Cardio Good Fitness

Benedict Egwuchukwu

6/12/2020

Contents

1. Project Objective

The objective of the report is to explore the cardio fitness data set ("CardioGoodFitness") in R and generate insights about the data set.

This exploration report will consist of the following:

- Importing the dataset in R
- Understanding the structure of dataset
- Exploratory data analysis using Graphical exploration and Descriptive statistics
- Insights from the dataset
- Recommendations that will help the company in targeting new customers

2. Assumptions

The dataset provided reflects the customer base of the treadmill product(s) of a retail store called Cardio Good Fitness.

The assumptions made in this report are as follows:

- Dataset is clean and has no errors in entries
- There is no relationship between independent and dependent variables
- The data has a normal distribution

3. Exploratory Data Analysis - Step by step approach

3.1 Environment Set up and Data Import

```
# Environment Set up and Data Import

# Invoking Libraries
library(readr) # To import csv files
library(ggplot2) # To create plots
library(corrplot) # To plot correlation plot between numerical variables
library(dplyr) # To manipulate dataset
library(gridExtra) # To plot multiple ggplot graphs in a grid
library(knitr) # Necessary to generate sourcecodes from a .Rmd File
library(markdown) # To convert to HTML
library(rmarkdown) # To convret analyses into high quality documents
```

3.1.1 Install necessary packages and load libraries

```
# Set working directory
setwd("C:/Users/egwuc/Desktop/PGP-DSBA-UT Austin/Introduction to R/Week 4 - Project/")
```

3.1.2 Set up working Directory

```
# Read input file
Cardio_fitness <- read_csv("CardioGoodFitness.csv")
```

3.1.3 Import and Read the Dataset

```
## Parsed with column specification:
## cols(
##   Product = col_character(),
##   Age = col_double(),
##   Gender = col_character(),
##   Education = col_double(),
##   MaritalStatus = col_character(),
##   Usage = col_double(),
##   Fitness = col_double(),
##   Income = col_double(),
##   Miles = col_double()
## )
```

```
# Global options settings
options(scipen = 999) # turn off scientific notation like 1e+06
```

3.1.4 Global options settings

3.2 Variable Identification

In order for us to get familiar with the Cardio Good Fitness data, we would be using the following functions to get an overview

1. dim(): this gives us the dimension of the dataset provided. Knowing the data dimension gives us an idea of how large the data is. 2. head(): this shows the first 6 rows(observations) of the dataset. It is

essential for us to get a glimpse of the dataset in a tabular format without revealing the entire dataset if we are to properly analyse the data.

2. tail(): this shows the last 6 rows(observations) of the dataset. Knowing what the dataset looks like at the end rows also helps us ensure the data is consistent.
3. str(): this shows us the structure of the dataset. It helps us determine the datatypes of the features and identify if there are datatype mismatches, so that we handle these ASAP to avoid inappropriate results from our analysis.
4. summary(): this provides statistical summaries of the dataset. This function is important as we can quickly get statistical summaries (mean,median, quartiles, min, frequencies/counts, max values etc.) which can help us derive insights even before diving deep into the data.
5. View(): helps to look at the entire dataset at a glance.

3.2.1 Variable Identification - Insights

Insight(s) from dim():

```r
# Variable identification
# check dimension of dataset
dim(Cardio_fitness)
```

```
## [1] 180   9
```

- The dataset has 180 rows and 9 columns.

Insight(s) from head():

```r
# check first 6 rows(observations) of dataset
head(Cardio_fitness)
```

```
## # A tibble: 6 x 9
##   Product   Age Gender Education MaritalStatus Usage Fitness Income Miles
##   <chr>   <dbl> <chr>      <dbl> <chr>         <dbl>   <dbl>  <dbl> <dbl>
## 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
```

- The product, gender and maritalstatus variables are characters
- Age, education, usage, fitness, income and miles variables are integers
- Gender contains male and female
- MaritalStatus contains single and partnered

Insight(s) from tail():

```r
# check last 6 rows(observations) of dataset
tail(Cardio_fitness)
```

```
## # A tibble: 6 x 9
##   Product   Age Gender Education MaritalStatus Usage Fitness Income Miles
##   <chr>   <dbl> <chr>      <dbl> <chr>         <dbl>   <dbl>  <dbl> <dbl>
## 1 TM798      38 Male          18 Partnered         5       5 104581   150
## 2 TM798      40 Male          21 Single            6       5  83416   200
## 3 TM798      42 Male          18 Single            5       4  89641   200
## 4 TM798      45 Male          16 Single            5       5  90886   160
## 5 TM798      47 Male          18 Partnered         4       5 104581   120
## 6 TM798      48 Male          18 Partnered         4       5  95508   180
```

- Values in all fields are consistent in each column
- Product includes TM798 different from TM195.

Insight(s) from str():

```r
# check structure of dataset
str(Cardio_fitness)
```

```
## tibble [180 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Product      : chr [1:180] "TM195" "TM195" "TM195" "TM195" ...
##  $ Age          : num [1:180] 18 19 19 19 20 20 21 21 21 21 ...
##  $ Gender       : chr [1:180] "Male" "Male" "Female" "Male" ...
##  $ Education    : num [1:180] 14 15 14 12 13 14 14 13 15 15 ...
##  $ MaritalStatus: chr [1:180] "Single" "Single" "Partnered" "Single" ...
##  $ Usage        : num [1:180] 3 2 4 3 4 3 3 3 5 2 ...
##  $ Fitness      : num [1:180] 4 3 3 3 2 3 3 3 4 3 ...
##  $ Income       : num [1:180] 29562 31836 30699 32973 35247 ...
##  $ Miles        : num [1:180] 112 75 66 85 47 66 75 85 141 85 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Product = col_character(),
##   ..   Age = col_double(),
##   ..   Gender = col_character(),
##   ..   Education = col_double(),
##   ..   MaritalStatus = col_character(),
##   ..   Usage = col_double(),
##   ..   Fitness = col_double(),
##   ..   Income = col_double(),
##   ..   Miles = col_double()
##   .. )
```

- Product, gender and maritalstatus variables are read as character. It would not provide any meaningful insights in this format. It needs to be changed to a factor variable.
- Age, education, usage, fitness, income and miles are numeric variables as they should be.

```r
# change product, gender and maritalstatus to factor variable
Cardio_fitness$Product <- as.factor(Cardio_fitness$Product)
Cardio_fitness$Gender <- as.factor(Cardio_fitness$Gender)
Cardio_fitness$MaritalStatus <- as.factor(Cardio_fitness$MaritalStatus)
```

- Product, gender and maritalstatus are factor variables and provide more meaning to the dataset.

Insight(s) from summary():

```r
# get summary of dataset
summary(Cardio_fitness)
```

```
##   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
```

- Product consists of three equipment namely TM195, TM498 and TM798.
- The age variable ranges from 18 to 50.
- The gender varible consists of 104 males and 76 females.
- The maritalstatus consists of 73 singles and 107 partners.
- The mean and median of numeric variables are not too far apart.
- The median equals the 3rd quartile in the education variable.
- Usage is measured on a scale of 2 to 7.
- Fitness is measurre on a scale of 1 to 5, where 1 is poor and 5 is excellent.
- Both income and miles variables are highly skewed to the right. This is a clear indication of presence of outliers.
- Maximum income = 140,581 is clearly an extreme situation but it cannot be ruled out.

Insight(s) from View():

```
# View the dataset
View(Cardio_fitness)
```

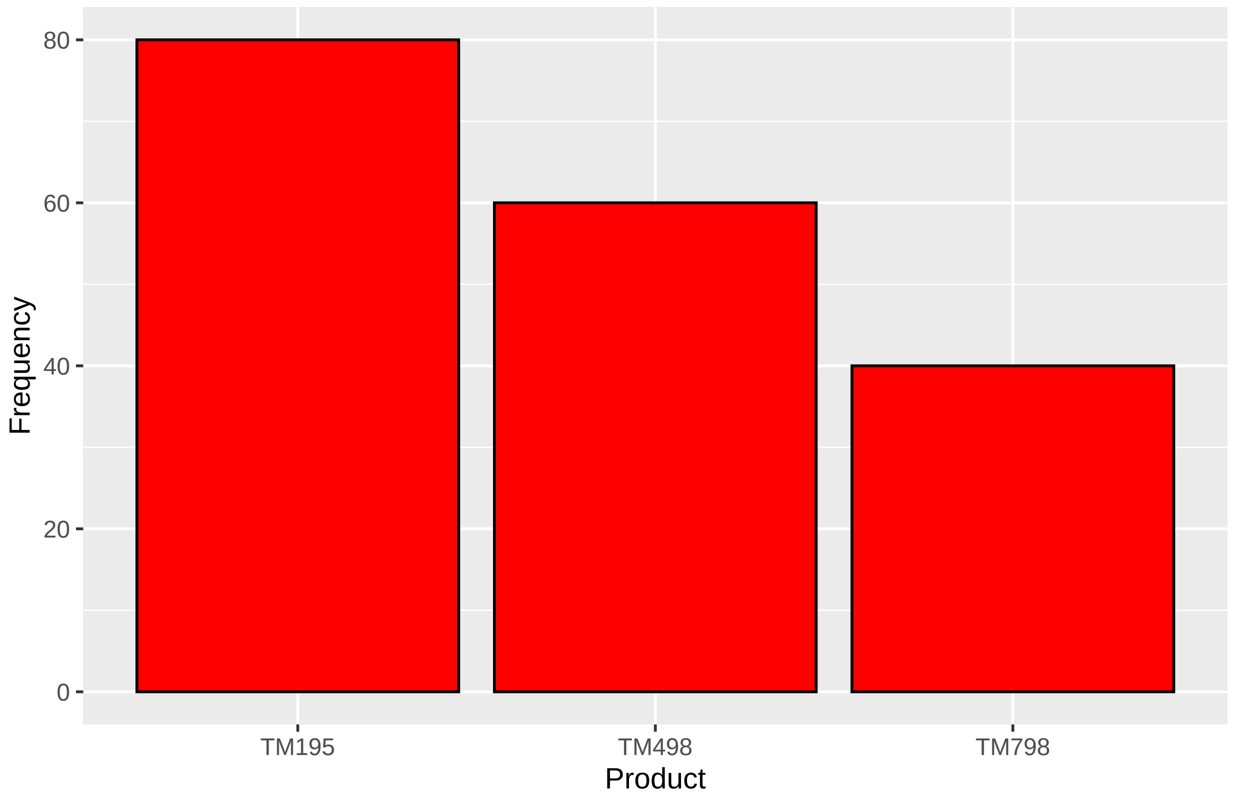
- The data shows the customer profiles of the treadmill product(s) of a retail store called Cardio Good Fitness.

3.3 Univariate Analysis

Categorical

1. Observartions on Product

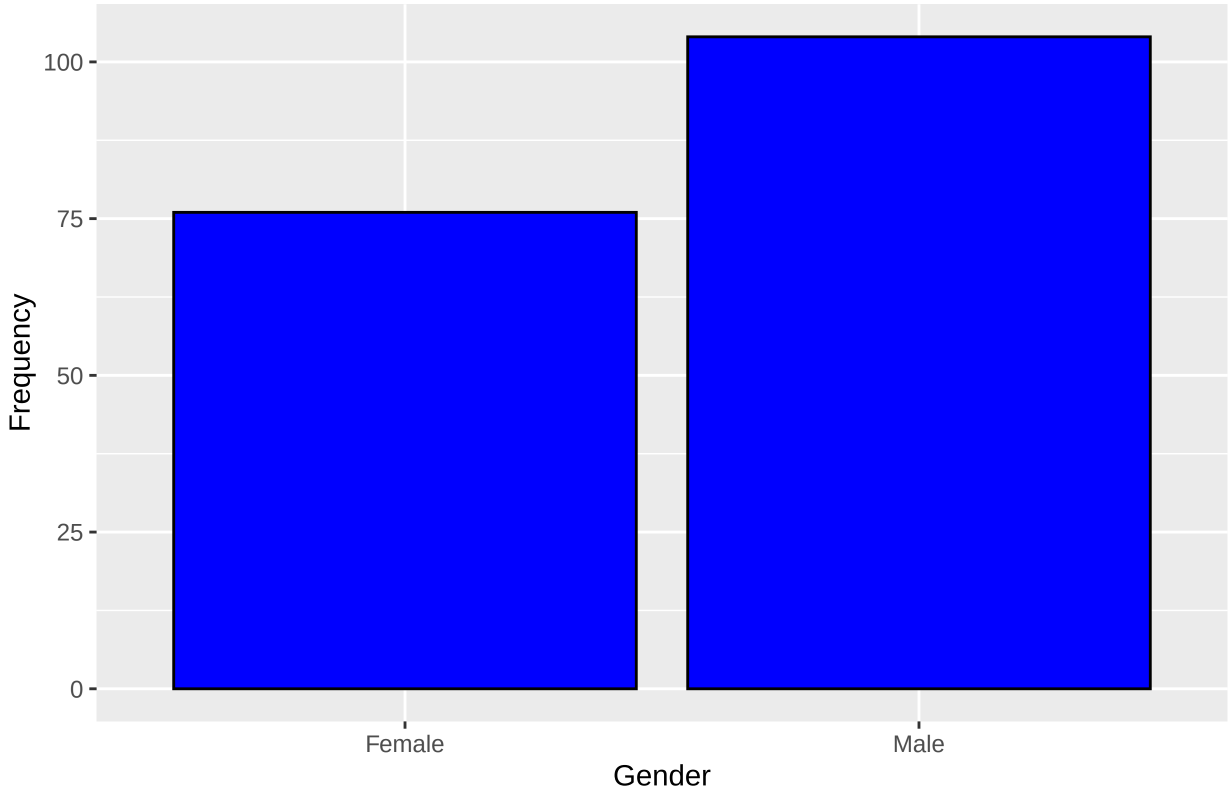
```
ggplot(Cardio_fitness, aes(x = Product)) +
  geom_bar(fill = c("red"), color="black") +
  labs(x = "Product",
       y = "Frequency",
       title = "")
```



- Customers seem to prefer the TM195 product.
- The least purchased product is the TM798.

2. Observations on Gender

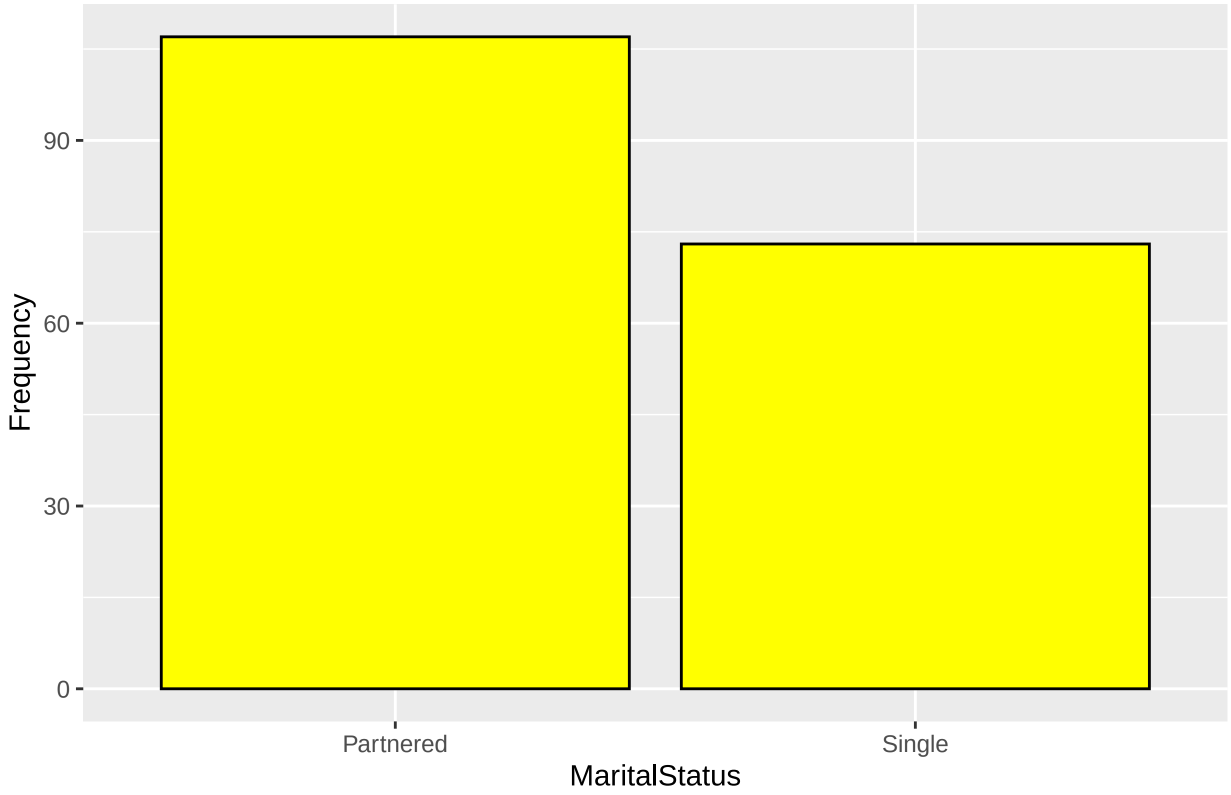
```
ggplot(Cardio_fitness, aes(x = Gender)) +
  geom_bar(fill = c("blue"), color="black") +
  labs(x = "Gender",
       y = "Frequency",
       title = "")
```



- The customer base is populated with more males than females.

