CMM703 - Data Analysis Coursework

Kaneel Dias

2025-04-14

```r
suppressWarnings(suppressMessages({
  library(ggplot2)
  require(gridExtra)
  library(glue)
  library(ggcorrplot)
  library(vcd)
  library(tidyr)
  library(dplyr)
  library(pheatmap)
  library(caTools)
  library(pROC)
}))
```

TASK 1: CANDY DATASET

The objective of this exercise would be to determine the effect that the variables contained within the dataset have on the target value `winpercent`. All visualizations included in this report will be made with that objective in mind.

```r
candy_data <- read.csv("~/CMM703/candy-data.csv")
```

1.1 Effect of the categorical variables

First we should convert all the categorical columns, from numeric to factor.

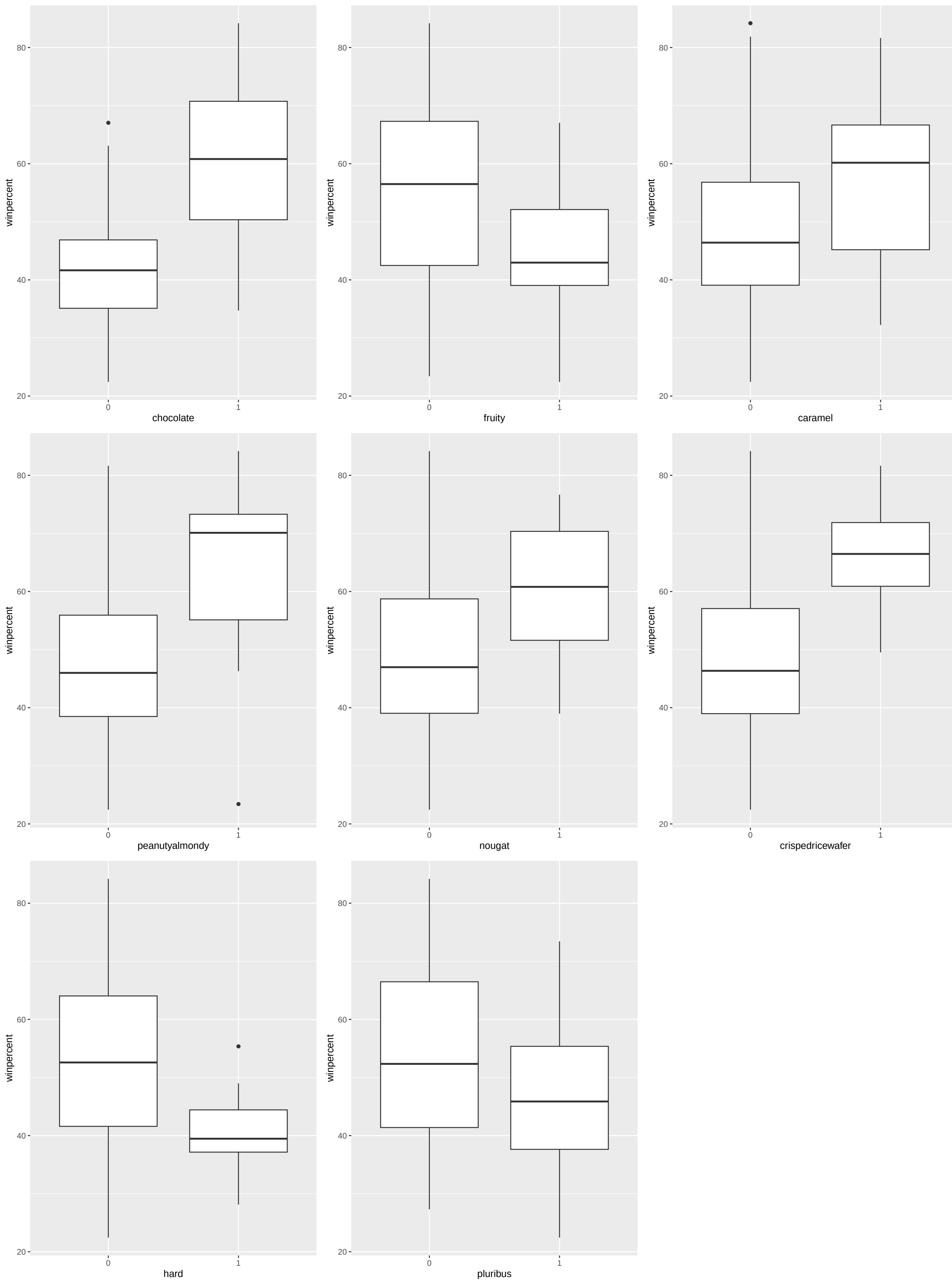
```r
candy_data$chocolate <- as.factor(candy_data$chocolate)
candy_data$fruity <- as.factor(candy_data$fruity)
candy_data$caramel <- as.factor(candy_data$caramel)
candy_data$peanutyalmondy <- as.factor(candy_data$peanutyalmondy)
candy_data$nougat <- as.factor(candy_data$nougat)
candy_data$crispedricewafer <- as.factor(candy_data$crispedricewafer)
candy_data$hard <- as.factor(candy_data$hard)
candy_data$bar <- as.factor(candy_data$bar)
candy_data$pluribus <- as.factor(candy_data$pluribus)
```

