Income HH"))
TotalDistance <- ifelse(Miles<61, "Low Usage",
                        ifelse(Miles>120, "High Usage",
                               "Medium Usage"))
#
# Append the newly created features with the original Data
cgf_data_1 <- cbind(cgf_data, AgeGroup, EducationLevel,
                    Usagelevel, FitnessLevel, HHIncome, TotalDistance)
#
```

```
# Check dimensions of the newly created data,
# Sample Records and Summary Statistics
#
dim(cgf_data_1)

## [1] 180  15

head(cgf_data_1)

tail(cgf_data_1)

str(cgf_data_1)

summary(cgf_data_1)

#
# Create Product wise three Subsets of the modified dataset
#
product1 <- cgf_data_1[which(cgf_data_1$Product == "TM195"),]
product2 <- cgf_data_1[which(cgf_data_1$Product == "TM498"),]
product3 <- cgf_data_1[which(cgf_data_1$Product == "TM798"),]
#
#=============================================================================
# Feature Exploration
#=============================================================================
#
# Data Visualization using Graphs:
# Generic Plot for Categorical variables
# Histogram for Continuous Variables
# Box Plot to see the Outliers in Continuous Variables

#Create Partitions in the Panel
#
par(mfrow=c(1,3))
plot(Product,main='Product',xlab = "Product",
     ylab = "Frequency",col = "turquoise")
plot(Gender,main='Gender',xlab = "Gender",
     ylab = "Frequency",col = "turquoise1")
plot(MaritalStatus,main='Marital Status',xlab = "Marital Status",
     ylab = "Frequency",col = "turquoise2")


#
dev.off() # To Reset the earlier partition command.

par(mfrow=c(2,3))
hist(Age,main='Age',xlab = "Age",ylab = "Frequency",col = "turquoise")
hist(Education,main='Education',xlab = "Education",
     ylab = "Frequency",col = "turquoise1")
hist(Usage,main='Usage',xlab = "Usage",
     ylab = "Frequency",col = "turquoise2")
#
boxplot(Age,main='Age',xlab = "Age",
        ylab = "Frequency",col = "turquoise",horizontal = TRUE)
```

```
boxplot(Education,main='Education',xlab = "Education",
        ylab = "Frequency",col = "turquoise1",horizontal = TRUE)
boxplot(Usage,main='Usage',xlab = "Usage",
        ylab = "Frequency",col = "turquoise2",horizontal = TRUE)
#
hist(Fitness,main='Fitness',xlab = "Fitness",
     ylab = "Frequency",col = "turquoise")
hist(Income,main='Income',xlab = "Income",
     ylab = "Frequency",col = "turquoise1")
hist(Miles,main='Miles',xlab = "Miles",
     ylab = "Frequency",col = "turquoise2")
#
boxplot(Fitness,main='Fitness',xlab = "Fitness",
        ylab = "Frequency",col = "turquoise",horizontal = TRUE)
boxplot(Income,main='Income',xlab = "Income",
        ylab = "Frequency",col = "turquoise1",horizontal = TRUE)
boxplot(Miles,main='Miles',xlab = "Miles",
        ylab = "Frequency",col = "turquoise2",horizontal = TRUE)
#
#=============================================================================
# Feature Exploration - Age
#=============================================================================
#
dev.off() # To Reset the earlier partition command.

par(mfrow=c(2,2))
hist(cgf_data_1$Age, main="Age of the Customer-Overall",
     xlab="Age", ylab="Frequency", labels=TRUE,
     col="turquoise", xlim=c(10,50), ylim=c(0,150))
hist(product1$Age, main="Age of the Customer-TM195",
     xlab="Age", ylab="Frequency", labels=TRUE,
     col="turquoise1", xlim=c(10,50), ylim=c(0,150))
hist(product2$Age, main="Age of the Customer-TM498",
     xlab="Age", ylab="Frequency", labels=TRUE,
     col="turquoise2", xlim=c(10,50), ylim=c(0,150))
hist(product3$Age, main="Age of the Customer-TM798",
     xlab="Age", ylab="Frequency", labels=TRUE,
     col="turquoise3", xlim=c(10,50), ylim=c(0,150))
#
#=============================================================================
# Mean, Standard Deviation and Variance
#=============================================================================
# Normal mean(),sd(),var() Functions are used.
# round() function is used with parameter digits,
# to display the results upto two decimal places
#=============================================================================
#
Col_Head <- c("Mean","Standard Deviation","Variance","Remark")
Prod0_Stats <- c(round(mean(cgf_data_1$Age),digits = 2),
                 round(sd(cgf_data_1$Age),digits = 2),
                 round(var(cgf_data_1$Age),digits = 2),
                 "Age Stats for all Products")
Prod1_Stats <- c(round(mean(product1$Age),digits = 2),
                 round(sd(product1$Age),digits = 2),
```