3. Observations on MaritalStatus

```
ggplot(Cardio_fitness, aes(x = MaritalStatus)) +
  geom_bar(fill = c("yellow"), color="black") +
  labs(x = "MaritalStatus",
       y = "Frequency",
       title = "")
```



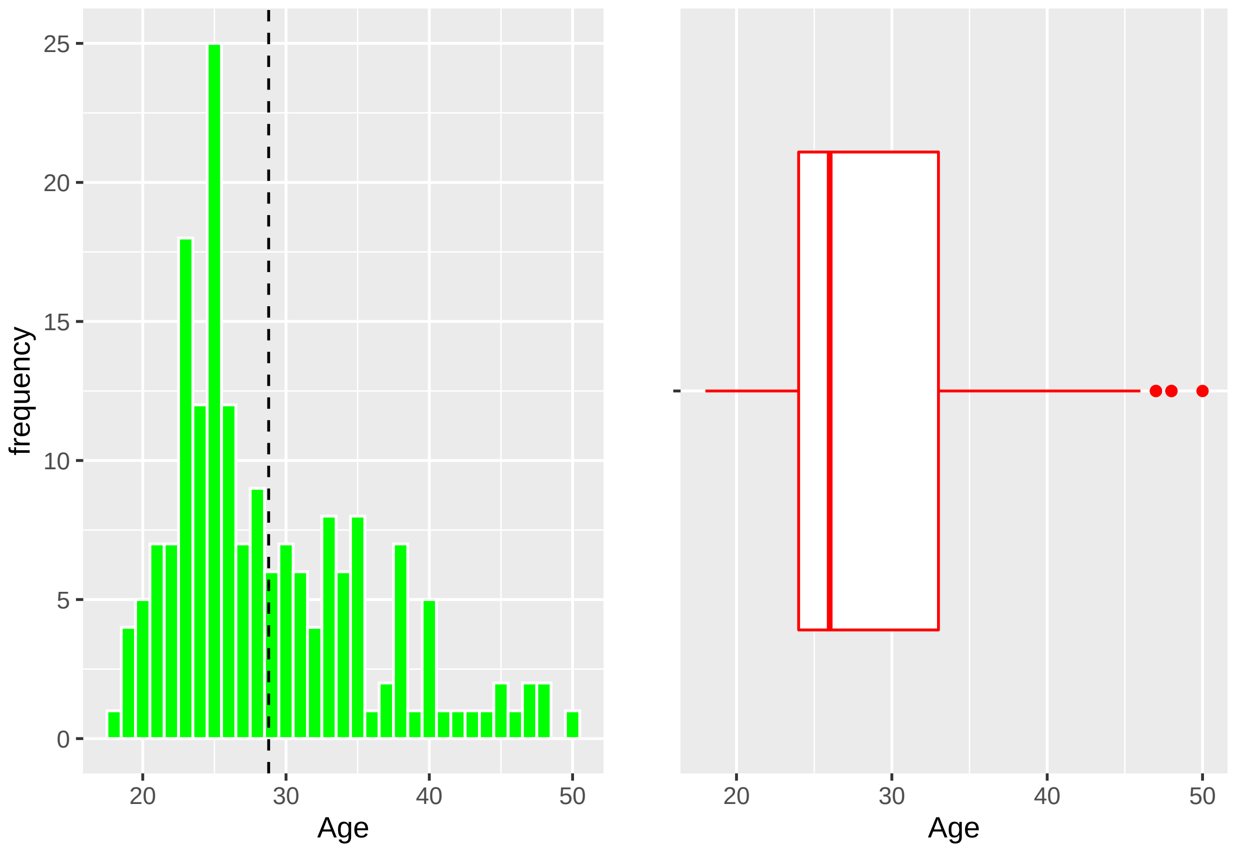
- The customer base is greatly populated by partnered than single.

```
plot_histogram_n_boxplot = function(variable, variableNameString, binw){

  a = ggplot(data = Cardio_fitness, aes(x= variable)) +
    labs(x = variableNameString,y ='frequency')+
    geom_histogram(fill = 'green',col = 'white', binwidth = binw) +
    geom_vline(aes(xintercept = mean(variable)),
               color = "black", linetype = "dashed", size = 0.5)

  b = ggplot(data = Cardio_fitness, aes('',variable))+
    geom_boxplot(outlier.colour = 'red',col = 'red', outlier.shape = 19)+
    labs(x = '', y = variableNameString) + coord_flip()
  grid.arrange(a,b,ncol = 2)
}
```

Quantitative

4. Observations on Age

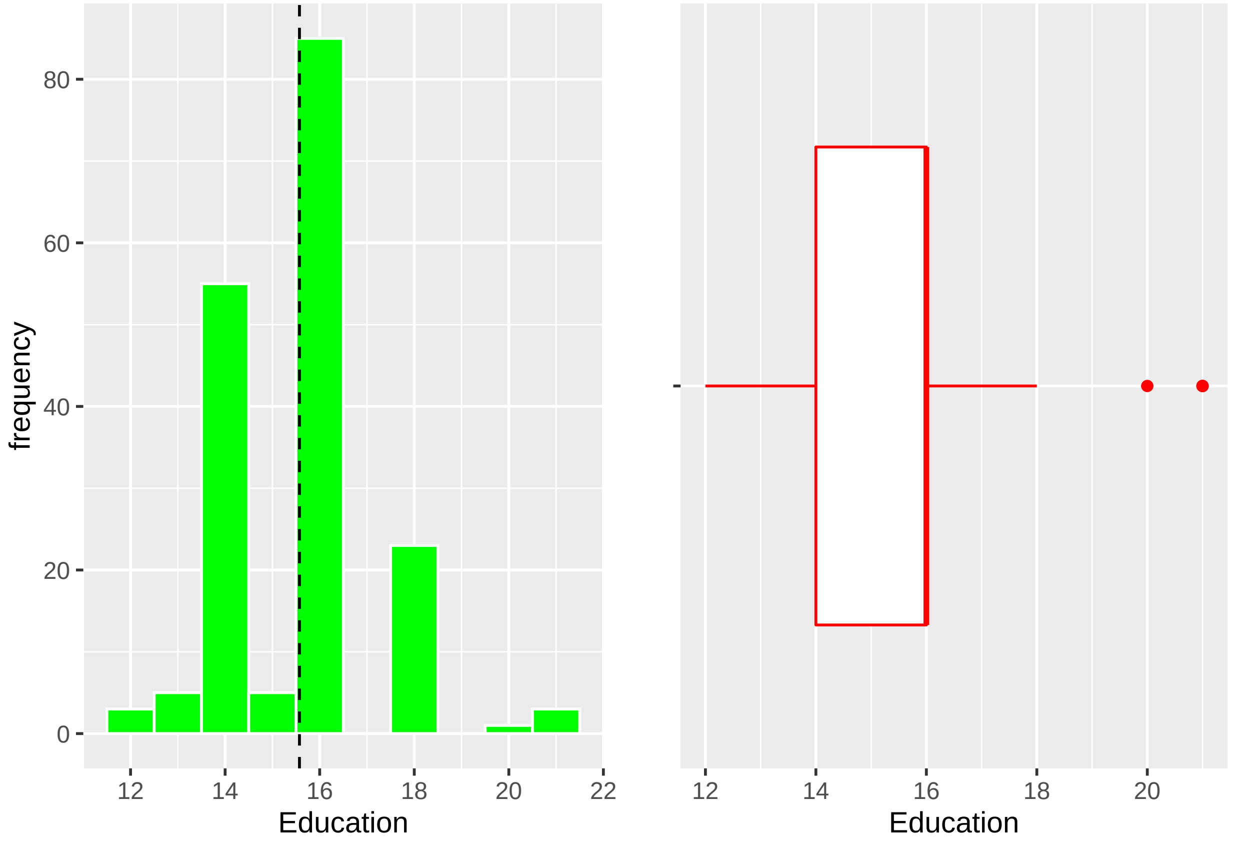
```
plot_histogram_n_boxplot(Cardio_fitness$Age, 'Age', 1)
```



- There is a right skew in the distribution
- The variable contains outliers
- Average age is 28.8
- As the customer base ages, the tendency to purchase a product reduces.

5. Observations on Education

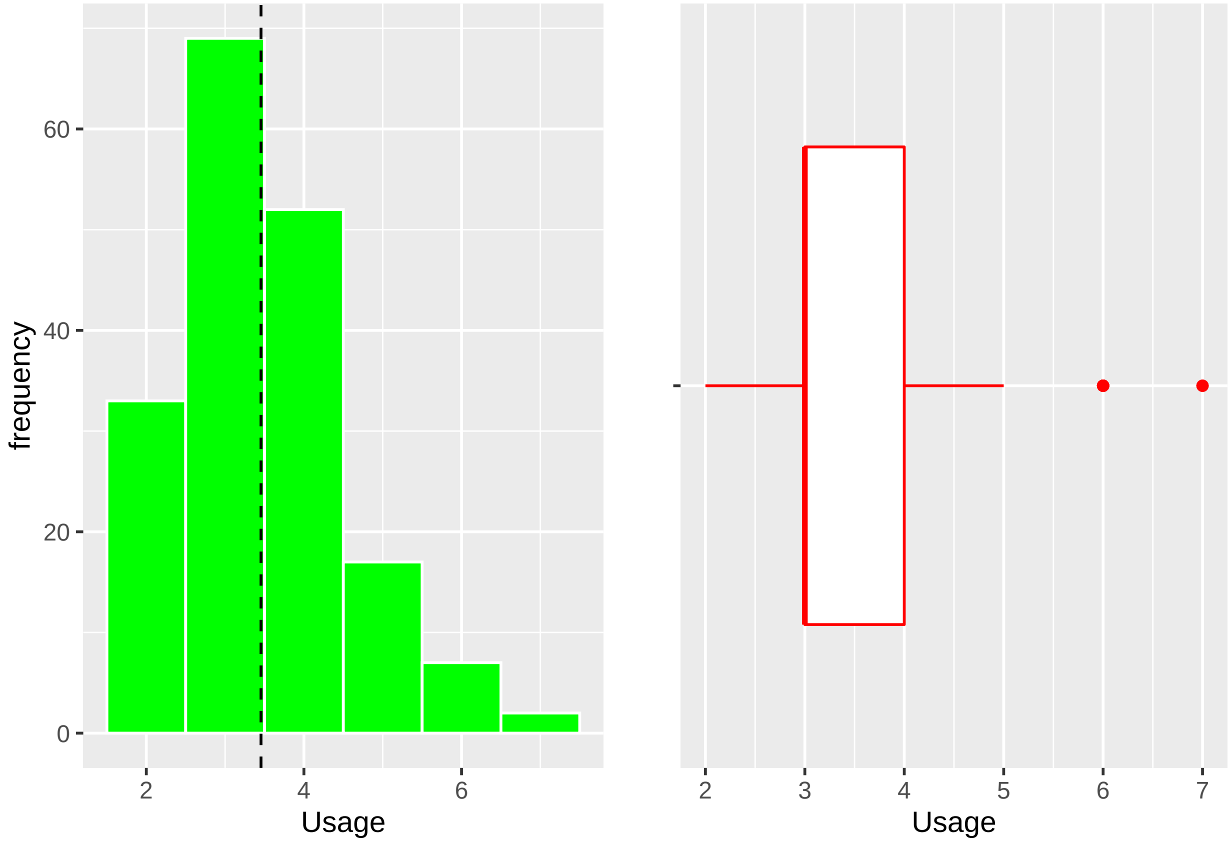
```
plot_histogram_n_boxplot(Cardio_fitness$Education, 'Education', 1)
```



- Education is left skewed
- There are however outliers towards the right, indicating that there are few customers with a higher level of education interested in buying a product.

6. Observations on Usage

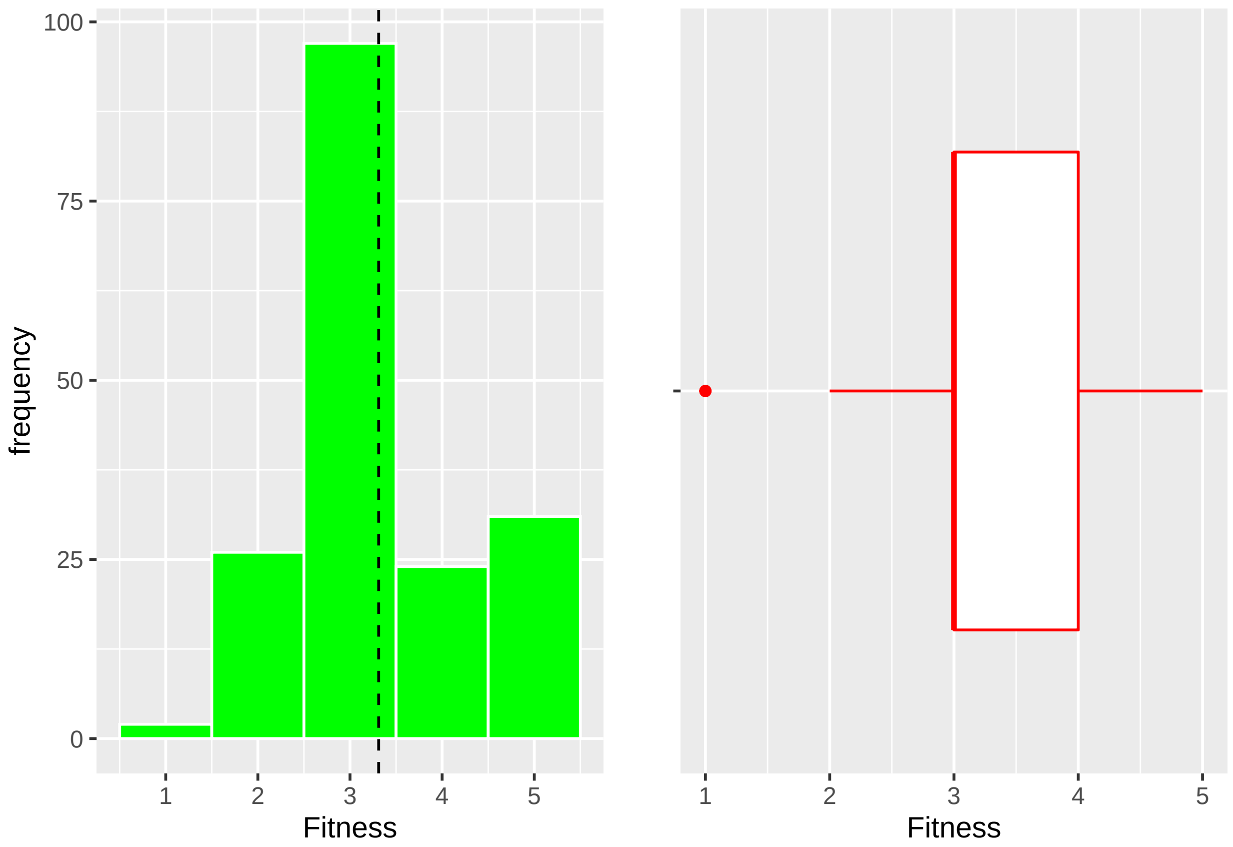
```
plot_histogram_n_boxplot(Cardio_fitness$Usage, 'Usage', 1)
```



- The usage variable suggests a right skew in distribution.
- There are a few outliers towards the right, with the maximum at 7.

7. Observations on Fitness

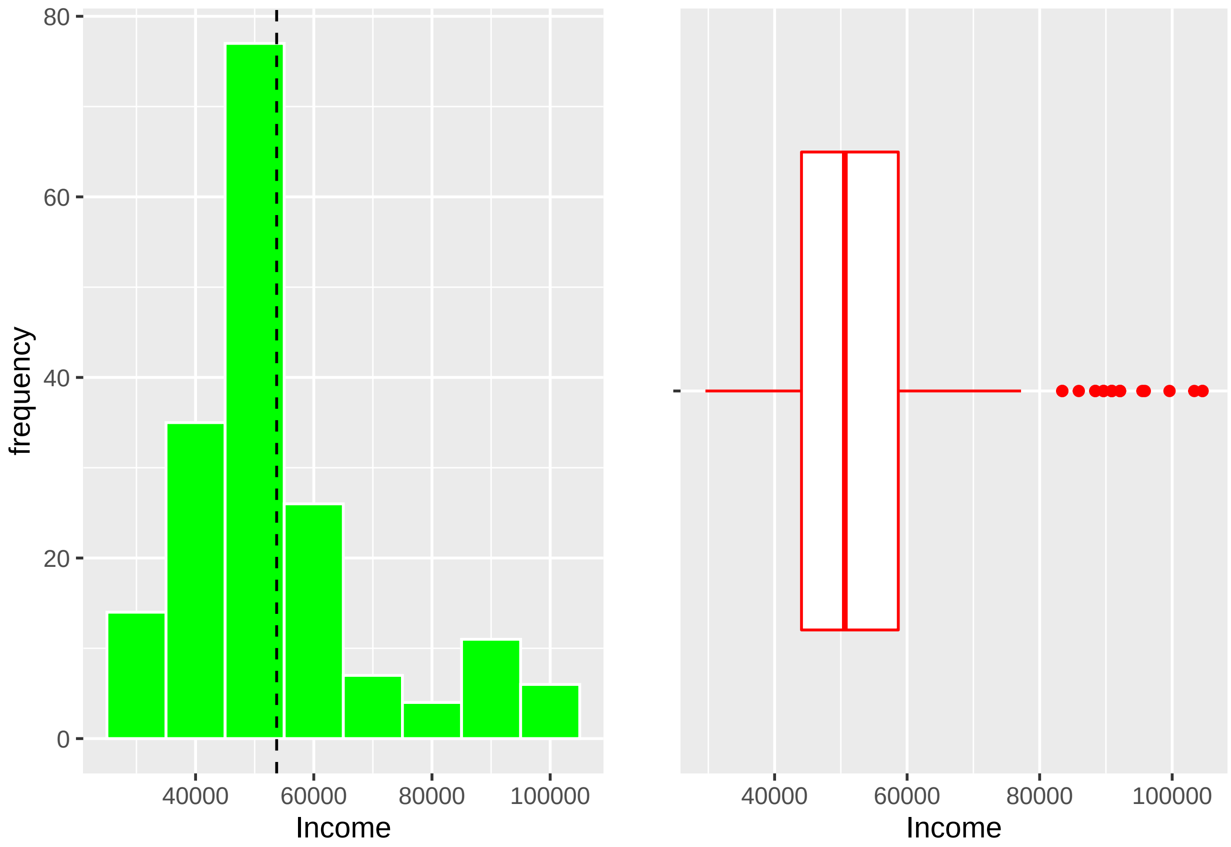
```r
plot_histogram_n_boxplot(Cardio_fitness$Fitness, 'Fitness', 1)
```



- The fitness variable is skewed to the right.
- There is however an outlier towards the left.

8. Obervations on Income

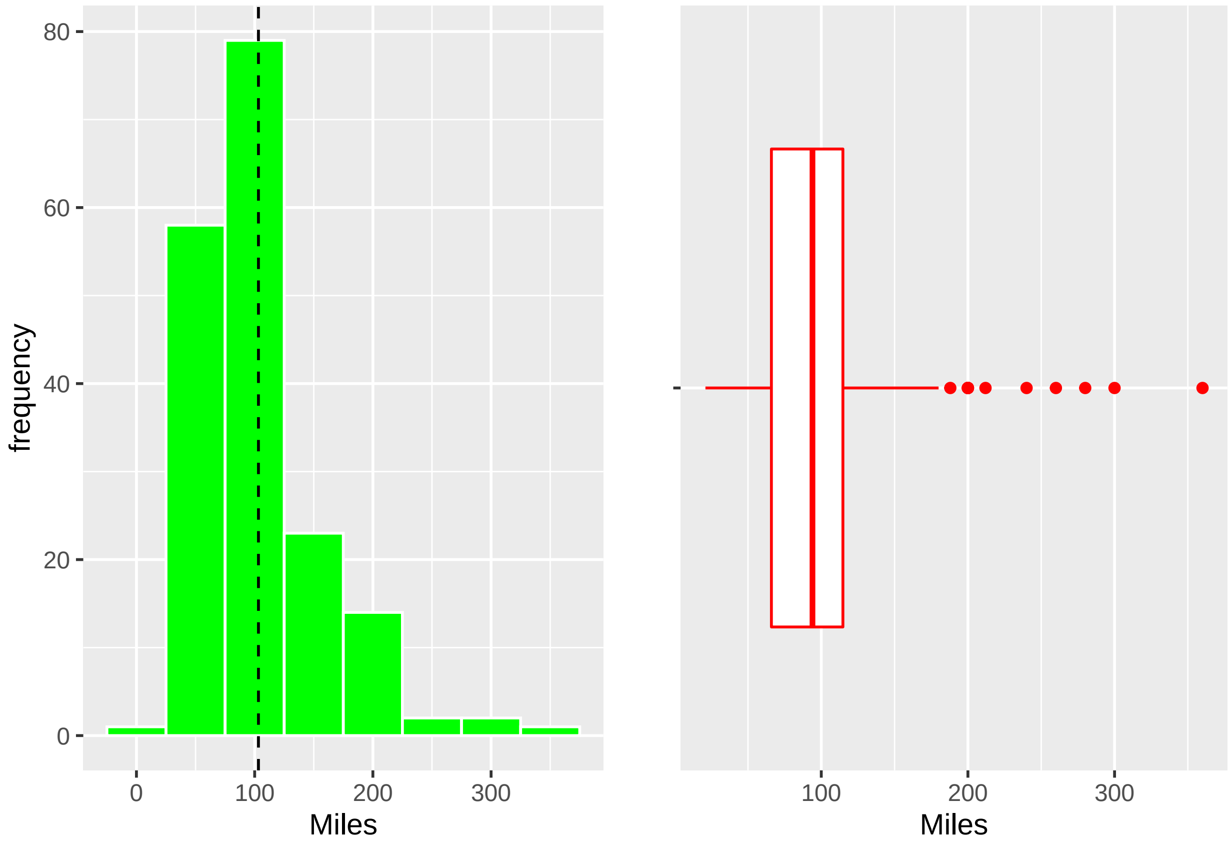
```
plot_histogram_n_boxplot(Cardio_fitness$Income, 'Income', 10000)
```



- The income variable is skewed to the right.
- Both mean and median suggest most of the customer base are average income earners.
- There are however outliers towards the right, indicating a wealthy customer base with the highest at 104,581.