Next, we can plot boxplots for each categorical variable, and the effect it has on the winning percentage.

```r
plot_categorical_boxplot <- function(data, variable) {
  plot <- ggplot(data=data, aes(x=data[,variable], y=winpercent)) +
    xlab(variable) +
    geom_boxplot()

  return(plot)
```

```r
}

plot_all_categorical_boxplots <- function(data) {
  chocolate_plot <- plot_categorical_boxplot(data, "chocolate")
  fruity_plot <- plot_categorical_boxplot(data, "fruity")
  caramel_plot <- plot_categorical_boxplot(data, "caramel")
  peanutyalmondy_plot <- plot_categorical_boxplot(data, "peanutyalmondy")
  nougat_plot <- plot_categorical_boxplot(data, "nougat")
  crispedricewafer_plot <- plot_categorical_boxplot(data, "crispedricewafer")
  hard_plot <- plot_categorical_boxplot(data, "hard")
  pluribus_plot <- plot_categorical_boxplot(data, "pluribus")


  grid.arrange(chocolate_plot, fruity_plot, caramel_plot, peanutyalmondy_plot, nougat_plot, crispedrice
}
```

```r
plot_all_categorical_boxplots(candy_data)
```


winpercent
chocolate
fruity
caramel
peanutyalmondy
nougat
crispedricewafer
hard
pluribus
20
40
60
80
0
1