```
                  round(var(product1$Age),digits = 2),
                  "Age Stats for TM195")
Prod2_Stats <- c(round(mean(product2$Age),digits = 2),
                  round(sd(product2$Age),digits = 2),
                  round(var(product2$Age),digits = 2),
                  "Age Stats for TM498")
Prod3_Stats <- c(round(mean(product3$Age),digits = 2),
                  round(sd(product3$Age),digits = 2),
                  round(var(product3$Age),digits = 2),
                  "Age Stats for TM798")
#
Age_Stats <- rbind(Col_Head,Prod0_Stats,Prod1_Stats,
                   Prod2_Stats,Prod3_Stats)
Age_Stats
```

```
##             [,1]    [,2]                 [,3]
## Col_Head    "Mean"  "Standard Deviation" "Variance"
## Prod0_Stats "28.79" "6.94"               "48.21"
## Prod1_Stats "28.55" "7.22"               "52.15"
## Prod2_Stats "28.9"  "6.65"               "44.16"
## Prod3_Stats "29.1"  "6.97"               "48.61"
##             [,4]
## Col_Head    "Remark"
## Prod0_Stats "Age Stats for all Products"
## Prod1_Stats "Age Stats for TM195"
## Prod2_Stats "Age Stats for TM498"
## Prod3_Stats "Age Stats for TM798"
```

```
#
#=============================================================================
# Feature Exploration - Gender
#=============================================================================
#
dev.off() # To Reset the earlier partition command.

par(mfrow=c(2,2))
plot(cgf_data_1$Gender,main='Gender of the Customer-Overall',
     xlab = "Gender",ylab = "Frequency",col = "turquoise")
plot(product1$Gender, main='Gender of the Customer-TM195',
     xlab="Gender", ylab="Frequency", col="turquoise1")
plot(product2$Gender, main="Gender of the Customer-TM498",
     xlab="Gender", ylab="Frequency", col="turquoise2")
plot(product3$Gender, main="Gender of the Customer-TM798",
     xlab="Gender", ylab="Frequency", col="turquoise3")
#
#=============================================================================
# Frequency Table
#=============================================================================
#
table(cgf_data_1$Gender)

table(product1$Gender)

table(product2$Gender)
```

```
table(product3$Gender)

#
#=============================================================================
# Feature Exploration - Education
#=============================================================================
#
dev.off() # To Reset the earlier partition command.

par(mfrow=c(2,2))
hist(cgf_data_1$Education, main="Education of Cust. -Overall",
     xlab="Education", ylab="Frequency", labels=TRUE,
     col="turquoise", xlim=c(10,25), ylim=c(0,100))
hist(product1$Education, main="Education of Cust. -TM195",
     xlab="Education", ylab="Frequency", labels=TRUE,
     col="turquoise1", xlim=c(10,25), ylim=c(0,60))
hist(product2$Education, main="Education of Cust -TM498",
     xlab="Education", ylab="Frequency", labels=TRUE,
     col="turquoise2", xlim=c(10,25), ylim=c(0,40))
hist(product3$Education, main="Education of Cust -TM798",
     xlab="Education", ylab="Frequency", labels=TRUE,
     col="turquoise3", xlim=c(10,25), ylim=c(0,40))
#
#=============================================================================
# Mean, Standard Deviation and Variance
#=============================================================================
# Normal mean(),sd(),var() Functions are used.
# round() function is used with parameter digits,
# to display the results upto two decimal places
#=============================================================================
#
Col_Head <- c("Mean","Standard Deviation","Variance","Remark")
Prod0_Stats <- c(round(mean(cgf_data_1$Education),digits = 2),
                 round(sd(cgf_data_1$Education),digits = 2),
                 round(var(cgf_data_1$Education),digits = 2),
                 "Education Stats for all Products")
Prod1_Stats <- c(round(mean(product1$Education),digits = 2),
                 round(sd(product1$Education),digits = 2),
                 round(var(product1$Education),digits = 2),
                 "Education Stats for TM195")
Prod2_Stats <- c(round(mean(product2$Education),digits = 2),
                 round(sd(product2$Education),digits = 2),
                 round(var(product2$Education),digits = 2),
                 "Education Stats for TM498")
Prod3_Stats <- c(round(mean(product3$Education),digits = 2),
                 round(sd(product3$Education),digits = 2),
                 round(var(product3$Education),digits = 2),
                 "Education Stats for TM798")
#
Education_Stats <- rbind(Col_Head,Prod0_Stats,Prod1_Stats,
                         Prod2_Stats,Prod3_Stats)
Education_Stats
```