9. Observations on Miles

```r
plot_histogram_n_boxplot(Cardio_fitness$Miles, 'Miles', 50)
```



- The miles variable is skewed to the right.
- There are numerous outliers towards the right. This could suggest the customer base is healthy and atheletic.

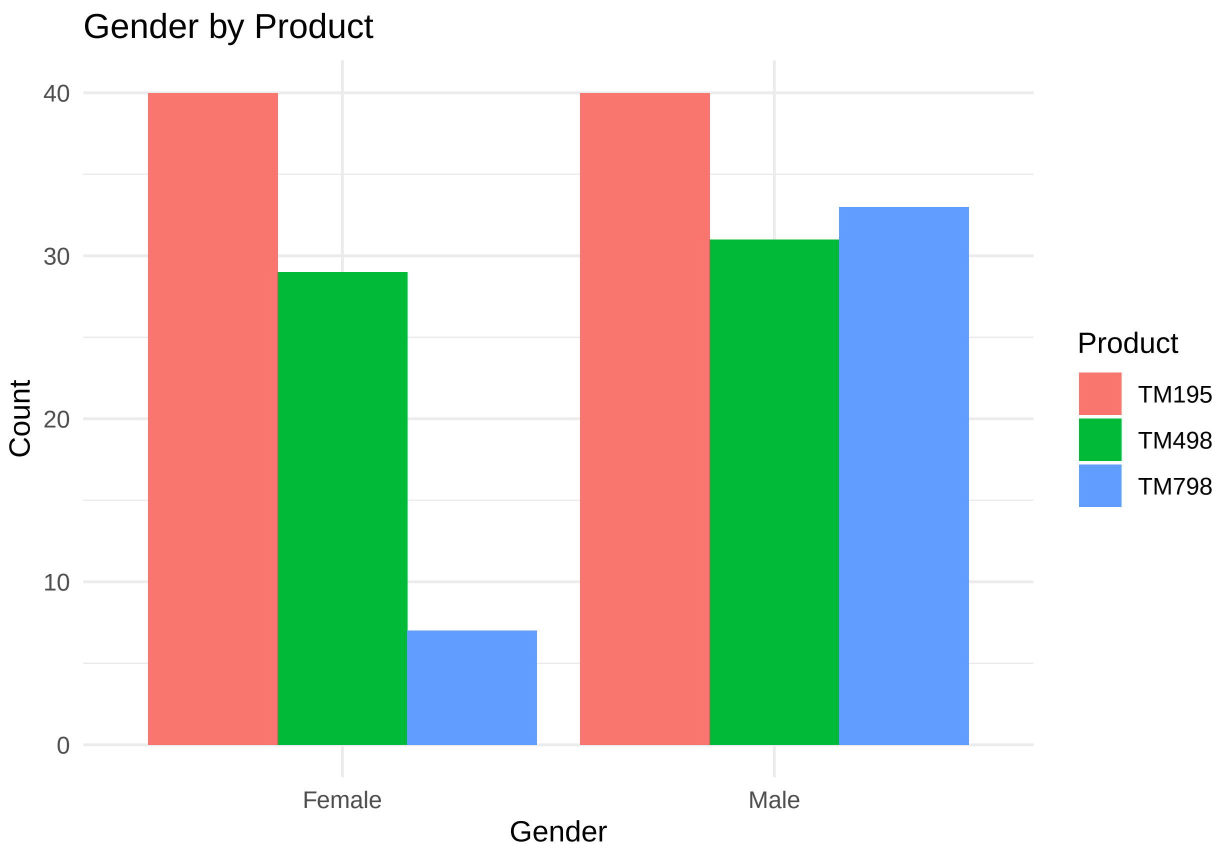
3.4 Bivariate Analysis

Plot bivariate charts between variables to understand their relationship with each other.

1.

Relationship between Product and Gender

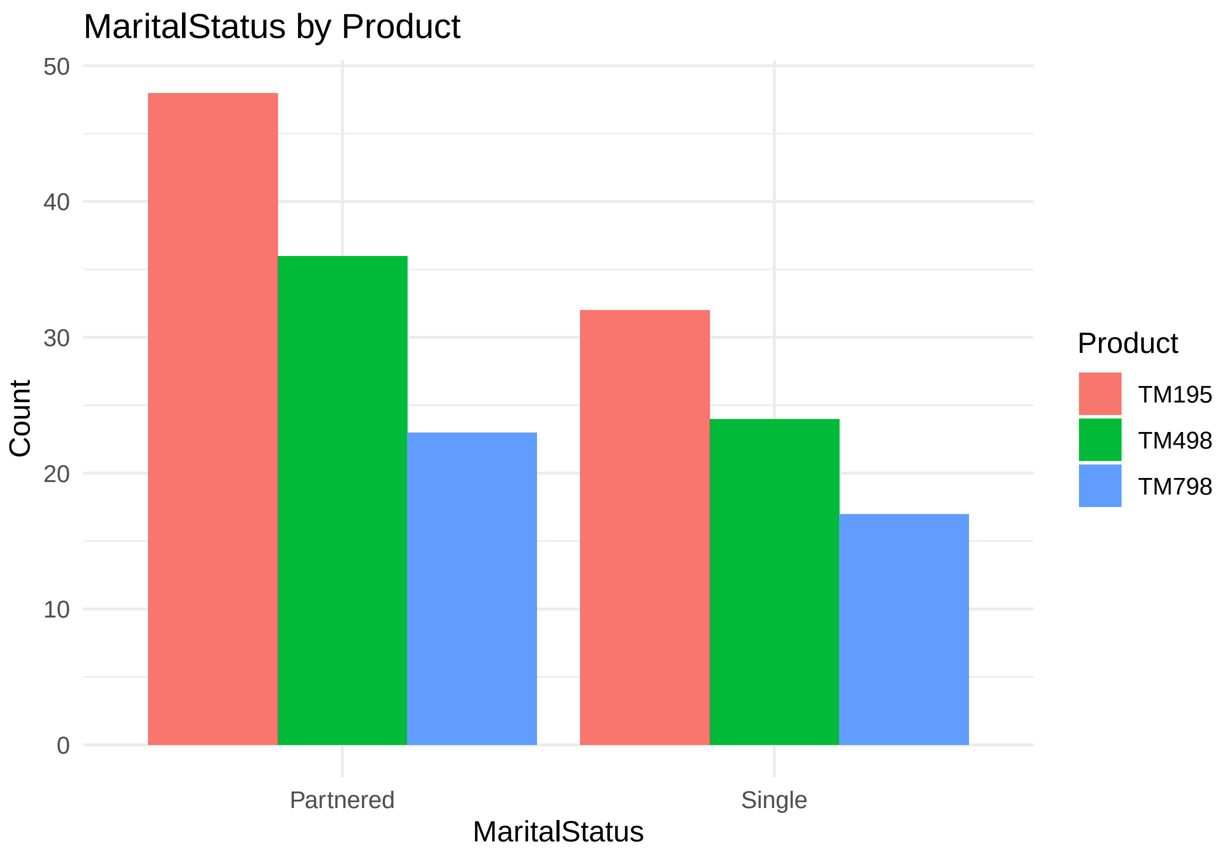
```r
ggplot(Cardio_fitness, aes(x = Gender, fill = Product)) +
  geom_bar(position = "dodge") +
  labs(y = "Count",
       fill = "Product",
       x = "Gender",
       title = "Gender by Product") +
  theme_minimal()
```



- Both male and female seem to purchase more of TM195.
- Females tend to require less of TM798.
- Males purchase more product in total.

Relationship between Product and MaritalStatus

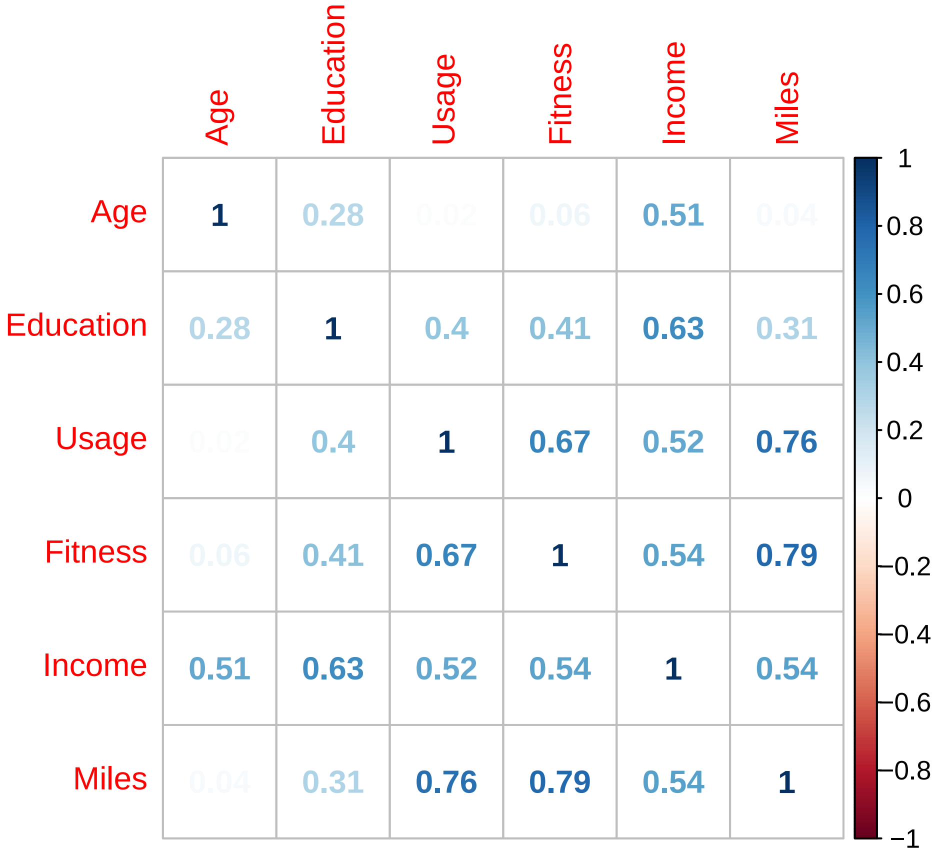
```r
ggplot(Cardio_fitness, aes(x = MaritalStatus, fill = Product)) +
  geom_bar(position = "dodge") +
  labs(y = "Count",
       fill = "Product",
       x = "MaritalStatus",
       title = "MaritalStatus by Product") +
  theme_minimal()
```



- Both single and partnered prefer the TM195 over other product.
- In total, partnered tend to purchase more product than single.

2. Check for correlation among numerical variables

```
# Numeric variables in the data
num_vars = sapply(Cardio_fitness, is.numeric)

# Correlation Plot
corrplot(cor(Cardio_fitness[,num_vars]), method = 'number')
```



- As expected, usage shows high correlation with miles.
- Likewise, fitness shows high correlation with miles as well.
- There is a moderate correlation between age and income.
- There is a moderate correlation between education and income which could sugegst education impacts the level of income.
- Education, usage, fitness and miles have weak or no correlation with age.
- It is important to note that correlation does not imply causation.

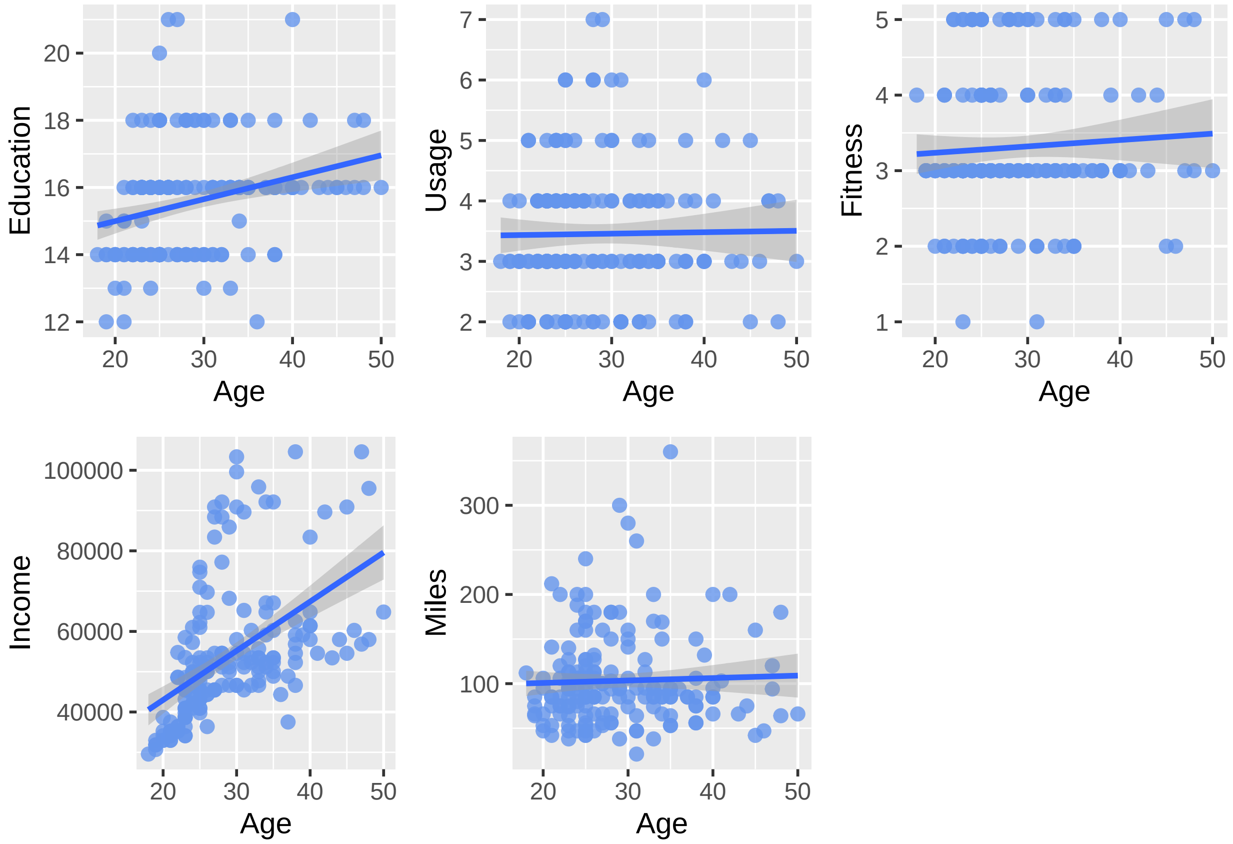
3. Bivariate scatter plots

```
plot_scatterplot = function(variableNameString, variable, binw){
  ggplot(data = Cardio_fitness, aes(x= Age, y = variable)) +
    labs(x = "Age", y = variableNameString) +
    geom_point(color="cornflowerblue", size =2, alpha=.8)+
    geom_smooth(method ="lm") # adds a linear trend line which is useful to summarize the relationship
}
```

Relationship between Age and other numeric variables

```
grid.arrange(plot_scatterplot('Education', Cardio_fitness$Education),
             plot_scatterplot('Usage', Cardio_fitness$Usage),
             plot_scatterplot('Fitness', Cardio_fitness$Fitness),
             plot_scatterplot('Income', Cardio_fitness$Income),
             plot_scatterplot('Miles', Cardio_fitness$Miles),
             ncol = 3)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



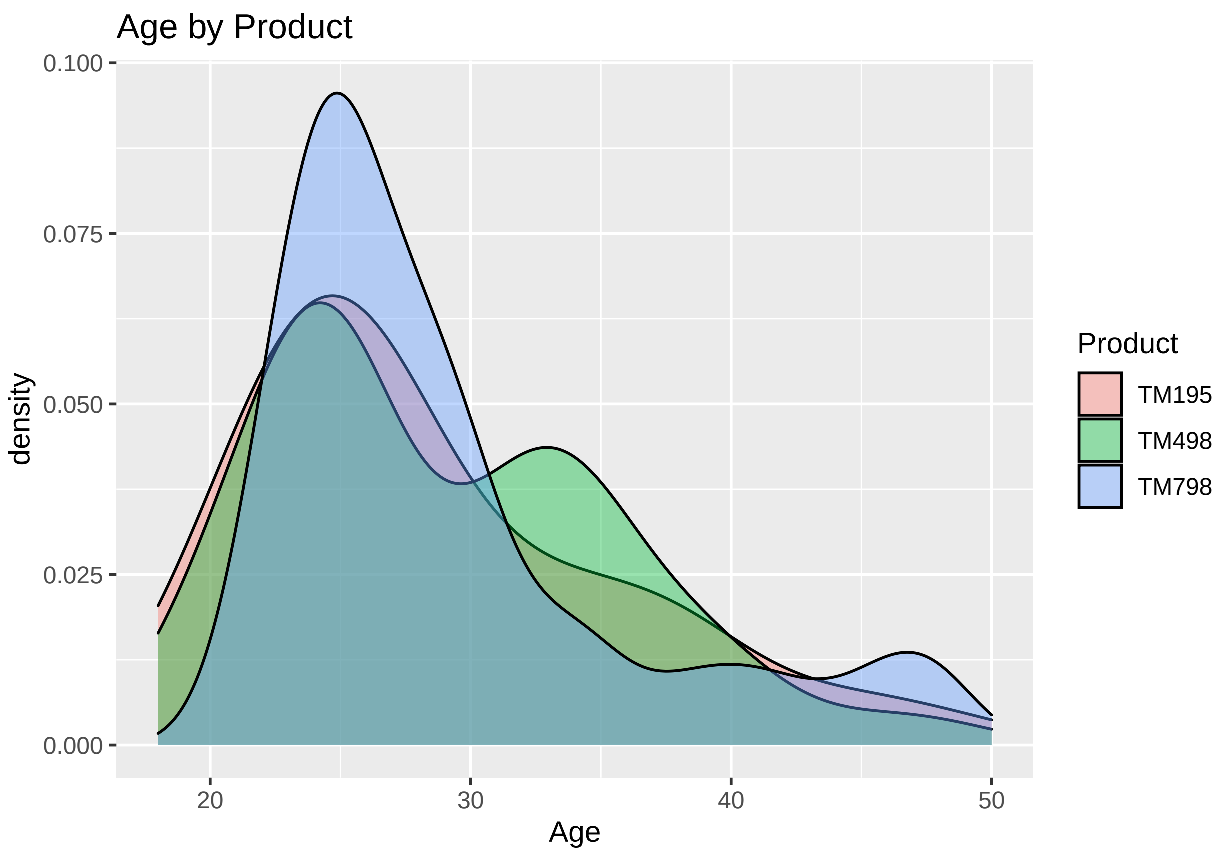
- There is a moderate correlation between age and income.
- There exist a weak or no correlation between age and education, usage, fitness and miles.

4. Bivariate grouped kernel density plots and box plots

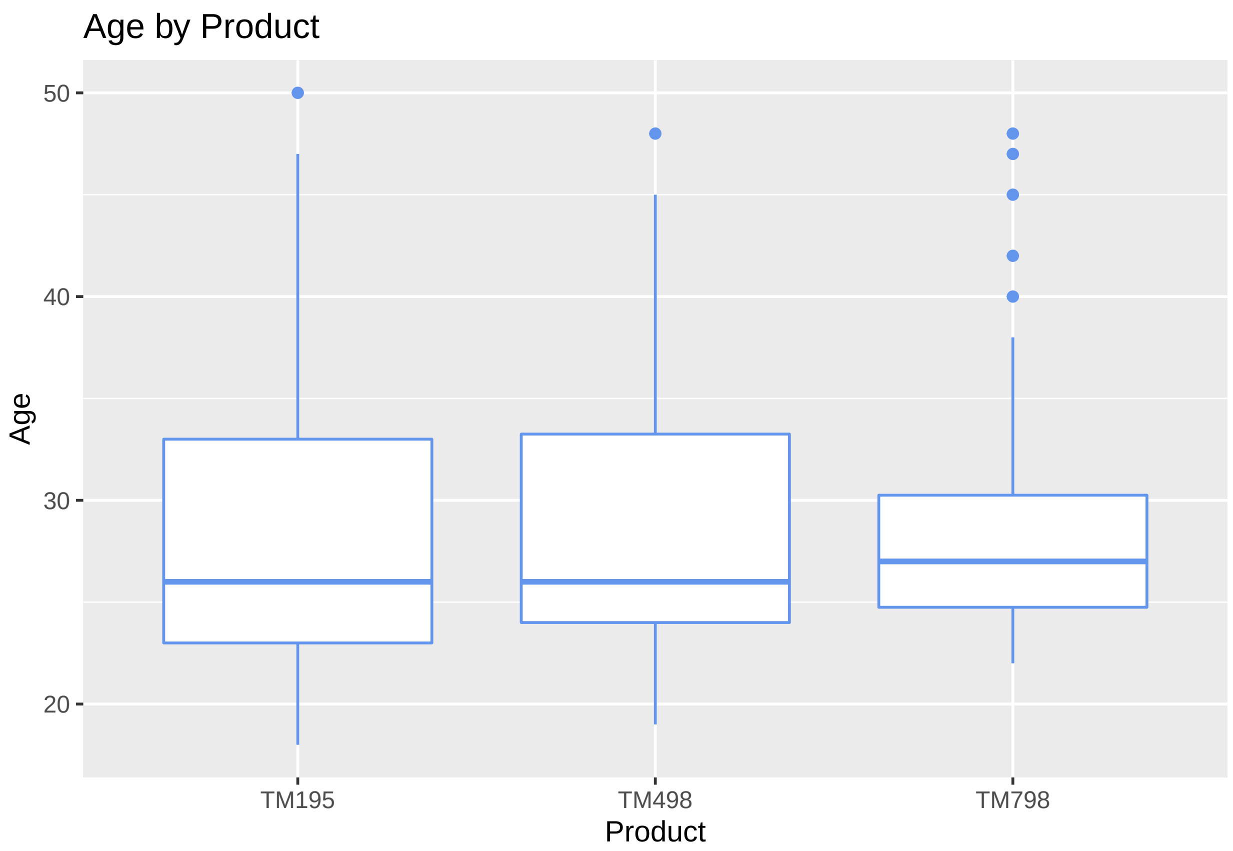
Relationship between Product and numeric variables

1. Age