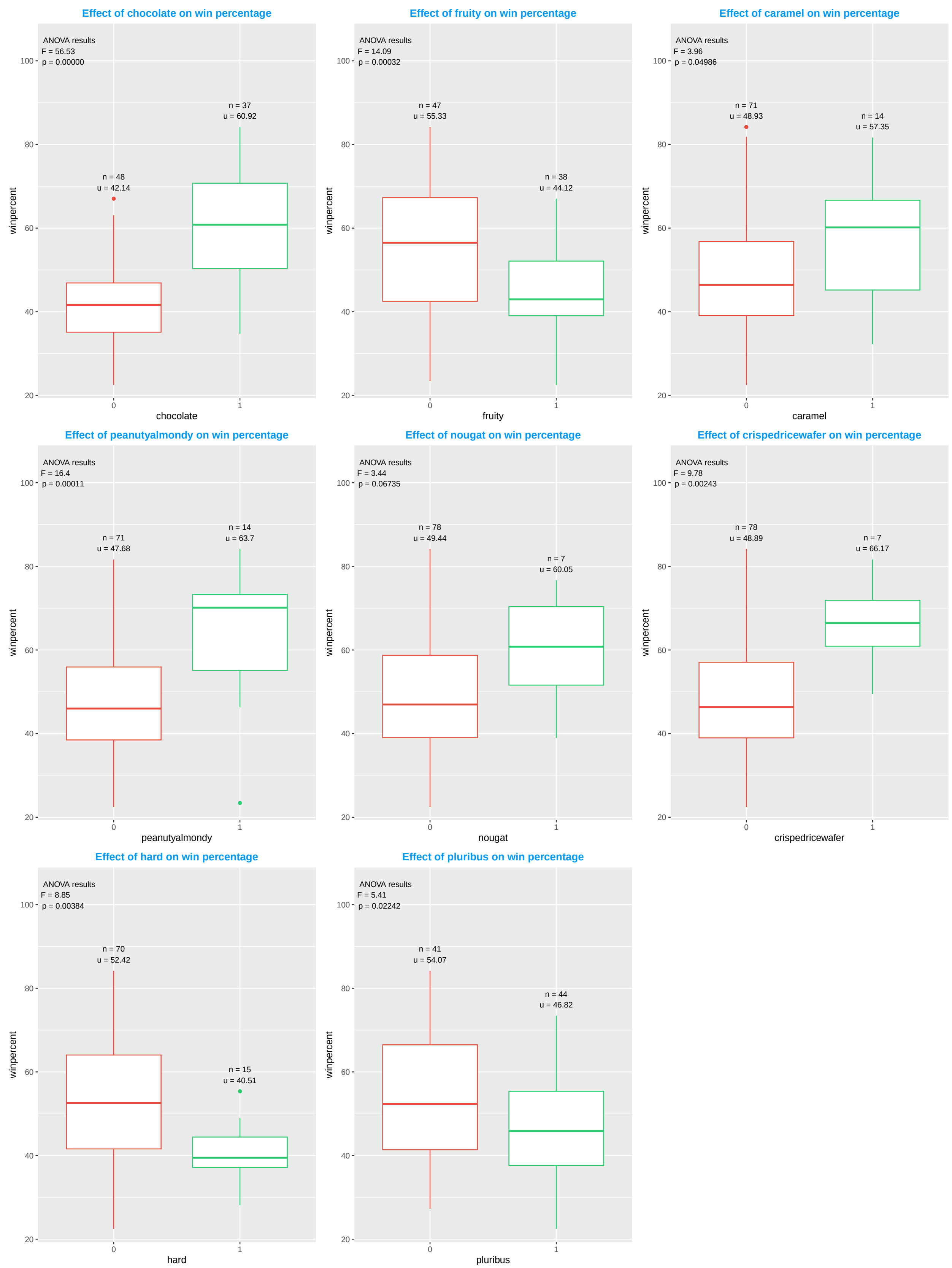
1.1.1 Potential improvements

We can suggest the following improvements
- Visually separate the `contains (1)` and `does not contain (0)` plots using colours
- Indicate for each boxplot, the
- Count of records with that value
- The mean of the `winpercent` value
- Indicate the effect that variable has on the `winpercent` using ANOVA
- Include the F value
- Include the P value

```
get_summary_stats <- function(y) {
  bxp_stats <- boxplot.stats(y)
  upper_whisker <- bxp_stats$stats[5]
  max_val <- max(c(upper_whisker, bxp_stats$out), na.rm = TRUE)

  n <- length(y)
  q1 <- quantile(y, 0.25, na.rm = TRUE, names = FALSE)
  avg <- mean(y, na.rm = TRUE)
  q3 <- quantile(y, 0.75, na.rm = TRUE, names = FALSE)

  label_str <- paste(
    paste("n =", n),
    paste("u =", round(avg, 2)),
    sep = "\n"
  )

  return(data.frame(
    y = max_val,
    label = label_str
  ))
}

plot_categorical_boxplot_improved <- function(data, variable) {
  anova <- aov(reformulate(variable, "winpercent"), data=data)
  p_val <- summary(anova)[[1]][["Pr(>F)"]][1]
  f_val <- summary(anova)[[1]][["F value"]][1]
  anova_text <- glue("ANOVA results\nF = {round(f_val, 2)}\np = {format(round(p_val, 5), nsmall = 5)}")

  plot <- ggplot(data=data, aes(x=data[,variable], y=winpercent, col=data[,variable])) +
    labs(
      title = glue('Effect of {variable} on win percentage'),
      x = variable
    ) +
    geom_boxplot() +
    scale_color_manual(values = c("#e74c3c", "#2ecc71")) +
    theme(
      legend.position="none",
      plot.title = element_text(color = "#0099f8", size = 12, face = "bold", hjust = 0.5),
    )

  plot <- plot + stat_summary(
    fun.data = get_summary_stats,
    geom = "text",
    hjust = 0.5,
```

```
    vjust = -0.5,
    size = 3,
    color = "black"
  )

  plot <- plot + annotate(
    geom = "text",
    x = -Inf,
    y = Inf,
    label = anova_text,
    hjust = -0.1,
    vjust = 1.5,
    size = 3,
    color = "black"
  ) +
    scale_y_continuous(expand = expansion(mult = c(0.05, 0.4))) # More space at top


  return(plot)
}

plot_all_categorical_boxplots_improved <- function(data) {
  chocolate_plot <- plot_categorical_boxplot_improved(data, "chocolate")
  fruity_plot <- plot_categorical_boxplot_improved(data, "fruity")
  caramel_plot <- plot_categorical_boxplot_improved(data, "caramel")
  peanutyalmondy_plot <- plot_categorical_boxplot_improved(data, "peanutyalmondy")
  nougat_plot <- plot_categorical_boxplot_improved(data, "nougat")
  crispedricewafer_plot <- plot_categorical_boxplot_improved(data, "crispedricewafer")
  hard_plot <- plot_categorical_boxplot_improved(data, "hard")
  pluribus_plot <- plot_categorical_boxplot_improved(data, "pluribus")


  grid.arrange(chocolate_plot, fruity_plot, caramel_plot, peanutyalmondy_plot, nougat_plot, crispedrice
}
```

```
plot_all_categorical_boxplots_improved(candy_data)
```


Effect of chocolate on win percentage
ANOVA results
F = 56.53
p = 0.00000
n = 48
u = 42.14
n = 37
u = 60.92
winpercent
chocolate
Effect of fruity on win percentage
ANOVA results
F = 14.09
p = 0.00032
n = 47
u = 55.33
n = 38
u = 44.12
winpercent
fruity
Effect of caramel on win percentage
ANOVA results
F = 3.96
p = 0.04986
n = 71
u = 48.93
n = 14
u = 57.35
winpercent
caramel
Effect of peanutyalmondy on win percentage
ANOVA results
F = 16.4
p = 0.00011
n = 71
u = 47.68
n = 14
u = 63.7
winpercent
peanutyalmondy
Effect of nougat on win percentage
ANOVA results
F = 3.44
p = 0.06735
n = 78
u = 49.44
n = 7
u = 60.05
winpercent
nougat
Effect of crispedricewafer on win percentage
ANOVA results
F = 9.78
p = 0.00243
n = 78
u = 48.89
n = 7
u = 66.17
winpercent
crispedricewafer
Effect of hard on win percentage
ANOVA results
F = 8.85
p = 0.00384
n = 70
u = 52.42
n = 15
u = 40.51
winpercent
hard
Effect of pluribus on win percentage
ANOVA results
F = 5.41
p = 0.02242
n = 41
u = 54.07
n = 44
u = 46.82
winpercent
pluribus