```
#
#=============================================================================
# Feature Exploration - MaritalStatus
#=============================================================================
#
dev.off() # To Reset the earlier partition command.

par(mfrow=c(2,2))
plot(cgf_data_1$MaritalStatus,main='MaritalStatus - Overall',
     xlab = "MaritalStatus",ylab = "Frequency",col = "turquoise")
plot(product1$MaritalStatus, main='MaritalStatus -TM195',
     xlab="MaritalStatus", ylab="Frequency", col="turquoise1")
plot(product2$MaritalStatus, main="MaritalStatus -TM498",
     xlab="MaritalStatus", ylab="Frequency", col="turquoise2")
plot(product3$MaritalStatus, main="MaritalStatus -TM798",
     xlab="MaritalStatus", ylab="Frequency", col="turquoise3")
#
#=============================================================================
# Frequency Table
#=============================================================================
#
table(cgf_data_1$MaritalStatus)

table(product1$MaritalStatus)

table(product2$MaritalStatus)

table(product3$MaritalStatus)

#
#=============================================================================
# Feature Exploration - Usage
#=============================================================================
#
dev.off() # To Reset the earlier partition command.

par(mfrow=c(2,2))
hist(cgf_data_1$Usage, main="Usage of Cust. -Overall",
     xlab="Usage", ylab="Frequency", labels=TRUE,
     col="turquoise", xlim=c(1,7), ylim=c(0,80))
hist(product1$Usage, main="Usage of Cust. -TM195",
     xlab="Usage", ylab="Frequency", labels=TRUE,
     col="turquoise1", xlim=c(1,7), ylim=c(0,50))
hist(product2$Usage, main="Usage of Cust -TM498",
     xlab="Usage", ylab="Frequency", labels=TRUE,
     col="turquoise2", xlim=c(1,7), ylim=c(0,40))
hist(product3$Usage, main="Usage of Cust -TM798",
     xlab="Usage", ylab="Frequency", labels=TRUE,
     col="turquoise3", xlim=c(1,7), ylim=c(0,30))
#
#=============================================================================
# Mean, Standard Deviation and Variance
#=============================================================================
# Normal mean(),sd(),var() Functions are used.
# round() function is used with parameter digits,
```

```
# to display the results upto two decimal places
#=============================================================================
#
Col_Head <- c("Mean","Standard Deviation","Variance","Remark")
Prod0_Stats <- c(round(mean(cgf_data_1$Usage),digits = 2),
                 round(sd(cgf_data_1$Usage),digits = 2),
                 round(var(cgf_data_1$Usage),digits = 2),
                 "Usage Stats for all Products")
Prod1_Stats <- c(round(mean(product1$Usage),digits = 2),
                 round(sd(product1$Usage),digits = 2),
                 round(var(product1$Usage),digits = 2),
                 "Usage Stats for TM195")
Prod2_Stats <- c(round(mean(product2$Usage),digits = 2),
                 round(sd(product2$Usage),digits = 2),
                 round(var(product2$Usage),digits = 2),
                 "Usage Stats for TM498")
Prod3_Stats <- c(round(mean(product3$Usage),digits = 2),
                 round(sd(product3$Usage),digits = 2),
                 round(var(product3$Usage),digits = 2),
                 "Usage Stats for TM798")
#
Usage_Stats <- rbind(Col_Head,Prod0_Stats,Prod1_Stats,
                     Prod2_Stats,Prod3_Stats)
Usage_Stats