```
ggplot(Cardio_fitness,
       aes(x = Age,
           fill = Product)) +
  geom_density(alpha = 0.4) +
  labs(title = "Age by Product")
```



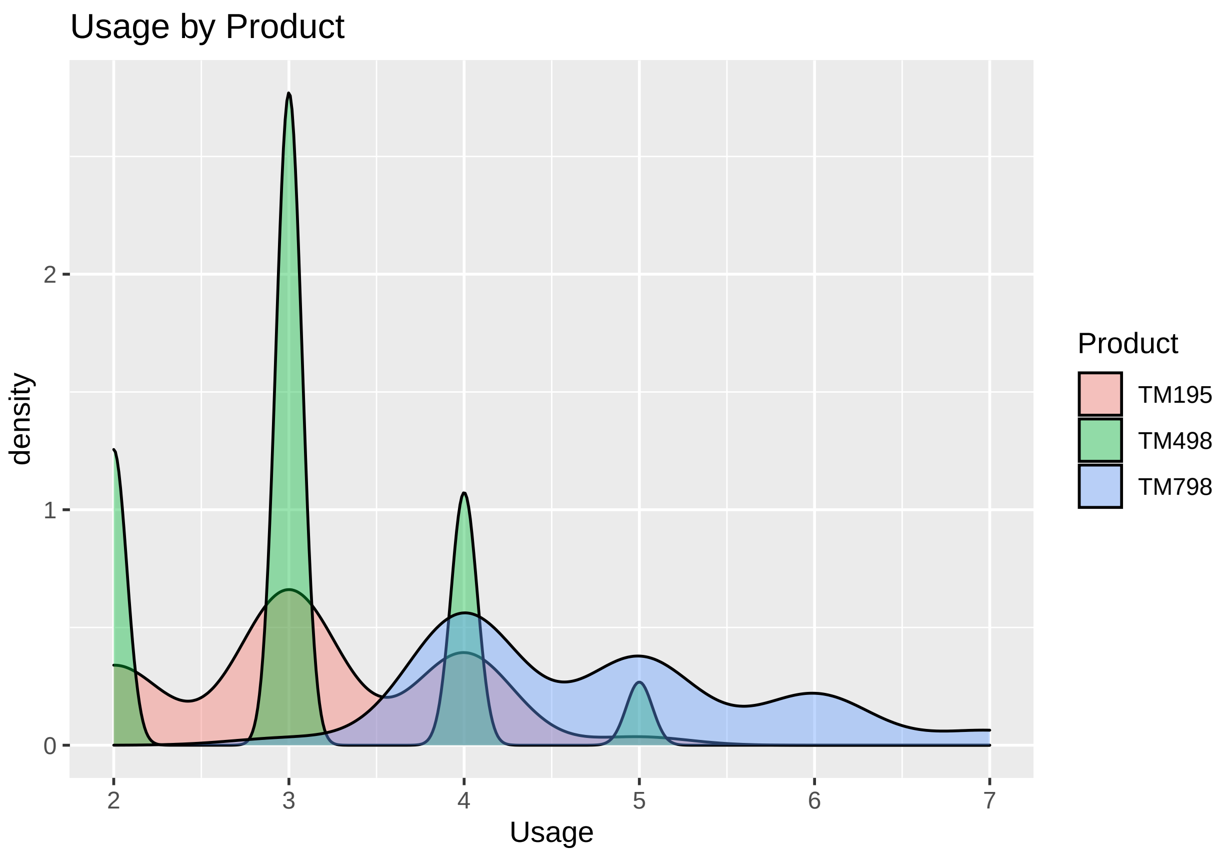
```r
ggplot(Cardio_fitness,
       aes(x = Product,
           y = Age)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Age by Product")
```



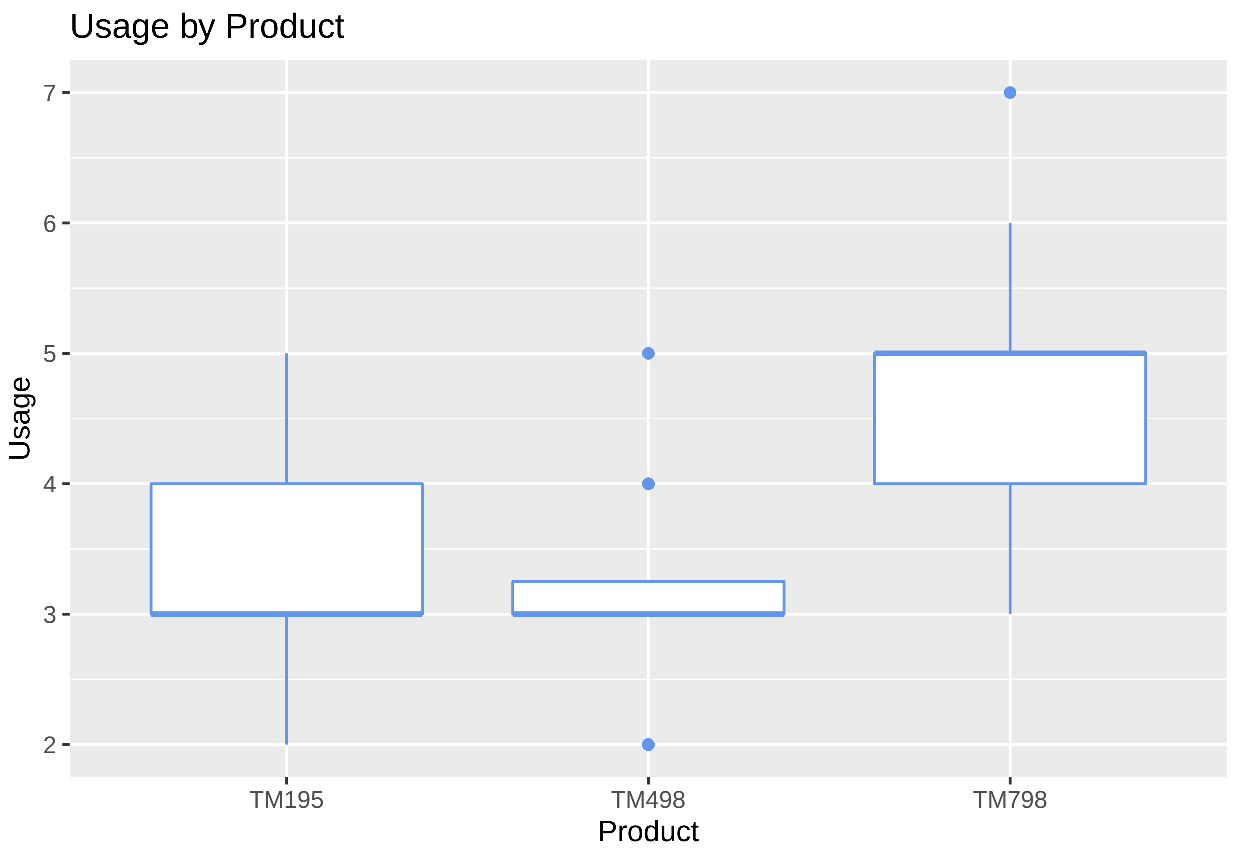
- All age group purchase a treadmill product.
- The age group can be classified into < 20, 20 to 30 and > 30.
- Customers whose age fall within 20 and 30 are likely to purchase treadmill product(s).
- The TM798 product is more common among customers with ages between 20 < x < 30.
- Customers whose age fall within 20 and 30 seem to be more interested in exercise.

2. Usage

```
ggplot(Cardio_fitness,
       aes(x = Usage,
           fill = Product)) +
  geom_density(alpha = 0.4) +
  labs(title = "Usage by Product")
```



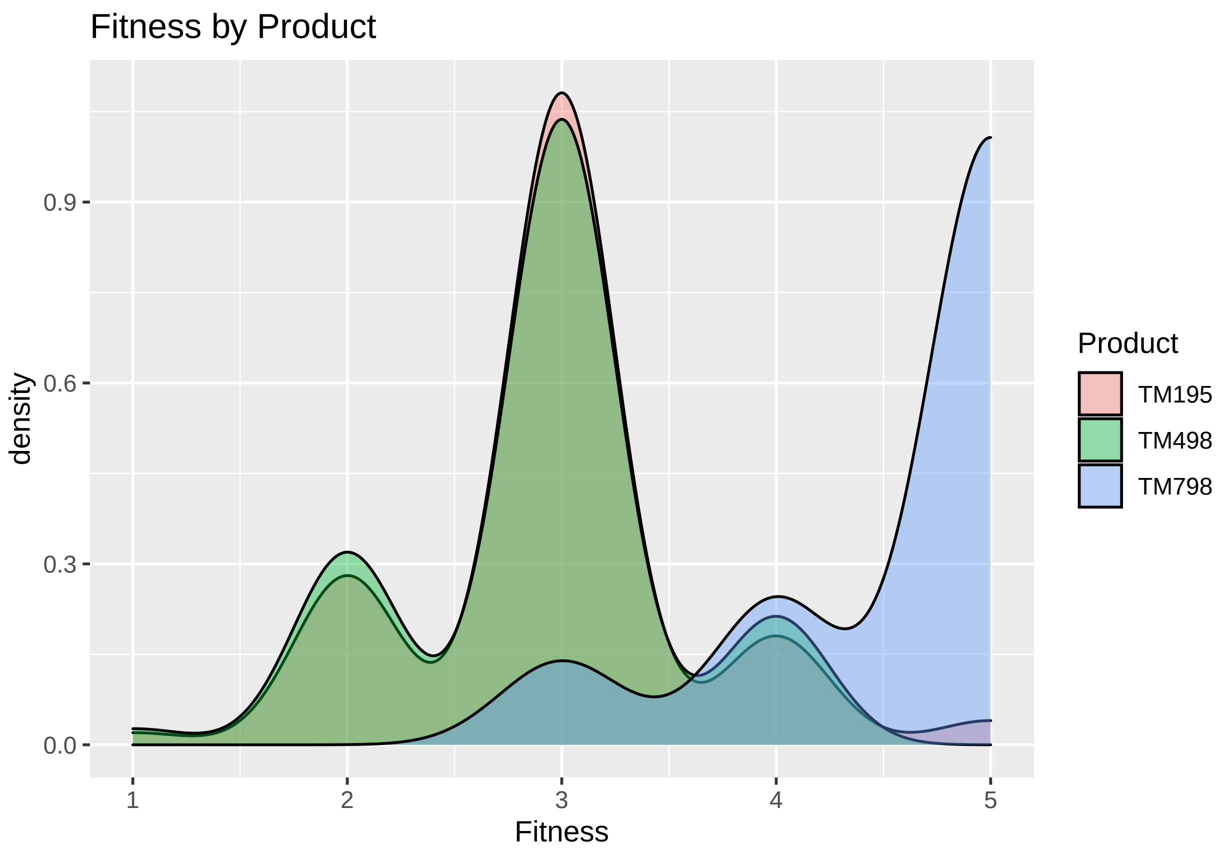
```
ggplot(Cardio_fitness,
       aes(x = Product,
           y = Usage)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Usage by Product")
```



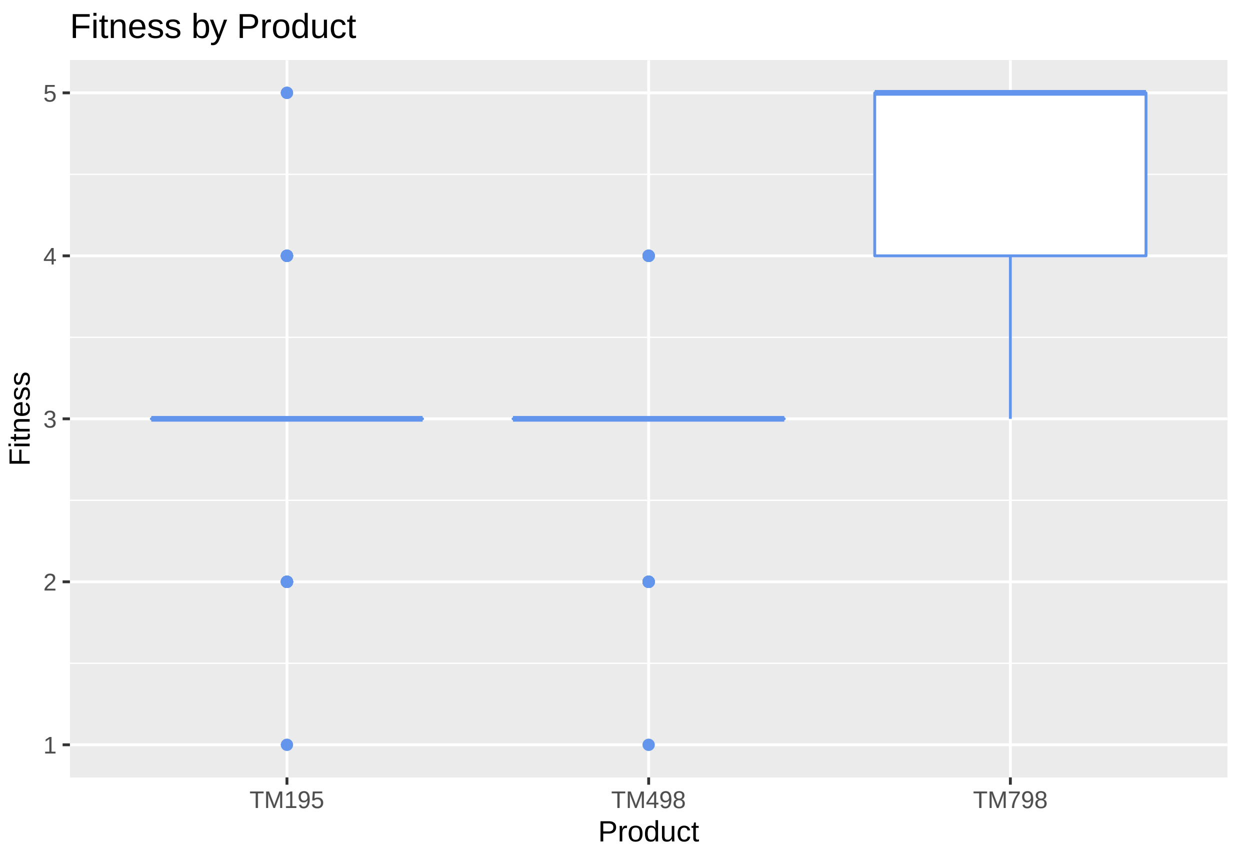
- Customers use the TM498 product between 2 to 5 times in a discrete manner.
- Customers use the TM195 product between 2 to 5 times in a continuous manner.
- Customers usage of TM798 is > 3.

3. Fitness