1.1.2 Insights

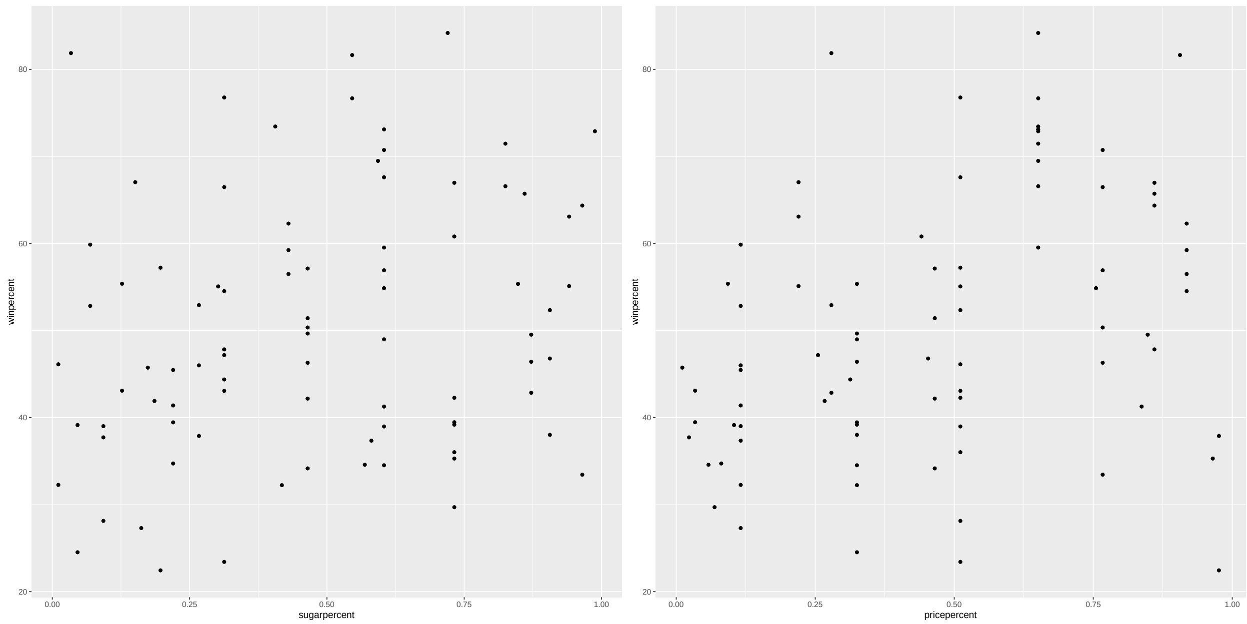
Here we can see that the `chocolate` variable has the highest effect on `winpercent` (highest f-value), and it has a very low p-value as well, indicating that it is most likely to be having an effect. On the other hand, `nougat` has a p-value > 0.05, (as well as a low f-value) which indicates that its effect on the winning percentage is not likely.

1.2 Effect of the numeric variables

There are two numeric, continuous variables: `sugarpercent` and `pricepercent` We can visualize their effect on `winpercent` using scatter plots.

```
plot_numeric_scatterplot <- function(data, variable) {
  plot <- ggplot(data=candy_data, aes(x=data[,variable], y=winpercent)) +
    labs(
      x = variable
    ) +
    geom_point()

  return (plot)
}

plot_all_numerical_scatterplots <- function(data) {
  sugar_plot <- plot_numeric_scatterplot(data, "sugarpercent")
  price_plot <- plot_numeric_scatterplot(data, "pricepercent")



  grid.arrange(sugar_plot, price_plot, nrow = 1)
}
```

```
plot_all_numerical_scatterplots(candy_data)
```




```
get_r2 <- function (x, y) cor(x, y) ^ 2
```