#
#=============================================================================
# Feature Exploration - Fitness
#=============================================================================
#
dev.off() # To Reset the earlier partition command.

par(mfrow=c(2,2))
hist(cgf_data_1$Fitness, main="Fitness of Cust. -Overall",
     xlab="Fitness", ylab="Frequency", labels=TRUE,
     col="turquoise", xlim=c(1,5), ylim=c(0,120))
hist(product1$Fitness, main="Fitness of Cust. -TM195",
     xlab="Fitness", ylab="Frequency", labels=TRUE,
     col="turquoise1", xlim=c(1,5), ylim=c(0,100))
hist(product2$Fitness, main="Fitness of Cust -TM498",
     xlab="Fitness", ylab="Frequency", labels=TRUE,
     col="turquoise2", xlim=c(1,5), ylim=c(0,100))
hist(product3$Fitness, main="Fitness of Cust -TM798",
     xlab="Fitness", ylab="Frequency", labels=TRUE,
     col="turquoise3", xlim=c(1,5), ylim=c(0,100))
#
#=============================================================================
# Mean, Standard Deviation and Variance
#=============================================================================
# Normal mean(),sd(),var() Functions are used.
# round() function is used with parameter digits,
# to display the results upto two decimal places
#=============================================================================
#
```

```
Col_Head <- c("Mean","Standard Deviation","Variance","Remark")
Prod0_Stats <- c(round(mean(cgf_data_1$Fitness),digits = 2),
                 round(sd(cgf_data_1$Fitness),digits = 2),
                 round(var(cgf_data_1$Fitness),digits = 2),
                 "Fitness Stats for all Products")
Prod1_Stats <- c(round(mean(product1$Fitness),digits = 2),
                 round(sd(product1$Fitness),digits = 2),
                 round(var(product1$Fitness),digits = 2),
                 "Fitness Stats for TM195")
Prod2_Stats <- c(round(mean(product2$Fitness),digits = 2),
                 round(sd(product2$Fitness),digits = 2),
                 round(var(product2$Fitness),digits = 2),
                 "Fitness Stats for TM498")
Prod3_Stats <- c(round(mean(product3$Fitness),digits = 2),
                 round(sd(product3$Fitness),digits = 2),
                 round(var(product3$Fitness),digits = 2),
                 "Fitness Stats for TM798")
#
Fitness_Stats <- rbind(Col_Head,Prod0_Stats,Prod1_Stats,
                       Prod2_Stats,Prod3_Stats)
Fitness_Stats

#
#=============================================================================
# Feature Exploration - Annual Household Income
#=============================================================================
#
dev.off() # To Reset the earlier partition command.

par(mfrow=c(2,2))
hist(cgf_data_1$Income, main="Income of Cust. -Overall",
     xlab="Income", ylab="Frequency", labels=TRUE,
     col="turquoise", xlim=c(0,120000), ylim=c(0,80))
hist(product1$Income, main="Income of Cust. -TM195",
     xlab="Income", ylab="Frequency", labels=TRUE,
     col="turquoise1", xlim=c(0,120000), ylim=c(0,40))
hist(product2$Income, main="Income of Cust -TM498",
     xlab="Income", ylab="Frequency", labels=TRUE,
     col="turquoise2", xlim=c(0,120000), ylim=c(0,40))
hist(product3$Income, main="Income of Cust -TM798",
     xlab="Income", ylab="Frequency", labels=TRUE,
     col="turquoise3", xlim=c(0,120000), ylim=c(0,40))
#
#=============================================================================
# Mean, Standard Deviation and Variance
#=============================================================================
# Normal mean(),sd(),var() Functions are used.
# round() function is used with parameter digits,
# to display the results upto two decimal places
#=============================================================================
#
Col_Head <- c("Mean","Standard Deviation","Variance","Remark")
Prod0_Stats <- c(round(mean(cgf_data_1$Income),digits = 2),
                 round(sd(cgf_data_1$Income),digits = 2),
```