```
ggplot(Cardio_fitness,
       aes(x = Fitness,
           fill = Product)) +
  geom_density(alpha = 0.4) +
  labs(title = "Fitness by Product")
```



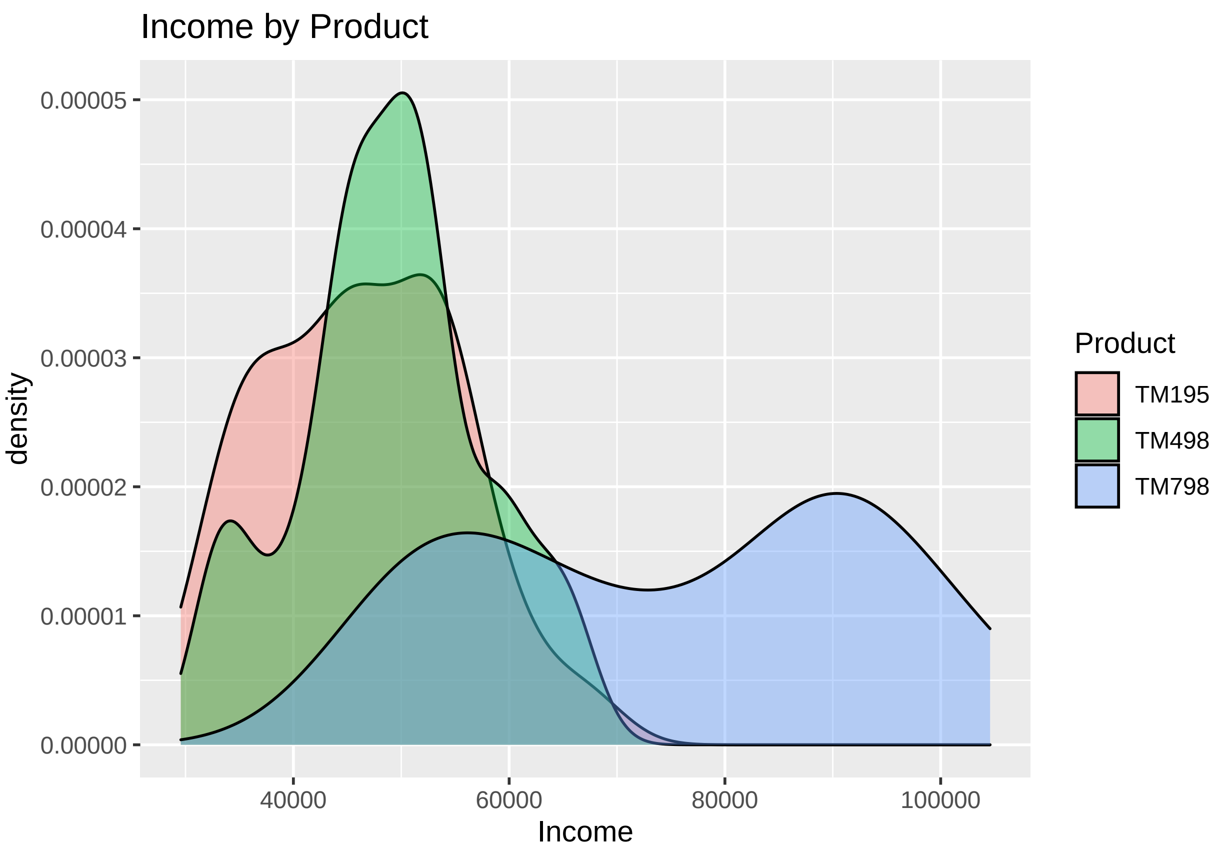
```
ggplot(Cardio_fitness,
       aes(x = Product,
           y = Fitness)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Fitness by Product")
```



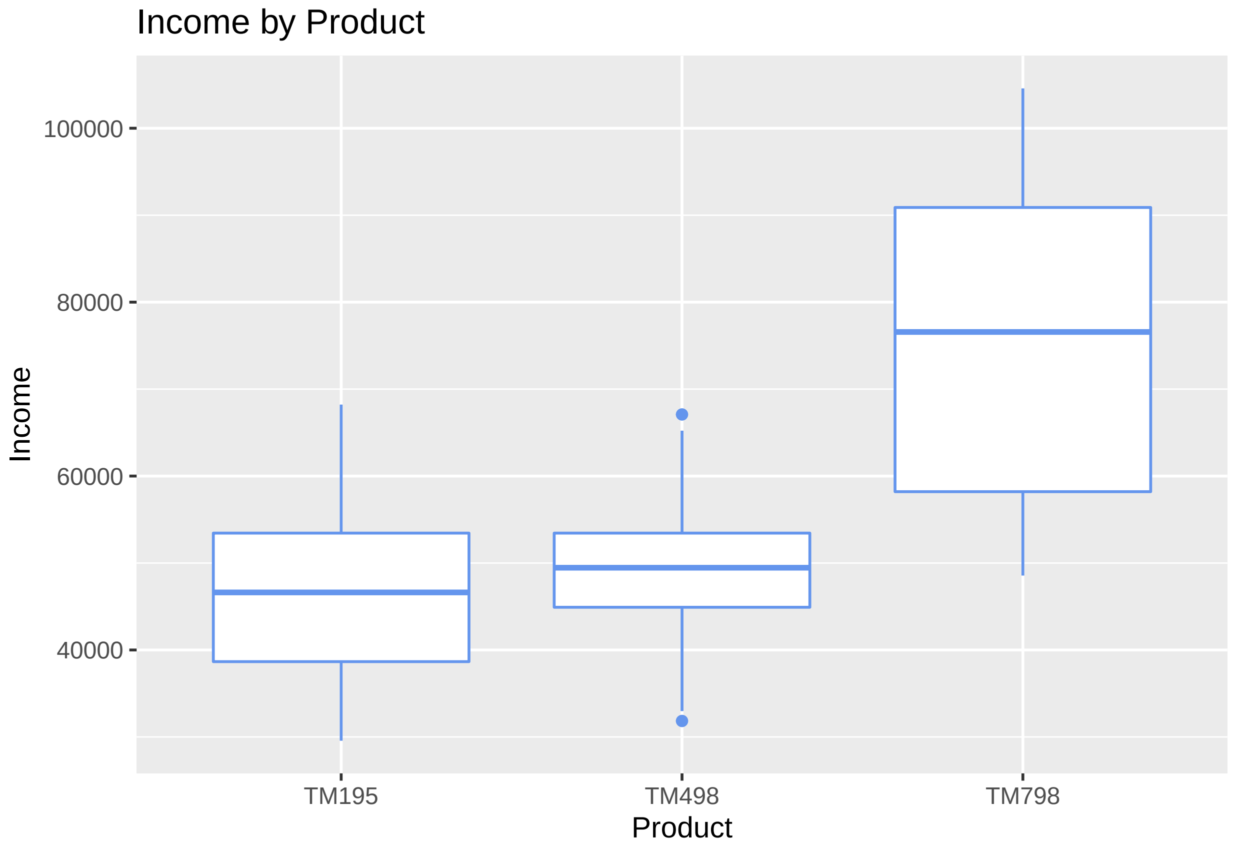
- Customers achieve fitness across all product.
- Customers who purchase TM195 and TM498 maintain an average fitness shape at 3.
- However, fitness shape tend to be better with the TM798 product.

4. Income

```
ggplot(Cardio_fitness,
       aes(x = Income,
           fill = Product)) +
  geom_density(alpha = 0.4) +
  labs(title = "Income by Product")
```



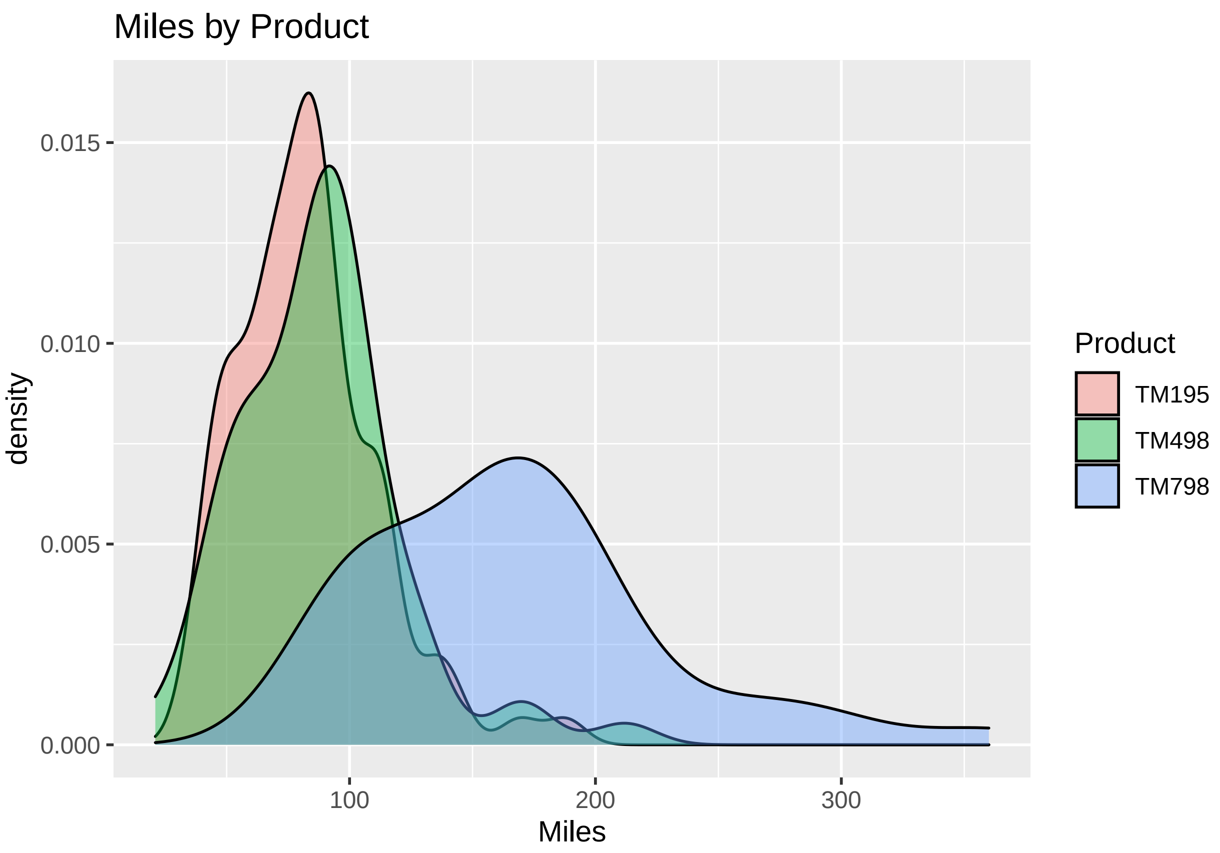
```
ggplot(Cardio_fitness,
       aes(x = Product,
           y = Income)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Income by Product")
```



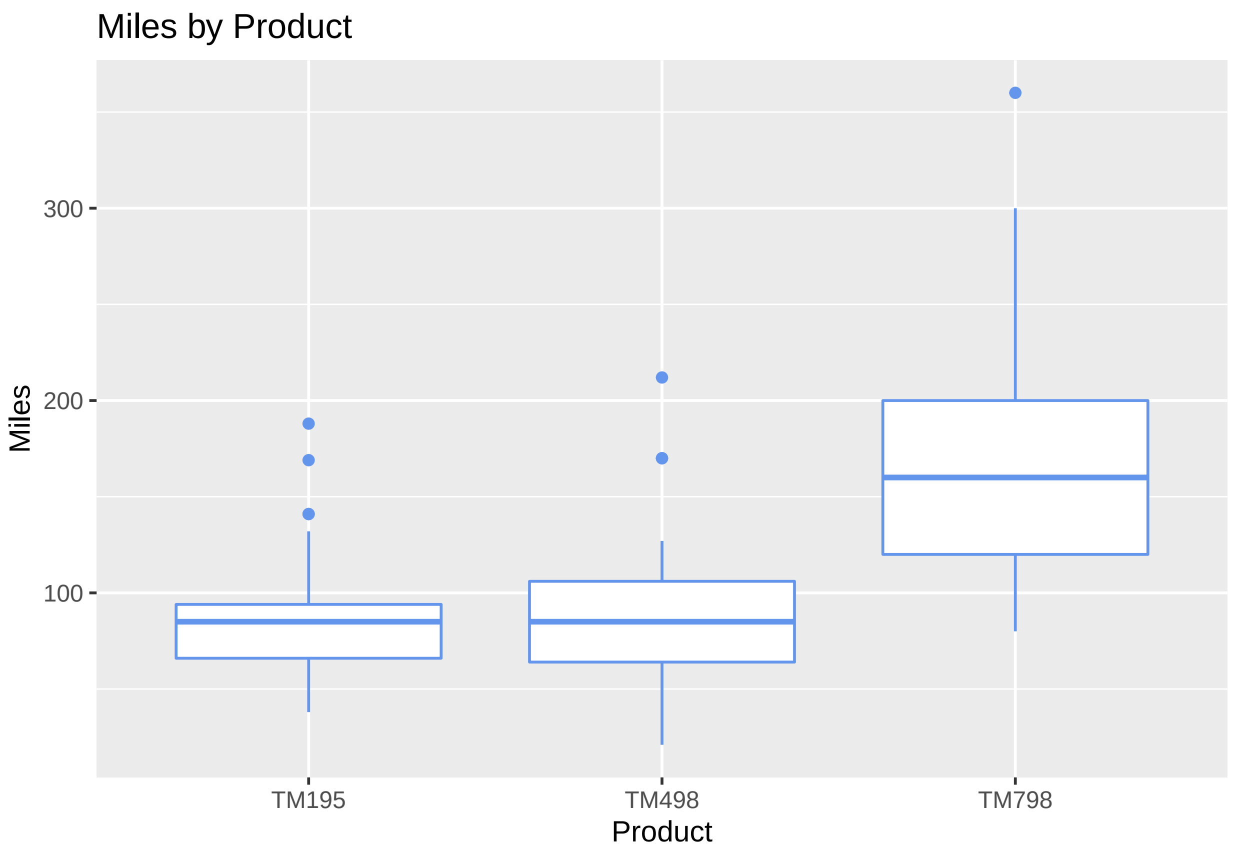
- Customers with income 29,562 < x < 70,000 purchase the TM195 and TM498 products.
- Customers with income 40,000 < x < 60,000 purchase more of TM498 and TM195.
- Customers of all income level purchase the TM798 although a higher proportion of that lies with income > 50,000.

5. Miles

```
ggplot(Cardio_fitness,
       aes(x = Miles,
           fill = Product)) +
  geom_density(alpha = 0.4) +
  labs(title = "Miles by Product")
```



```
ggplot(Cardio_fitness,
       aes(x = Product,
           y = Miles)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Miles by Product")
```

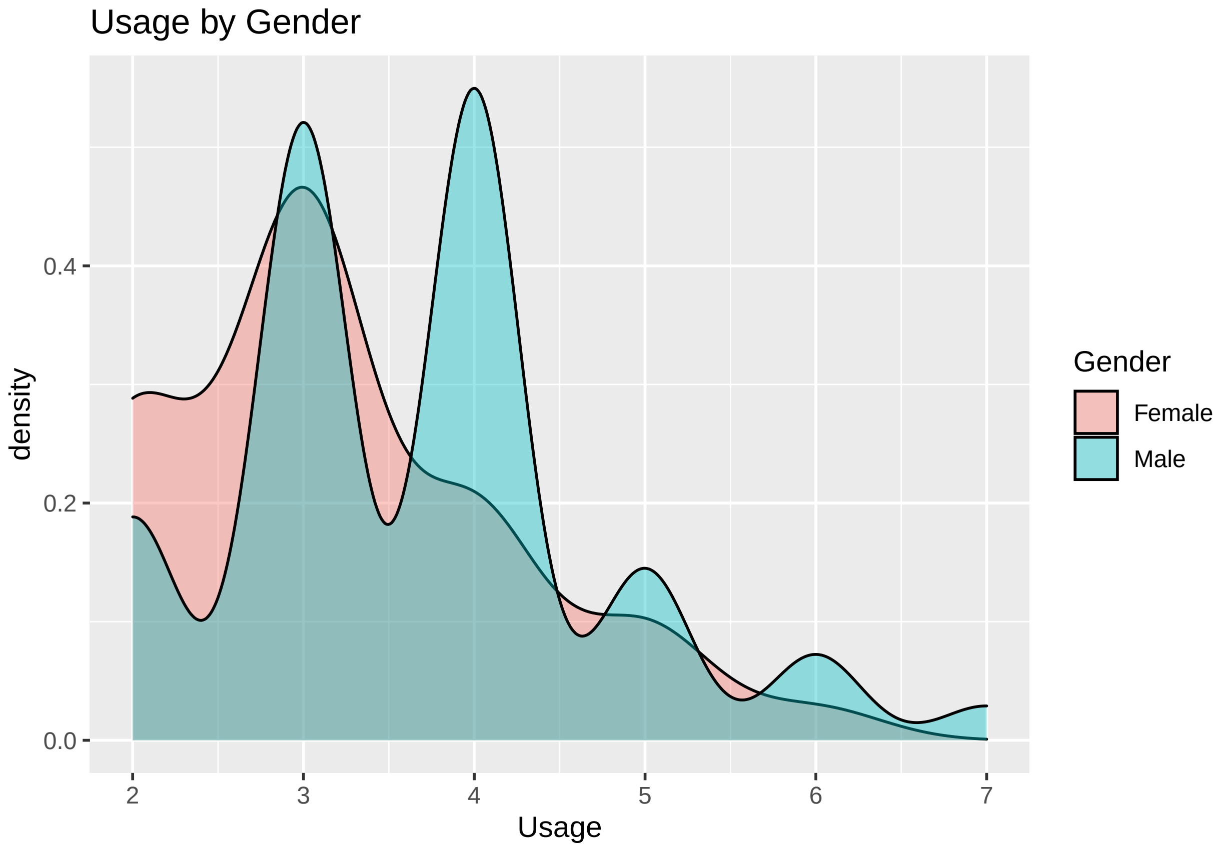


- Customers who use TM195 and TM498 tend to run a maximum of 140 miles with the exception of outliers.
- Customers who use TM798 tend to run more and have no limit to the number of miles covered, indicating the althetic group.

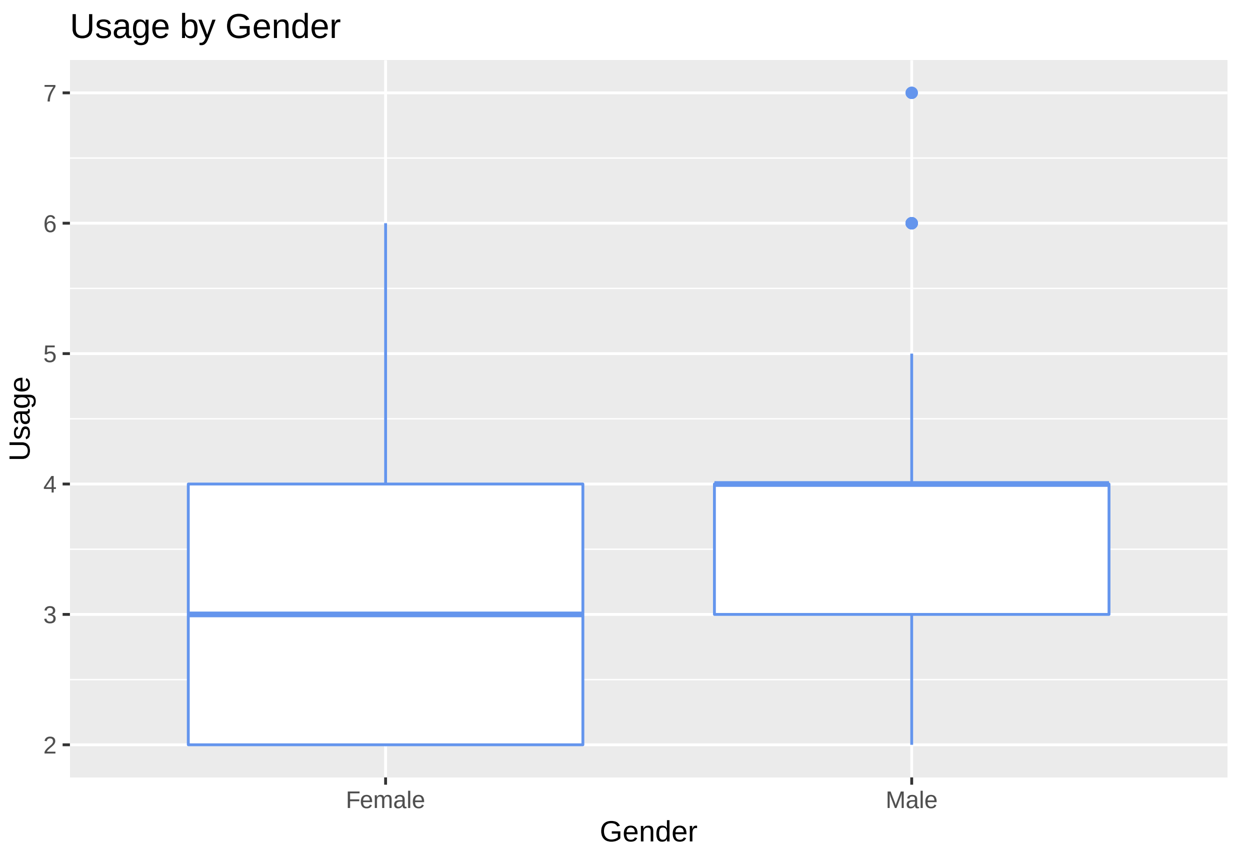
Relationship between Gender and numeric variables

1. Usage