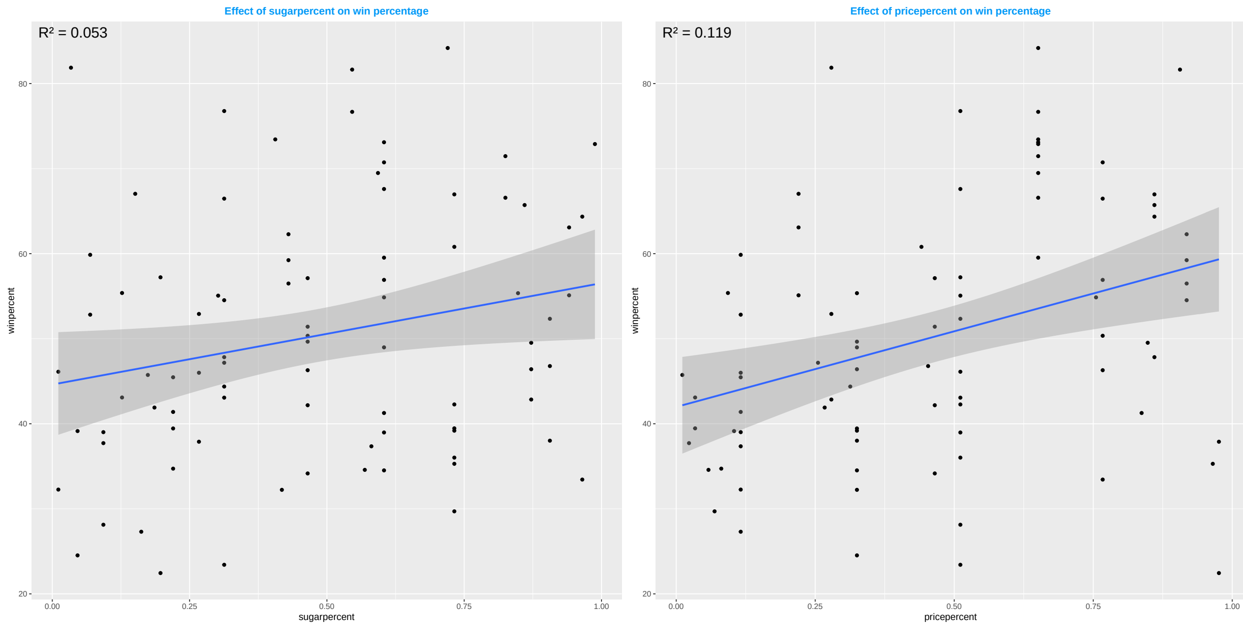
1.2.1 Potential Improvements

We can suggest the following improvements:
- Add a regression line to be able to view the relationship between the two variables and the winning percentage
- Include the R^2 value (coeffiecient of determination in the chart) to determine whether the they correlate

```
plot_numeric_scatterplot_improved <- function(data, variable) {
  r2 <- get_r2(data[,variable], data$winpercent)
  r2_text <- glue("R² = {format(round(r2, 3), nsmall = 3)}")


  plot <- ggplot(data=candy_data, aes(x=data[,variable], y=winpercent)) +
    labs(
      title = glue('Effect of {variable} on win percentage'),
      x = variable
    ) +
    geom_point() +
    geom_smooth(method=lm) +
    theme(
      legend.position="none",
      plot.title = element_text(color = "#0099f8", size = 12, face = "bold", hjust = 0.5),
    )

  plot <- plot + annotate(
    geom = "text",
    x = -Inf,
    y = Inf,
    label = r2_text,
    hjust = -0.1,
    vjust = 1.5,
    size = 6,
    color = "black"
  )

  return (plot)
}

plot_all_numerical_scatterplots_improved <- function(data) {
  sugar_plot <- plot_numeric_scatterplot_improved(data, "sugarpercent")
  price_plot <- plot_numeric_scatterplot_improved(data, "pricepercent")


  grid.arrange(sugar_plot, price_plot, nrow = 1)
}
```

```
plot_all_numerical_scatterplots_improved(candy_data)
```

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



1.2.2 Insights

From the above two scatter plots, even though we can see a slight positive correlation with `winpercent` for each of the two variables, they are quite insignificant. Therefore, we can conclude that thereis no significant correlation present.