```
                  round(var(cgf_data_1$Income),digits = 2),
                  "Income Stats for all Products")
Prod1_Stats <- c(round(mean(product1$Income),digits = 2),
                  round(sd(product1$Income),digits = 2),
                  round(var(product1$Income),digits = 2),
                  "Income Stats for TM195")
Prod2_Stats <- c(round(mean(product2$Income),digits = 2),
                  round(sd(product2$Income),digits = 2),
                  round(var(product2$Income),digits = 2),
                  "Income Stats for TM498")
Prod3_Stats <- c(round(mean(product3$Income),digits = 2),
                  round(sd(product3$Income),digits = 2),
                  round(var(product3$Income),digits = 2),
                  "Income Stats for TM798")
#
Income_Stats <- rbind(Col_Head,Prod0_Stats,Prod1_Stats,
                      Prod2_Stats,Prod3_Stats)
Income_Stats

#
#=============================================================================
# Feature Exploration - Total Distance Covered - Miles
#=============================================================================
#
dev.off() # To Reset the earlier partition command.

par(mfrow=c(2,2))
hist(cgf_data_1$Miles, main="Miles of Cust. -Overall",
     xlab="Miles", ylab="Frequency", labels=TRUE,
     col="turquoise", xlim=c(0,400), ylim=c(0,120))
hist(product1$Miles, main="Miles of Cust. -TM195",
     xlab="Miles", ylab="Frequency", labels=TRUE,
     col="turquoise1", xlim=c(0,400), ylim=c(0,40))
hist(product2$Miles, main="Miles of Cust -TM498",
     xlab="Miles", ylab="Frequency", labels=TRUE,
     col="turquoise2", xlim=c(0,400), ylim=c(0,40))
hist(product3$Miles, main="Miles of Cust -TM798",
     xlab="Miles", ylab="Frequency", labels=TRUE,
     col="turquoise3", xlim=c(0,400), ylim=c(0,40))
#
#=============================================================================
# Mean, Standard Deviation and Variance
#=============================================================================
# Normal mean(),sd(),var() Functions are used.
# round() function is used with parameter digits,
# to display the results upto two decimal places
#=============================================================================
#
Col_Head <- c("Mean","Standard Deviation","Variance","Remark")
Prod0_Stats <- c(round(mean(cgf_data_1$Miles),digits = 2),
                  round(sd(cgf_data_1$Miles),digits = 2),
                  round(var(cgf_data_1$Miles),digits = 2),
                  "Miles Stats for all Products")
Prod1_Stats <- c(round(mean(product1$Miles),digits = 2),
```

```
                  round(sd(product1$Miles),digits = 2),
                  round(var(product1$Miles),digits = 2),
                  "Miles Stats for TM195")
Prod2_Stats <- c(round(mean(product2$Miles),digits = 2),
                  round(sd(product2$Miles),digits = 2),
                  round(var(product2$Miles),digits = 2),
                  "Miles Stats for TM498")
Prod3_Stats <- c(round(mean(product3$Miles),digits = 2),
                  round(sd(product3$Miles),digits = 2),
                  round(var(product3$Miles),digits = 2),
                  "Miles Stats for TM798")
#
Miles_Stats <- rbind(Col_Head,Prod0_Stats,Prod1_Stats,
                     Prod2_Stats,Prod3_Stats)
Miles_Stats

#
#=============================================================================
# Bi-variate Analysis
#=============================================================================
#
dev.off() # To Reset the earlier partition command.

par(mfrow=c(2,4))
plot(Product,Age, main='Product Vs Age',xlab = "Product",
     ylab = "Age",col = "Aquamarine")
plot(Product,Gender, main='Product Vs Gender',xlab = "Product",
     ylab = "Gender",col = "Turquoise")
plot(Product,Education, main='Product Vs Education',xlab = "Product",
     ylab = "Education",col = "Light Sea Green")
plot(Product,MaritalStatus, main='Product Vs Marital Status',
     xlab = "Product",ylab = "Marital Status",col = "Medium Turquoise")
plot(Product,Usage, main='Product Vs Usage',xlab = "Product",
     ylab = "Usage",col = "Medium Aquamarine")
plot(Product,Fitness, main='Product Vs Fitness',xlab = "Product",
     ylab = "Fitness",col = "Medium Sea Green")
plot(Product,Income, main='Product Vs Income',xlab = "Product",
     ylab = "Income",col = "Light Green")
plot(Product,Miles, main='Product Vs Miles',xlab = "Product",
     ylab = "Miles",col = "Dark Sea Green")
#
pairs(cgf_data,col = "dark blue")
#
# Correlation Coefficients Sample example
cor(Age,Education)

## [1] 0.2804957

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