```
ggplot(Cardio_fitness,
       aes(x = Usage,
           fill = Gender)) +
  geom_density(alpha = 0.4) +
  labs(title = "Usage by Gender")
```



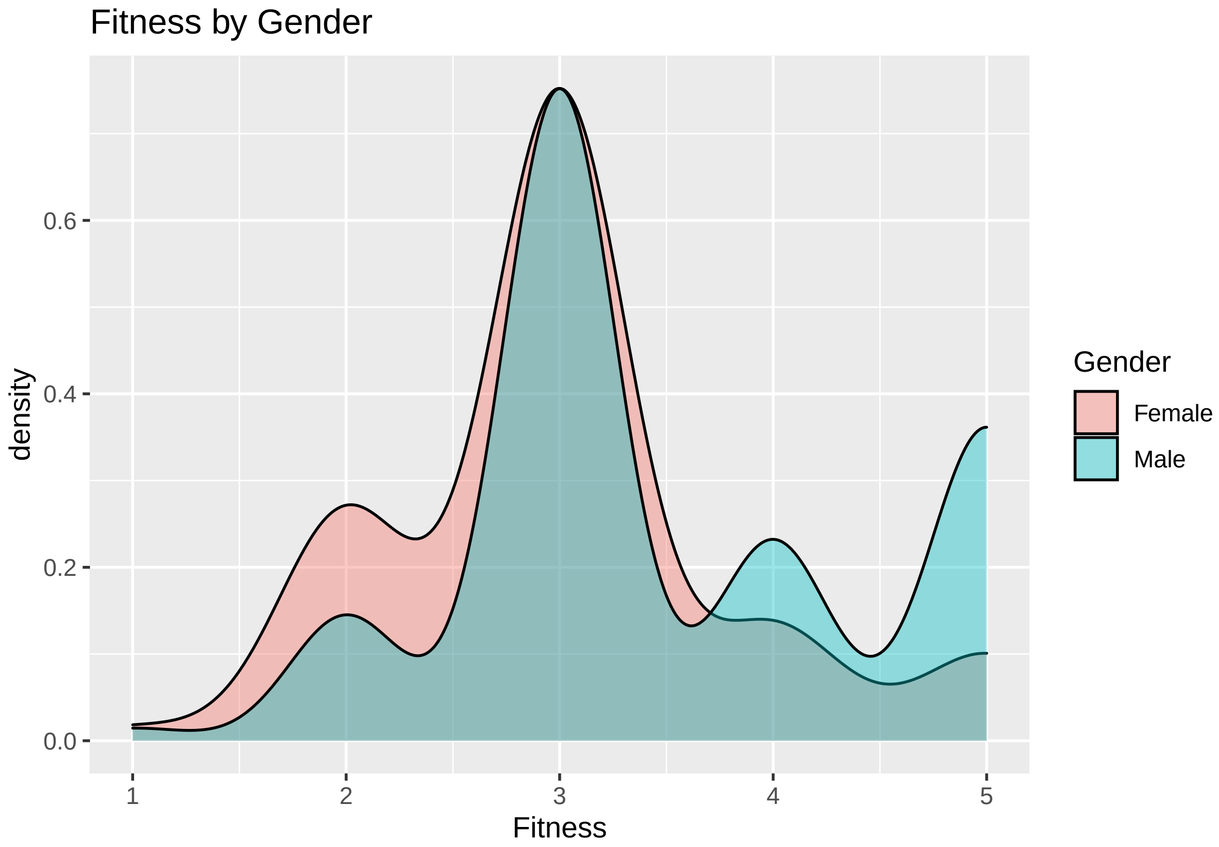
```
ggplot(Cardio_fitness,
       aes(x = Gender,
           y = Usage)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Usage by Gender")
```



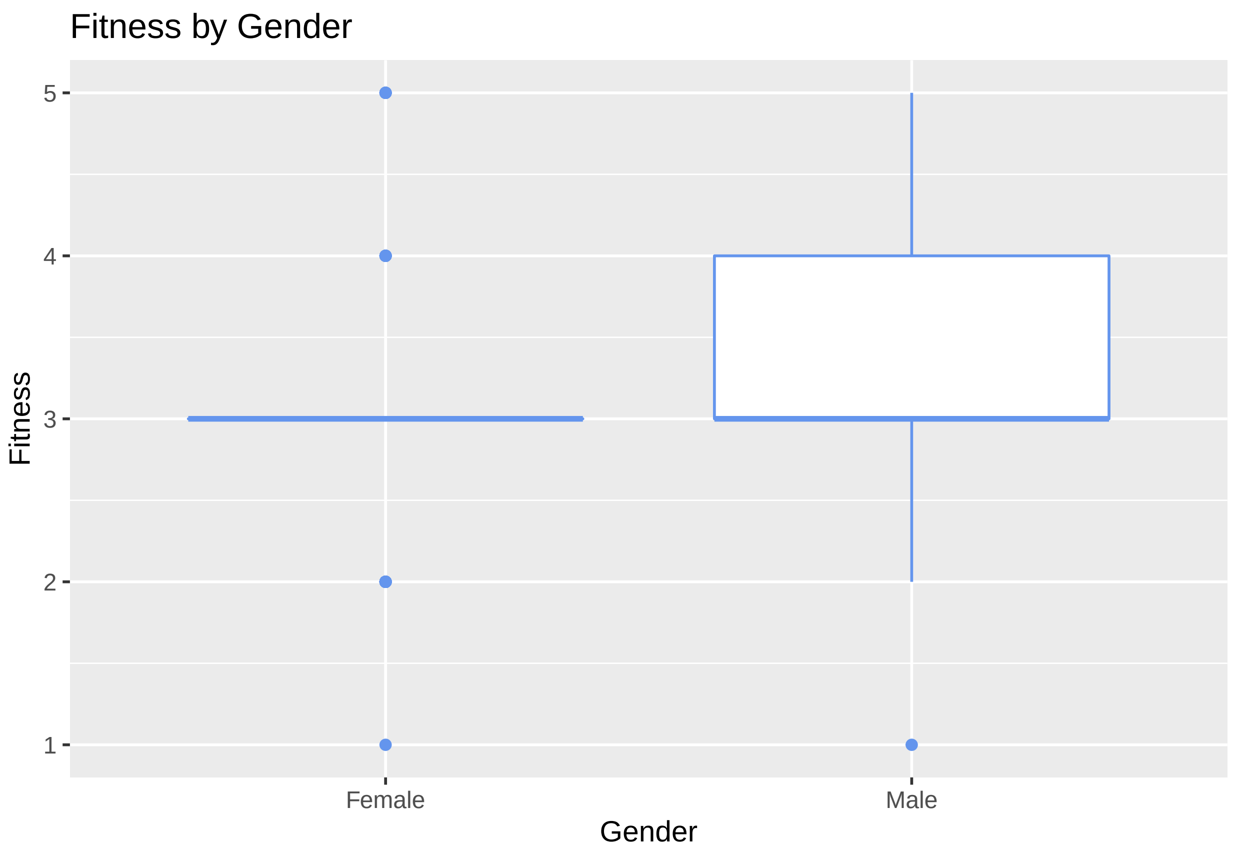
- Males make use of treadmill more than females.

2. Fitness

```
ggplot(Cardio_fitness,
       aes(x = Fitness,
           fill = Gender)) +
  geom_density(alpha = 0.4) +
  labs(title = "Fitness by Gender")
```



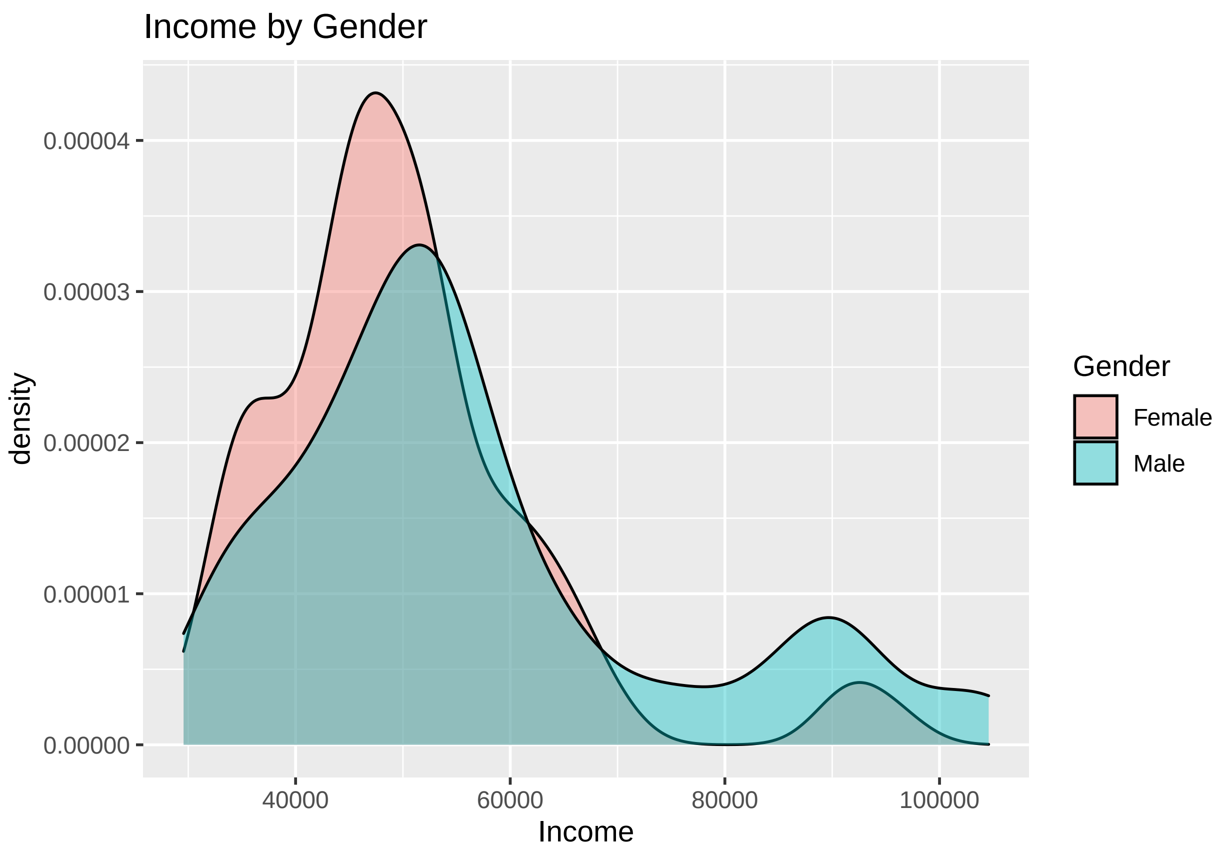
```
ggplot(Cardio_fitness,
       aes(x = Gender,
           y = Fitness)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Fitness by Gender")
```



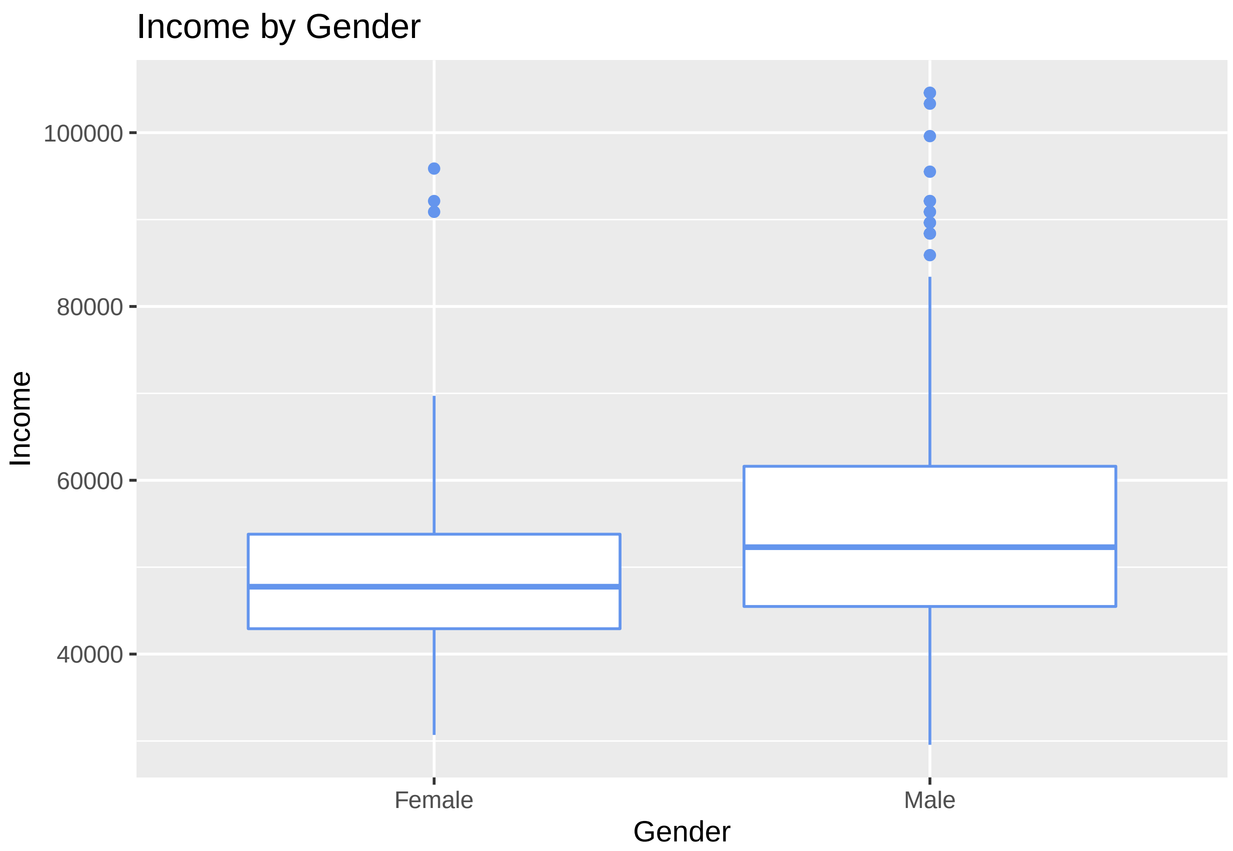
- Females maintain an average fitness shape of 3.
- Males have a higher fitness shape.

3. Income

```
ggplot(Cardio_fitness,
       aes(x = Income,
           fill = Gender)) +
  geom_density(alpha = 0.4) +
  labs(title = "Income by Gender")
```



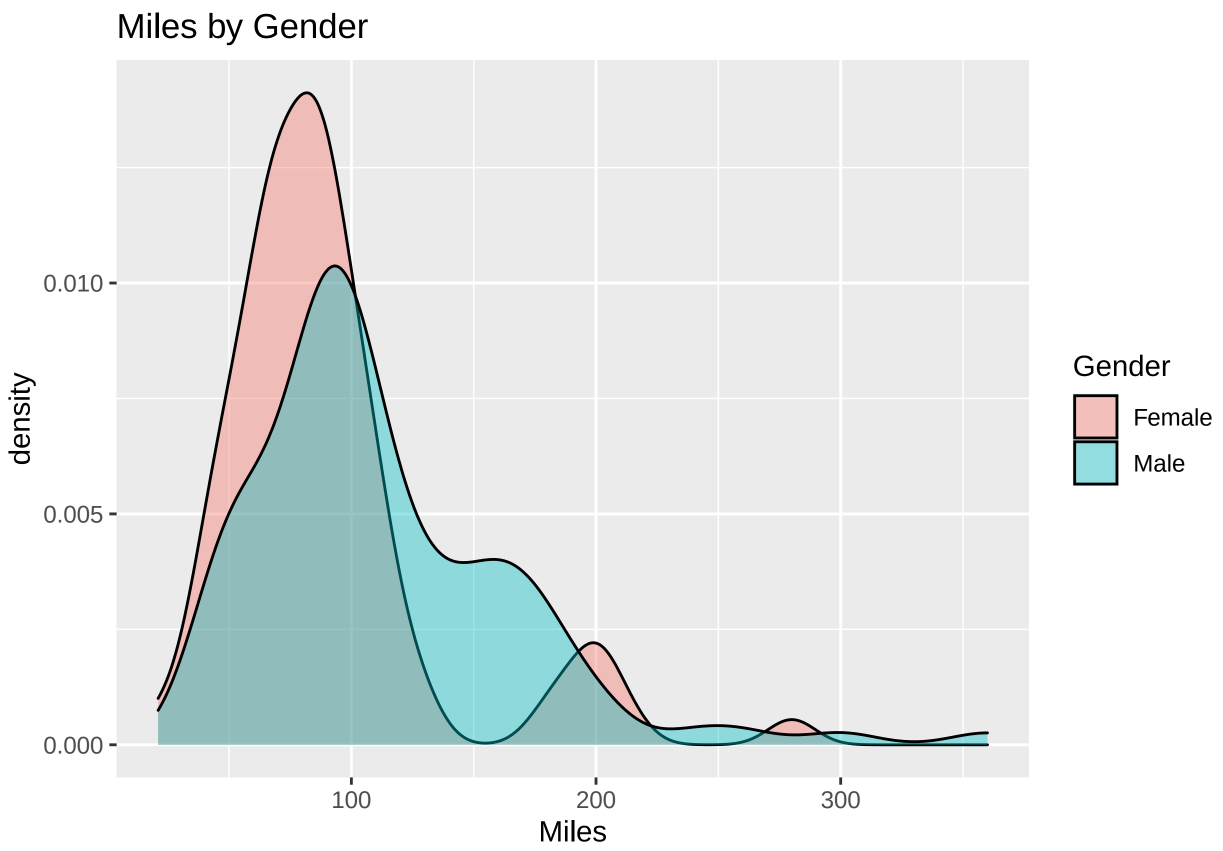
```
ggplot(Cardio_fitness,
       aes(x = Gender,
           y = Income)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Income by Gender")
```



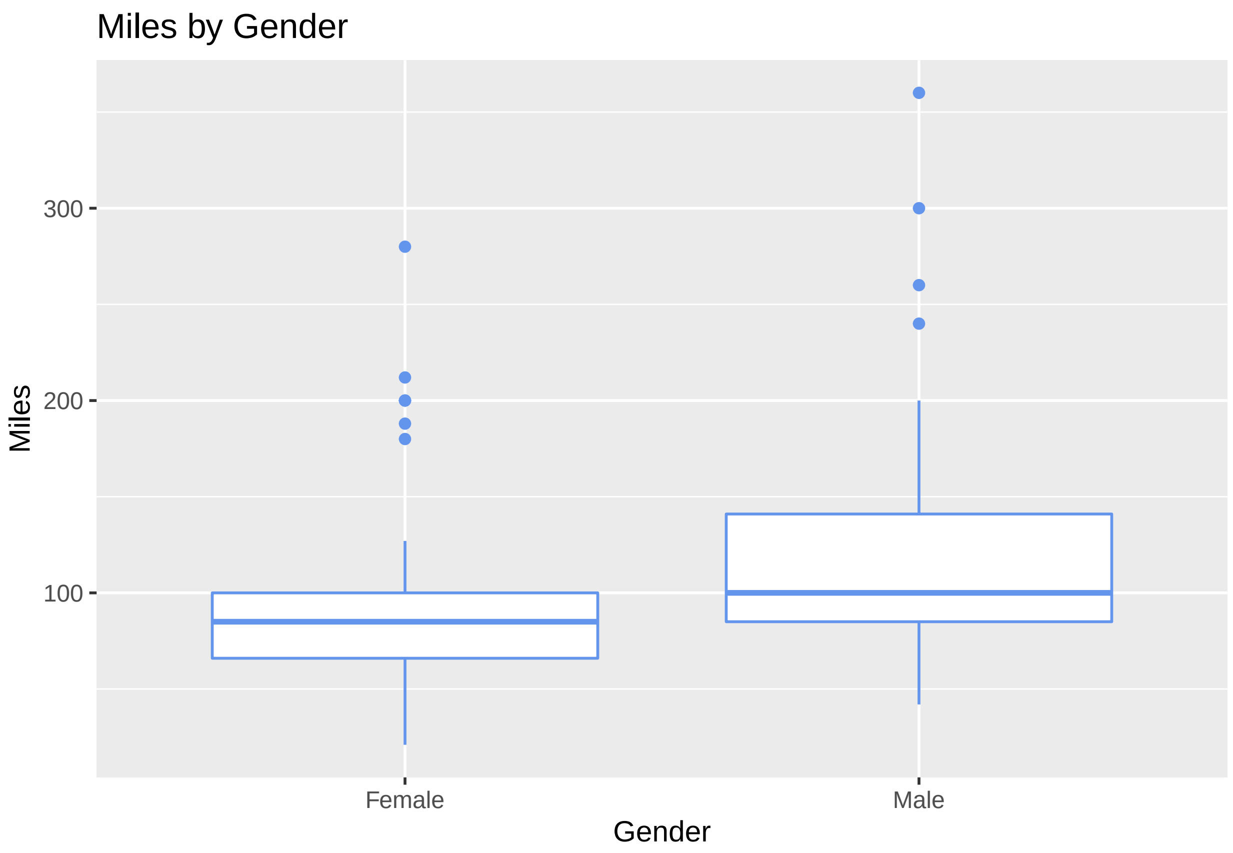
- There are more males in the > 70,000 income group.
- There are more females in the < 60,000 income group.

4. Miles