TASK 2: Bank Churn

2.1 Exploratory Data Analysis

```
bank_churn_data <- read.csv("~/CMM703/Bank_Churn.csv")
```

After loading the dataset, we can view a summary of all the variables

```
summary(bank_churn_data)
```

```
##    CustomerId         Surname            CreditScore      Geography
##  Min.   :15565701   Length:10000       Min.   :350.0   Length:10000
##  1st Qu.:15628528   Class :character   1st Qu.:584.0   Class :character
##  Median :15690738   Mode  :character   Median :652.0   Mode  :character
##  Mean   :15690941                      Mean   :650.5
##  3rd Qu.:15753234                      3rd Qu.:718.0
##  Max.   :15815690                      Max.   :850.0
##     Gender               Age            Tenure          Balance
##  Length:10000       Min.   :18.00   Min.   : 0.000   Min.   :     0
##  Class :character   1st Qu.:32.00   1st Qu.: 3.000   1st Qu.:     0
##  Mode  :character   Median :37.00   Median : 5.000   Median : 97199
##                     Mean   :38.92   Mean   : 5.013   Mean   : 76486
##                     3rd Qu.:44.00   3rd Qu.: 7.000   3rd Qu.:127644
##                     Max.   :92.00   Max.   :10.000   Max.   :250898
##  NumOfProducts    HasCrCard      IsActiveMember   EstimatedSalary
##  Min.   :1.00   Min.   :0.0000   Min.   :0.0000   Min.   :    11.58
##  1st Qu.:1.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 51002.11
##  Median :1.00   Median :1.0000   Median :1.0000   Median :100193.91
##  Mean   :1.53   Mean   :0.7055   Mean   :0.5151   Mean   :100090.24
##  3rd Qu.:2.00   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:149388.25
##  Max.   :4.00   Max.   :1.0000   Max.   :1.0000   Max.   :199992.48
##      Exited
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2037
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

However, it would be much easier to visualize the data through plots

2.1.1 Visualizing numerical data

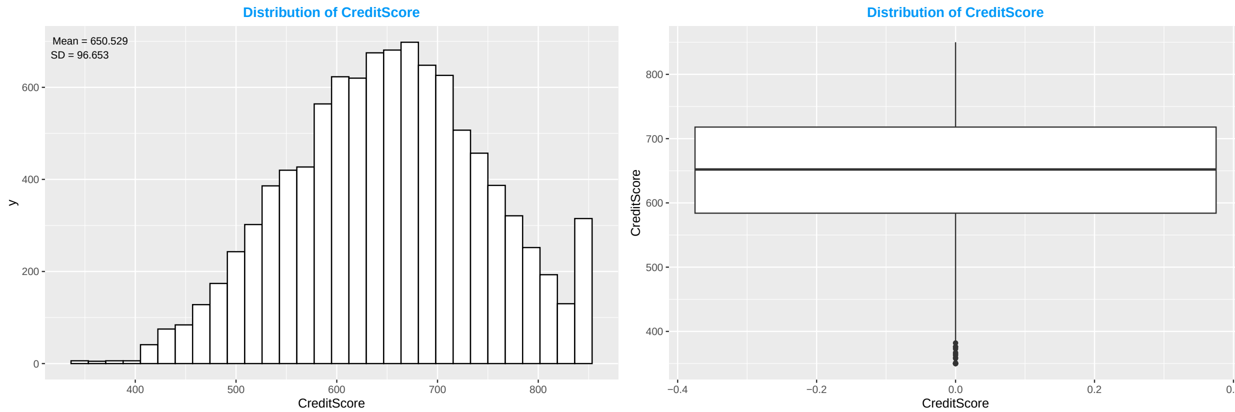
We can visualize numerical data using histograms and box plots.

```
plot_histogram <- function(data, variable) {
  mean = mean(data[,variable])
  sd = sd(data[,variable])
  summary_text = glue("Mean = {format(round(mean, 3), nsmall = 3)}\nSD = {format(round(sd, 3), nsmall =

  plot <- ggplot(bank_churn_data, aes(x=data[,variable])) +
    geom_histogram(color="black", fill="white") +
    labs(
      title = glue('Distribution of {variable}'),
      x = variable
    ) +
```

```
    theme(
      plot.title = element_text(color = "#0099f8", size = 12, face = "bold", hjust = 0.5),
    )

  plot <- plot + annotate(
    geom = "text",
    x = -Inf,
    y = Inf,
    label = summary_text,
    hjust = -0.1,
    vjust = 1.5,
    size = 3,
    color = "black"
  )


  return (plot)
}


plot_boxplot <- function(data, variable) {
  plot <- ggplot(data=data, aes(y=data[,variable])) +
    labs(
      title = glue('Distribution of {variable}'),
      x = variable,
      y = variable
    ) +
    geom_boxplot() +
    theme(
      plot.title = element_text(color = "#0099f8", size = 12, face = "bold", hjust = 0.5),
    )


  return (plot)
}

plot_numerical <- function(data, variable) {
  histogram <- suppressMessages(plot_histogram(bank_churn_data, variable))
  boxplot <- plot_boxplot(bank_churn_data, variable)

  grid.arrange(histogram, boxplot, nrow = 1)
}
```

```
plot_numerical(bank_churn_data, "CreditScore")
```

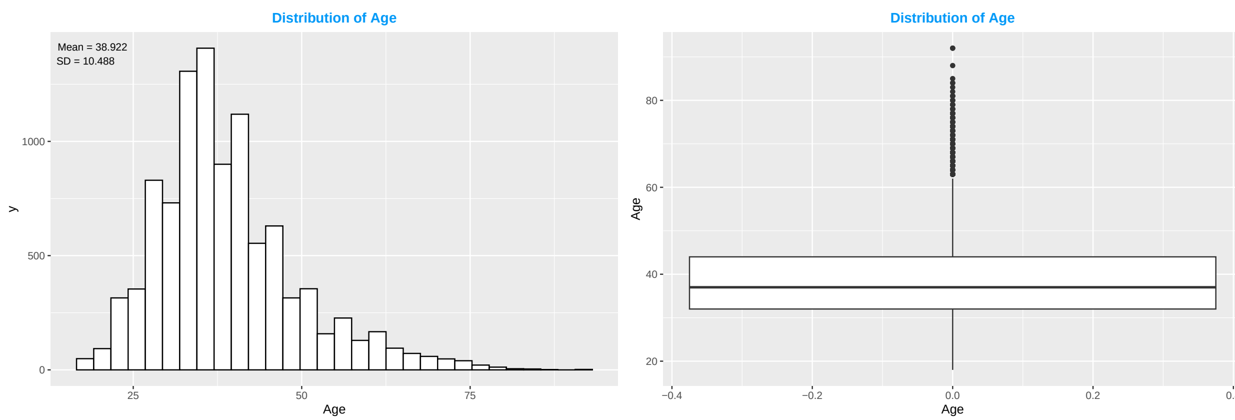
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The credit score seems to have a fairly normal distribution of values.

```
plot_numerical(bank_churn_data, "Age")
```

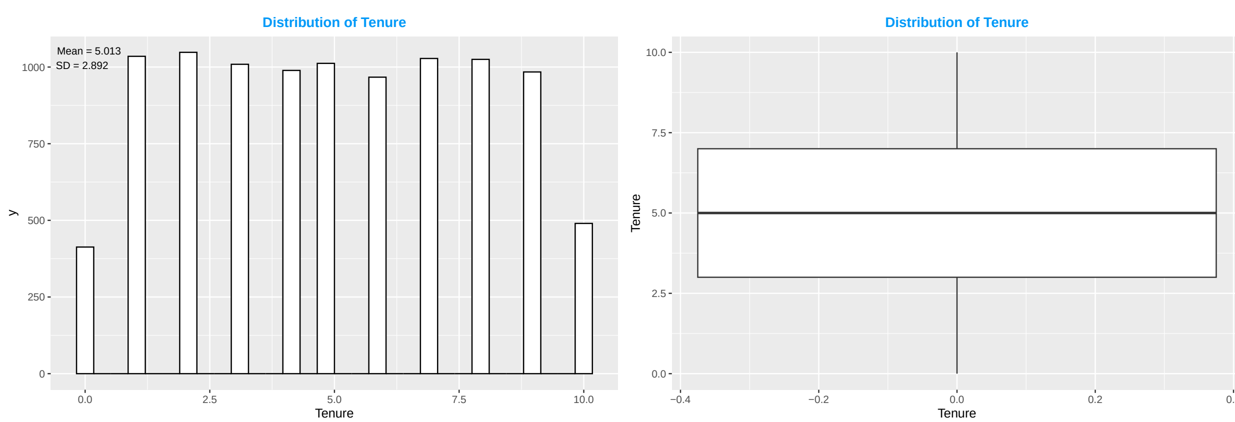
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The age also has a somewhat normal distribution, but with some irregularities.

```
plot_numerical(bank_churn_data, "Tenure")
```

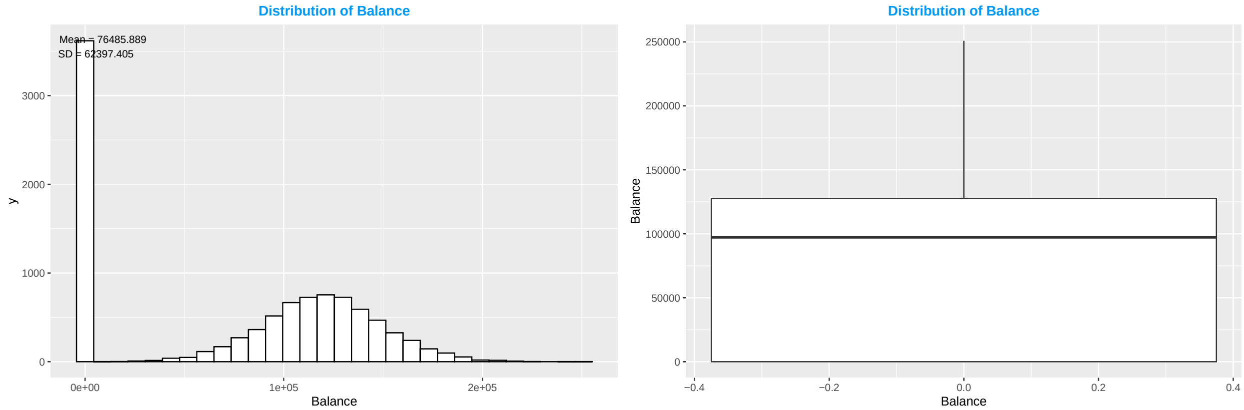
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



There appears to be no pattern to the tenure, with there being around 1000 records for each year.