```
ggplot(Cardio_fitness,
       aes(x = Miles,
           fill = Gender)) +
  geom_density(alpha = 0.4) +
  labs(title = "Miles by Gender")
```



```
ggplot(Cardio_fitness,
       aes(x = Gender,
           y = Miles)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Miles by Gender")
```

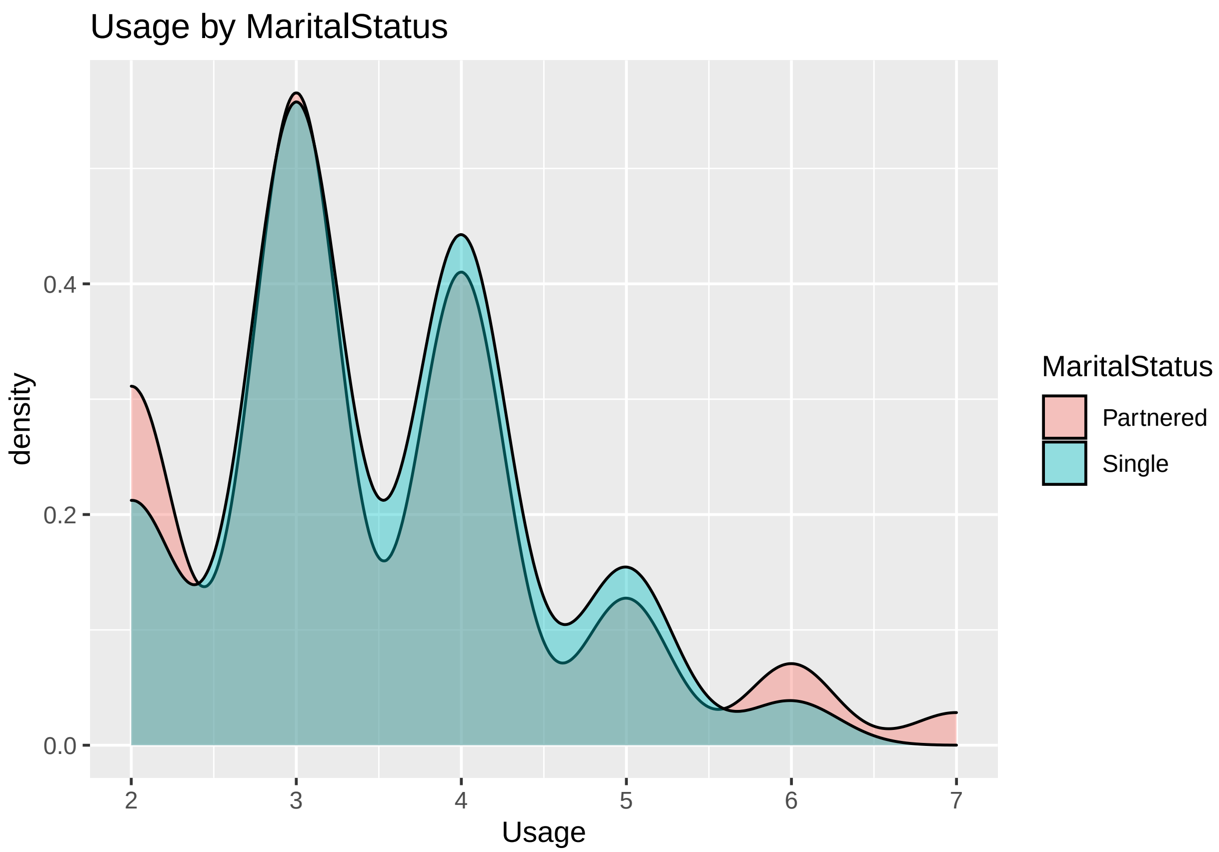


- Males cover more miles.
- Females tend to hit their peak in < 150 miles with the exception of outliers.

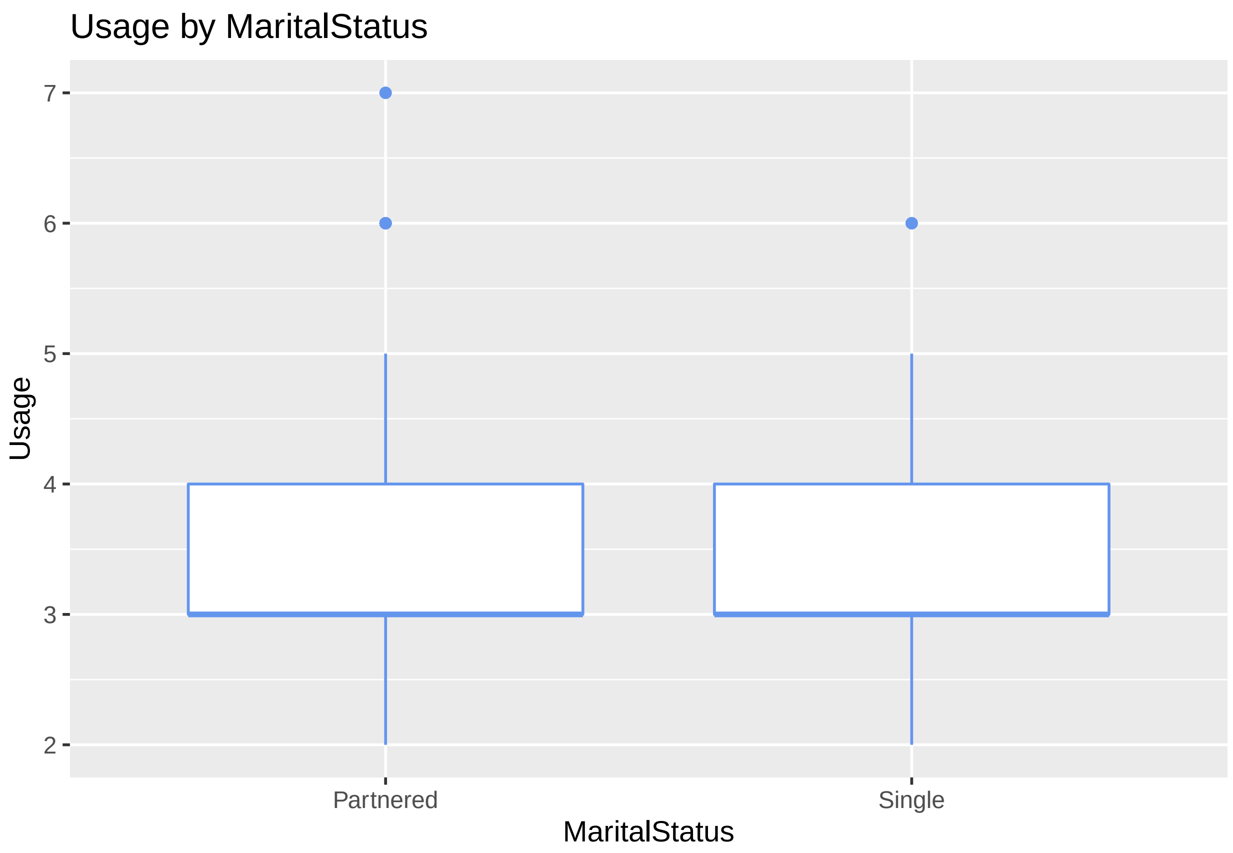
Relationship between MaritalStatus and numeric variables

1. Usage

```
ggplot(Cardio_fitness,
       aes(x = Usage,
           fill = MaritalStatus)) +
  geom_density(alpha = 0.4) +
  labs(title = "Usage by MaritalStatus")
```



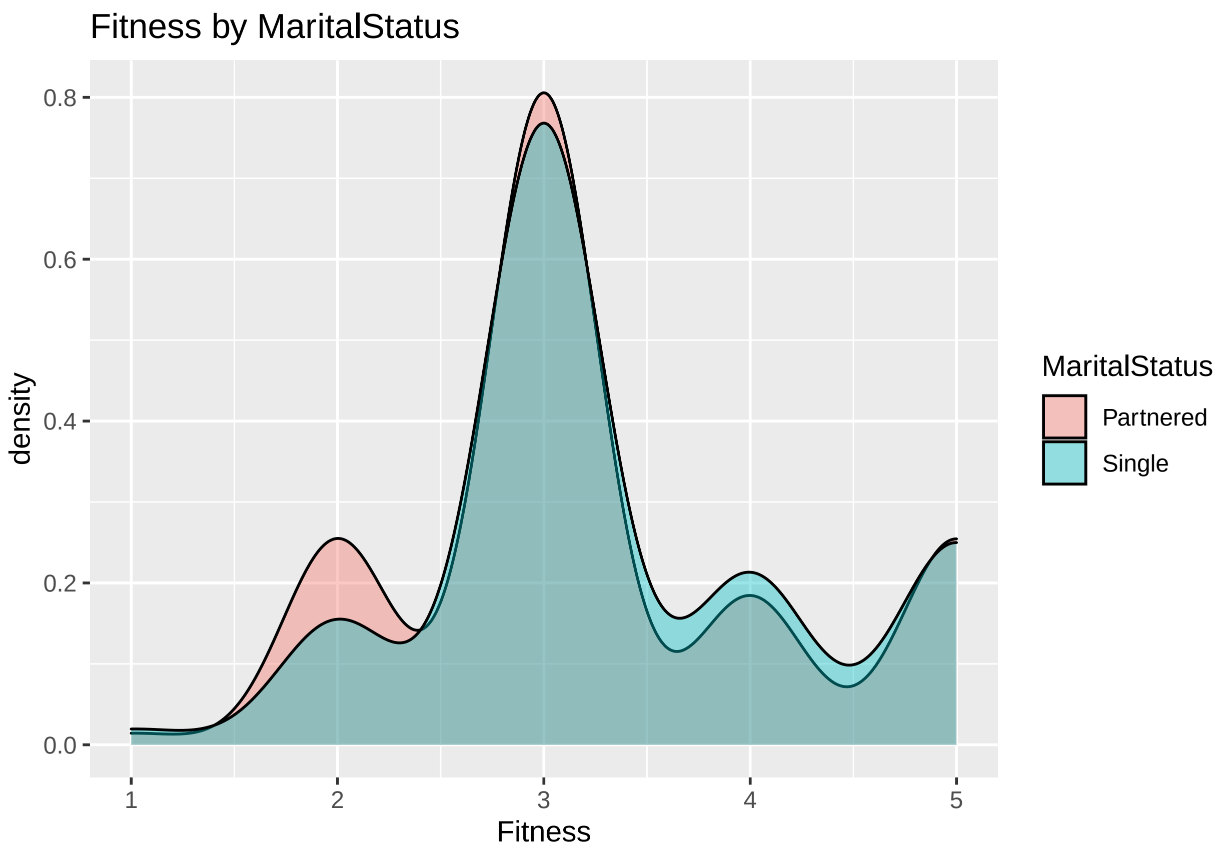
```
ggplot(Cardio_fitness,
       aes(x = MaritalStatus,
           y = Usage)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Usage by MaritalStatus")
```



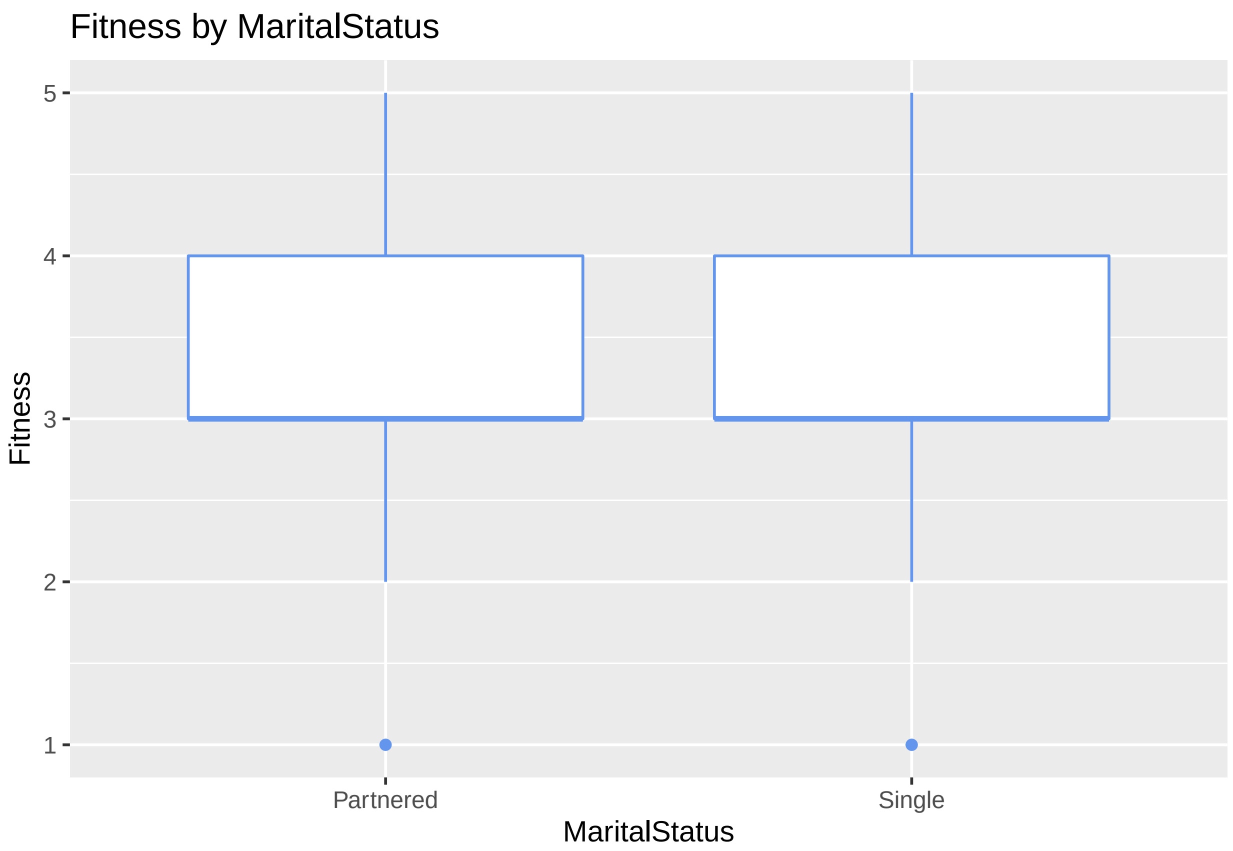
- Usage is fairly even between singles and partnered.

2. Fitness

```
ggplot(Cardio_fitness,
       aes(x = Fitness,
           fill = MaritalStatus)) +
  geom_density(alpha = 0.4) +
  labs(title = "Fitness by MaritalStatus")
```



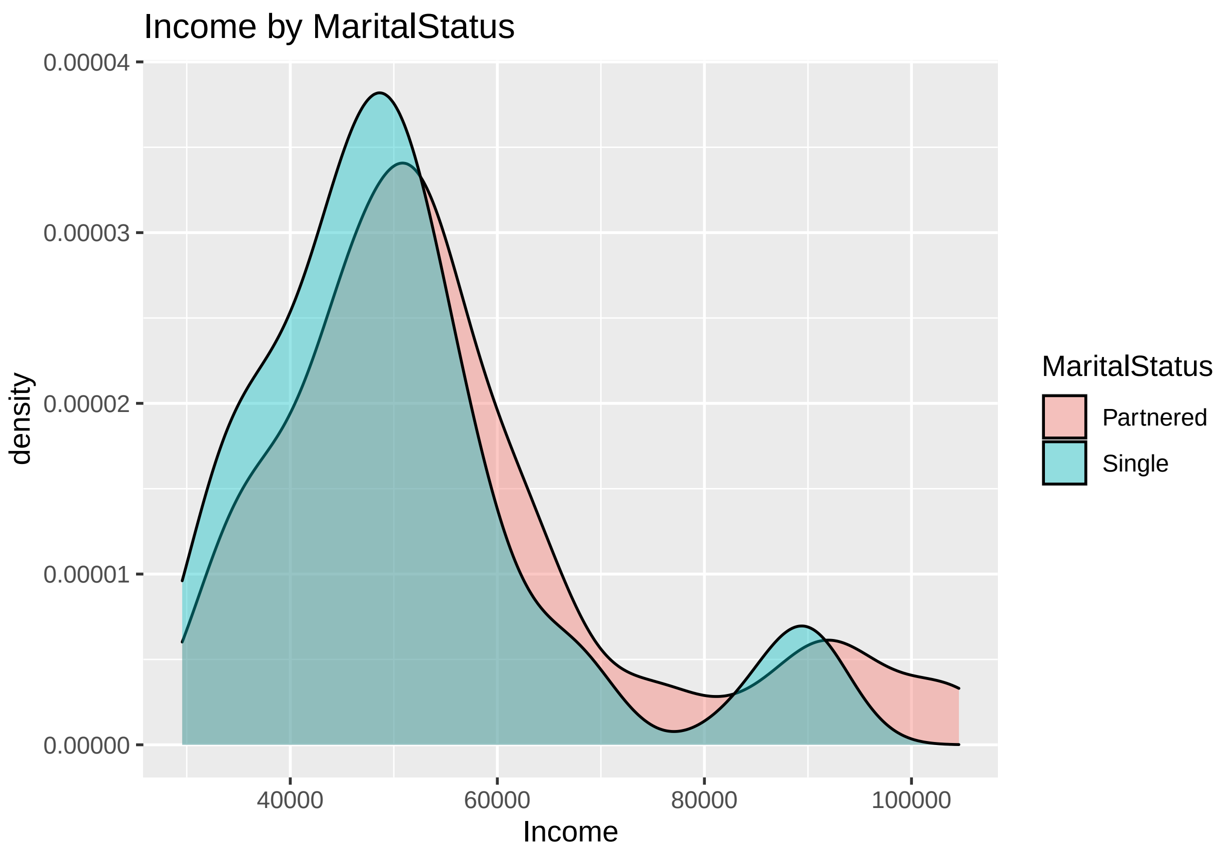
```
ggplot(Cardio_fitness,
       aes(x = MaritalStatus,
           y = Fitness)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Fitness by MaritalStatus")
```



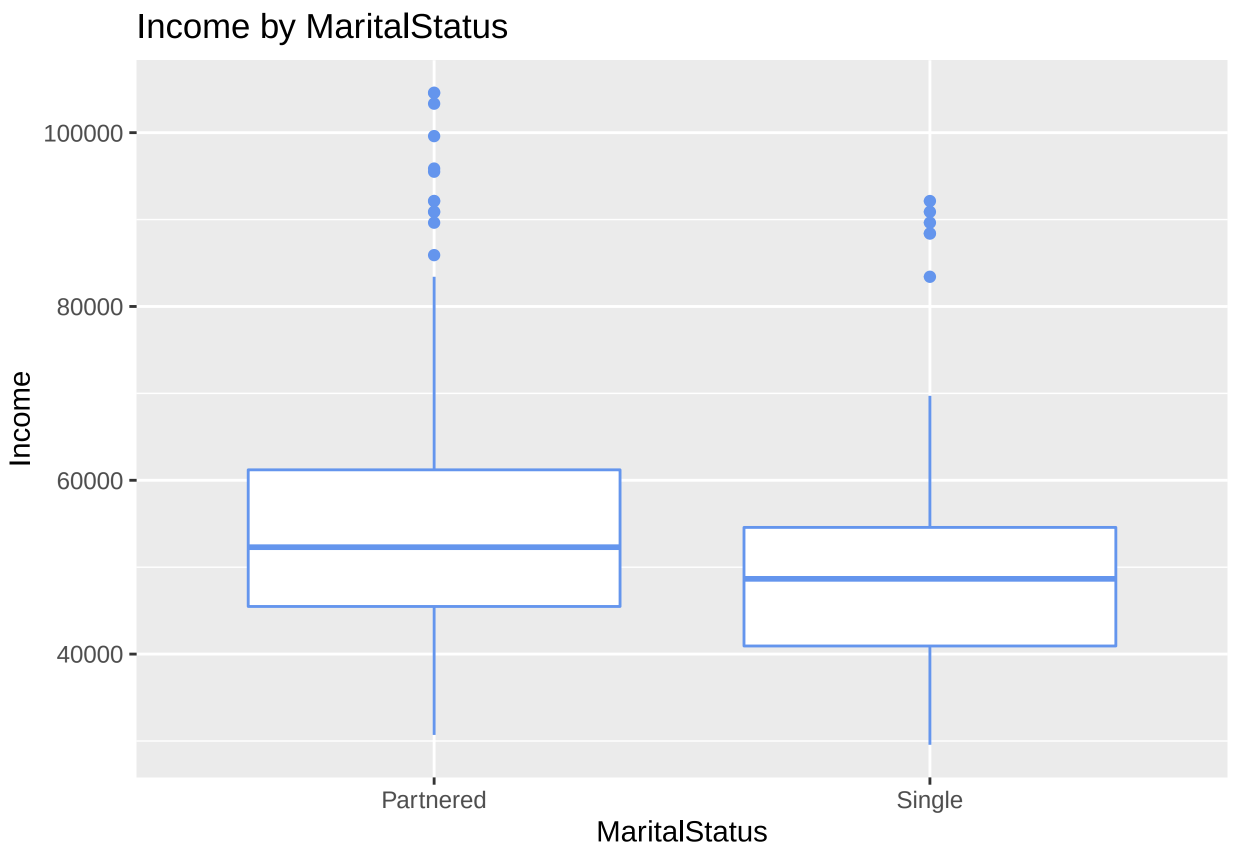
- Fitness shape is fairly even between singles and partnered.

3. Income

```
ggplot(Cardio_fitness,
       aes(x = Income,
           fill = MaritalStatus)) +
  geom_density(alpha = 0.4) +
  labs(title = "Income by MaritalStatus")
```



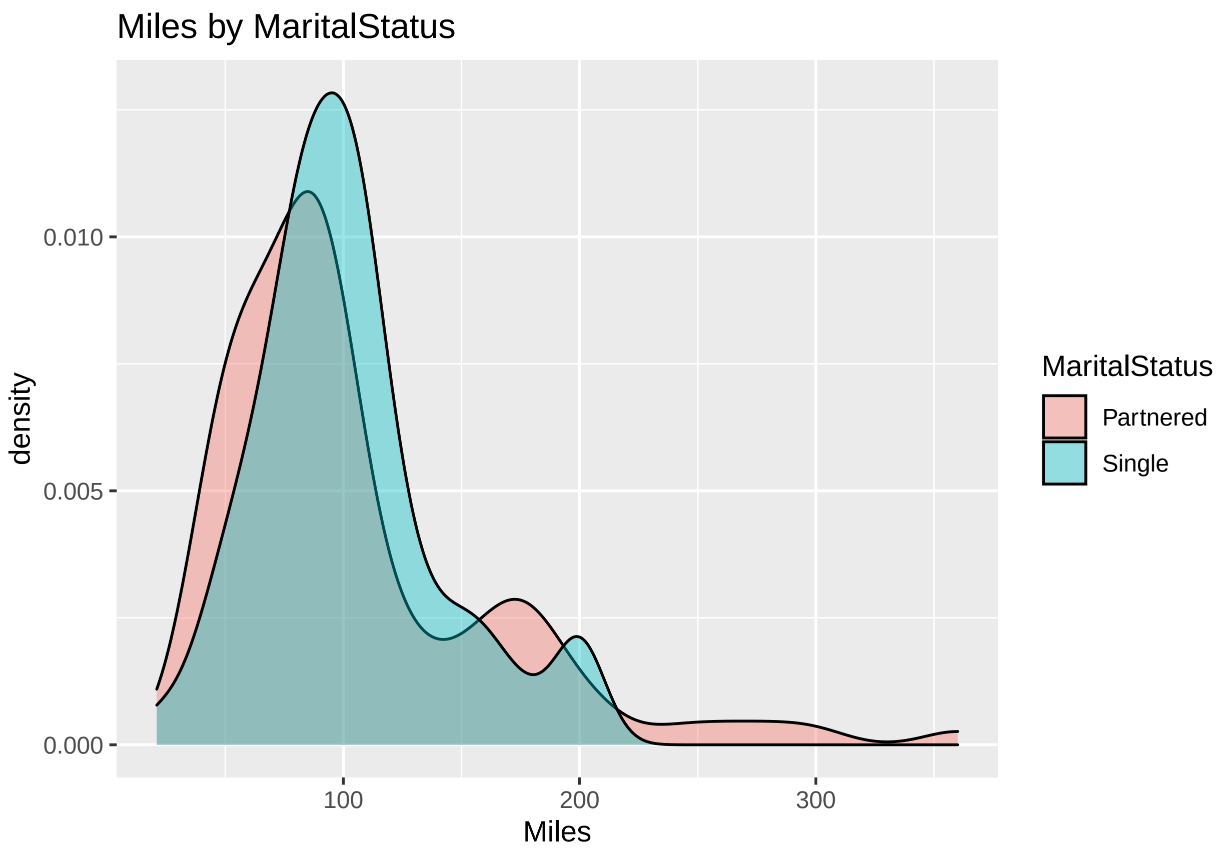
```
ggplot(Cardio_fitness,
       aes(x = MaritalStatus,
           y = Income)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Income by MaritalStatus")
```



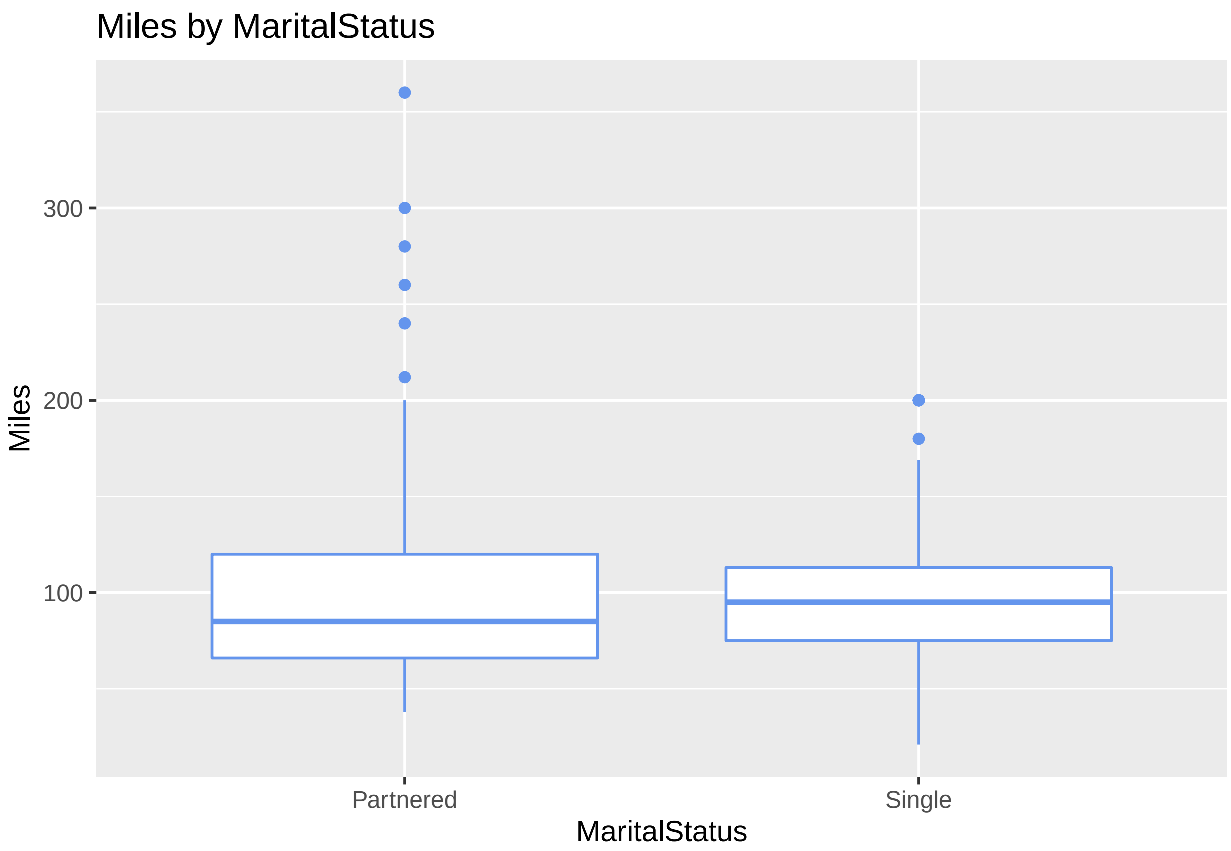
- Singles earn more than partnered in the $< 53{,}000$ income group.
- Partnered earn more than single in the $> 53{,}000$ income group.

4. Miles

```
ggplot(Cardio_fitness,
       aes(x = Miles,
           fill = MaritalStatus)) +
  geom_density(alpha = 0.4) +
  labs(title = "Miles by MaritalStatus")
```



```
ggplot(Cardio_fitness,
       aes(x = MaritalStatus,
           y = Miles)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Miles by MaritalStatus")
```



- Singles maintain a 0 < x < 200 miles with the exception of outliers.
- Partnered tend to cover more miles.

4. Conclusion and Recommendations

4.1 Conclusion

I analysed a dataset of 180 observations with 9 variables regarding a retail store, Cardio Goods Fitness, that sells treadmill products. The variables contain details of the product sold and profile of each customer such as age, gender, education, maritalstatus, usage, fitness, income and miles. The main feature of interest here is the product sold.

I have been able to conclude that

1. TM195 product is the most sold, followed by TM498 and TM798 respectively. This suggests that TM195 is affordable relative to TM798 which could be considered expensive.
2. Despite TM798 being the least sold, it records the highest fitness shape, usage and miles. This could indicate that it is tailored more to customers who exercise more.
3. Usgae and fitness have a high correlation with miles.
4. Younger customers exercise more compared to older customers.
5. Males purchase more treadmills than female. This could be a result of their higher income level.
6. Similarly, the data shows males exercise more than females and this is backed by higher usage, fitness shape and miles.
7. In terms of maritalstatus, there is a fairly even distribution across numeric variables. However, partnered customers tend to cover more miles relative to single customers.
8. Partnered customers purchase more product than single. This could be due to higher income from both spouses.

4.2 Recommendation to business

1. Demand for TM798 is highest among younger customers with high income level. Ensure its availability and increase sales.
2. Demand for TM195 is highest among customers with average fitness shape and average income level, and it makes up 44.4% of sales. Ensure availability and focus on supplying the product.
3. Target partnered customers as their stream of income is higher.
4. Given 57.8% of the customer base is male, focus on maximizing profit from here while seeking ways to increase interest among females. Likwise, apply same based on maritalstatus.
5. Procure data on price to get a better understanding of the revenue structure.

5. Appendix A – Source Code

```
#=======================================================================
#
# Exploratory Data Analysis - CardioGoodFitness
#
#=======================================================================

# Environment Set up and Data Import

# Invoking Libraries
library(readr) # To import csv files
library(ggplot2) # To create plots
library(corrplot) # To plot correlation plot between numerical variables
library(dplyr) # To manipulate dataset
library(gridExtra) # To plot multiple ggplot graphs in a grid
library(knitr) # Necessary to generate sourcecodes from a .Rmd File
library(markdown) # To convert to HTML
library(rmarkdown) # To convret analyses into high quality documents

# Set working directory
setwd("C:/Users/egwuc/Desktop/PGP-DSBA-UT Austin/Introduction to R/Week 4 - Project/")

# Read input file
Cardio_fitness <- read_csv("CardioGoodFitness.csv")

# Global options settings
options(scipen = 999) # turn off scientific notation like 1e+06

# Variable identification
# check dimension of dataset
dim(Cardio_fitness)

# check first 6 rows(observations) of dataset
head(Cardio_fitness)

# check last 6 rows(observations) of dataset
tail(Cardio_fitness)

# check structure of dataset
str(Cardio_fitness)

# change product, gender and maritalstatus to factor variable
Cardio_fitness$Product <- as.factor(Cardio_fitness$Product)
```

```
Cardio_fitness$Gender <- as.factor(Cardio_fitness$Gender)
Cardio_fitness$MaritalStatus <- as.factor(Cardio_fitness$MaritalStatus)

# get summary of dataset
summary(Cardio_fitness)

# View the dataset
View(Cardio_fitness)

ggplot(Cardio_fitness, aes(x = Product)) +
  geom_bar(fill = c("red"), color="black") +
  labs(x = "Product",
       y = "Frequency",
       title = "")

ggplot(Cardio_fitness, aes(x = Gender)) +
  geom_bar(fill = c("blue"), color="black") +
  labs(x = "Gender",
       y = "Frequency",
       title = "")

ggplot(Cardio_fitness, aes(x = MaritalStatus)) +
  geom_bar(fill = c("yellow"), color="black") +
  labs(x = "MaritalStatus",
       y = "Frequency",
       title = "")

plot_histogram_n_boxplot = function(variable, variableNameString, binw){

  a = ggplot(data = Cardio_fitness, aes(x= variable)) +
    labs(x = variableNameString,y ='frequency')+
    geom_histogram(fill = 'green',col = 'white', binwidth = binw) +
    geom_vline(aes(xintercept = mean(variable)),
               color = "black", linetype = "dashed", size = 0.5)

  b = ggplot(data = Cardio_fitness, aes('',variable))+
    geom_boxplot(outlier.colour = 'red',col = 'red', outlier.shape = 19)+
    labs(x = '', y = variableNameString) + coord_flip()
  grid.arrange(a,b,ncol = 2)
}

plot_histogram_n_boxplot(Cardio_fitness$Age, 'Age', 1)

plot_histogram_n_boxplot(Cardio_fitness$Education, 'Education', 1)

plot_histogram_n_boxplot(Cardio_fitness$Usage, 'Usage', 1)

plot_histogram_n_boxplot(Cardio_fitness$Fitness, 'Fitness', 1)

plot_histogram_n_boxplot(Cardio_fitness$Income, 'Income', 10000)

plot_histogram_n_boxplot(Cardio_fitness$Miles, 'Miles', 50)

ggplot(Cardio_fitness, aes(x = Gender, fill = Product)) +
  geom_bar(position = "dodge") +
  labs(y = "Count",
       fill = "Product",
       x = "Gender",
       title = "Gender by Product") +
  theme_minimal()

ggplot(Cardio_fitness, aes(x = MaritalStatus, fill = Product)) +
  geom_bar(position = "dodge") +
  labs(y = "Count",
       fill = "Product",
       x = "MaritalStatus",
       title = "MaritalStatus by Product") +
  theme_minimal()

# Numeric variables in the data
num_vars = sapply(Cardio_fitness, is.numeric)

# Correlation Plot
corrplot(cor(Cardio_fitness[,num_vars]), method = 'number')

plot_scatterplot = function(variableNameString, variable, binw){
  ggplot(data = Cardio_fitness, aes(x= Age, y = variable)) +
    labs(x = "Age", y = variableNameString) +
    geom_point(color="cornflowerblue", size =2, alpha=.8)+
    geom_smooth(method ="lm") # adds a linear trend line which is useful to summarize the relationship
}

grid.arrange(plot_scatterplot('Education', Cardio_fitness$Education),
             plot_scatterplot('Usage', Cardio_fitness$Usage),
             plot_scatterplot('Fitness', Cardio_fitness$Fitness),
             plot_scatterplot('Income', Cardio_fitness$Income),
             plot_scatterplot('Miles', Cardio_fitness$Miles),
             ncol = 3)

ggplot(Cardio_fitness,
       aes(x = Age,
           fill = Product)) +
  geom_density(alpha = 0.4) +
  labs(title = "Age by Product")


ggplot(Cardio_fitness,
       aes(x = Product,
           y = Age)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Age by Product")

ggplot(Cardio_fitness,
       aes(x = Usage,
           fill = Product)) +
  geom_density(alpha = 0.4) +
  labs(title = "Usage by Product")
```

```
ggplot(Cardio_fitness,
       aes(x = Product,
           y = Usage)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Usage by Product")

ggplot(Cardio_fitness,
       aes(x = Fitness,
           fill = Product)) +
  geom_density(alpha = 0.4) +
  labs(title = "Fitness by Product")


ggplot(Cardio_fitness,
       aes(x = Product,
           y = Fitness)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Fitness by Product")

ggplot(Cardio_fitness,
       aes(x = Income,
           fill = Product)) +
  geom_density(alpha = 0.4) +
  labs(title = "Income by Product")


ggplot(Cardio_fitness,
       aes(x = Product,
           y = Income)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Income by Product")

ggplot(Cardio_fitness,
       aes(x = Miles,
           fill = Product)) +
  geom_density(alpha = 0.4) +
  labs(title = "Miles by Product")


ggplot(Cardio_fitness,
       aes(x = Product,
           y = Miles)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Miles by Product")

ggplot(Cardio_fitness,
       aes(x = Usage,
           fill = Gender)) +
  geom_density(alpha = 0.4) +
  labs(title = "Usage by Gender")


ggplot(Cardio_fitness,
       aes(x = Gender,
           y = Usage)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Usage by Gender")

ggplot(Cardio_fitness,
       aes(x = Fitness,
           fill = Gender)) +
  geom_density(alpha = 0.4) +
  labs(title = "Fitness by Gender")


ggplot(Cardio_fitness,
       aes(x = Gender,
           y = Fitness)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Fitness by Gender")

ggplot(Cardio_fitness,
       aes(x = Income,
           fill = Gender)) +
  geom_density(alpha = 0.4) +
  labs(title = "Income by Gender")


ggplot(Cardio_fitness,
       aes(x = Gender,
           y = Income)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Income by Gender")

ggplot(Cardio_fitness,
       aes(x = Miles,
           fill = Gender)) +
  geom_density(alpha = 0.4) +
  labs(title = "Miles by Gender")


ggplot(Cardio_fitness,
       aes(x = Gender,
           y = Miles)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Miles by Gender")

ggplot(Cardio_fitness,
       aes(x = Usage,
           fill = MaritalStatus)) +
  geom_density(alpha = 0.4) +
  labs(title = "Usage by MaritalStatus")


ggplot(Cardio_fitness,
       aes(x = MaritalStatus,
           y = Usage)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Usage by MaritalStatus")

ggplot(Cardio_fitness,
       aes(x = Fitness,
           fill = MaritalStatus)) +
  geom_density(alpha = 0.4) +
  labs(title = "Fitness by MaritalStatus")


ggplot(Cardio_fitness,
       aes(x = MaritalStatus,
           y = Fitness)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Fitness by MaritalStatus")

ggplot(Cardio_fitness,
       aes(x = Income,
           fill = MaritalStatus)) +
  geom_density(alpha = 0.4) +
  labs(title = "Income by MaritalStatus")


ggplot(Cardio_fitness,
       aes(x = MaritalStatus,
           y = Income)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Income by MaritalStatus")

ggplot(Cardio_fitness,
       aes(x = Miles,
           fill = MaritalStatus)) +
  geom_density(alpha = 0.4) +
  labs(title = "Miles by MaritalStatus")


ggplot(Cardio_fitness,
       aes(x = MaritalStatus,
           y = Miles)) +
  geom_boxplot(color = "cornflowerblue") +
  labs(title = "Miles by MaritalStatus")

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

Generate .R file from this Rmd. The .R will contain only the R source code.

```
# Generate the .R file from this .Rmd to hold the source code

purl("Cardio Good Fitness Project.Rmd", documentation = 0)
```

To create word or pdf report -> click on Knit in the toolbar above, select knit to pdf.