```r
plot_numerical(bank_churn_data, "Balance")
```

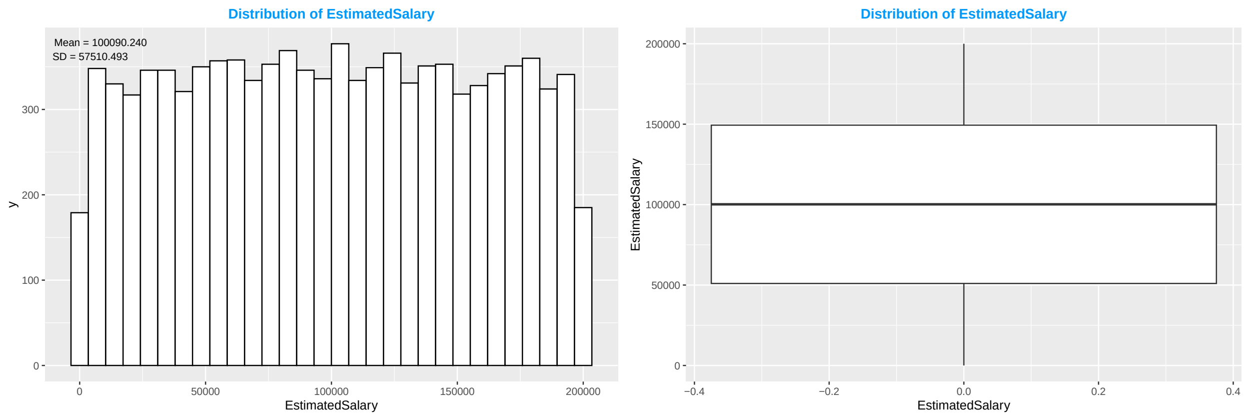
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The balance follows a normal distribution as well. However there is a peak at 0.

```r
plot_numerical(bank_churn_data, "EstimatedSalary")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



There appears to be no pattern to the Estimated Salary as well.

2.1.2 Visualizing categorical data

First, we should convert all categorical columns to the correct data type.

```r
bank_churn_data$Geography <- as.factor(bank_churn_data$Geography)
bank_churn_data$Gender <- as.factor(bank_churn_data$Gender)
bank_churn_data$NumOfProducts <- as.factor(bank_churn_data$NumOfProducts)
bank_churn_data$HasCrCard <- as.factor(bank_churn_data$HasCrCard)
bank_churn_data$IsActiveMember <- as.factor(bank_churn_data$IsActiveMember)
bank_churn_data$Exited <- as.factor(bank_churn_data$Exited)
```

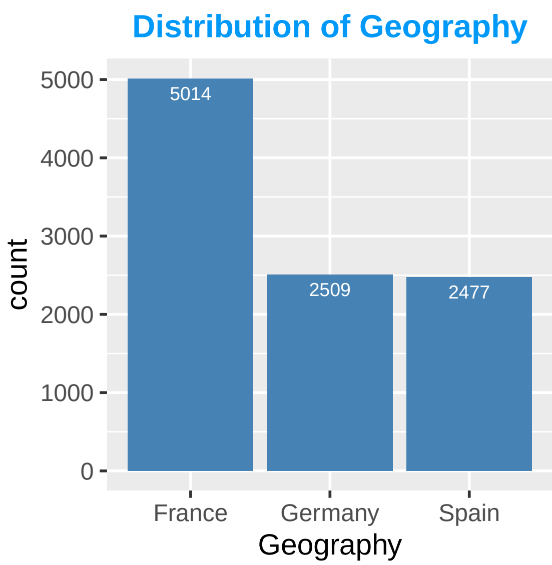
Now we can plot their distribution using bar charts.

```r
plot_bar_chart <- function(data, variable) {
  plot <- ggplot(data=data, aes(x=data[,variable])) +
    geom_bar(stat="count", fill="steelblue") +
    geom_text(stat="count", aes(label=..count..), vjust=1.6, color="white", size=2.5) +
```

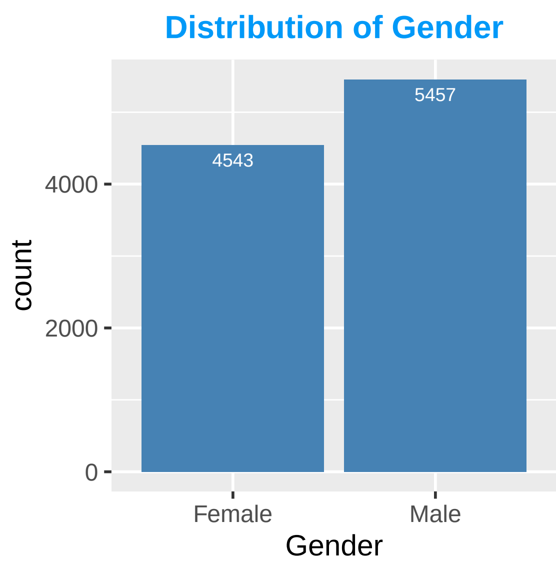
```r
    labs(
      title = glue('Distribution of {variable}'),
      x = variable,
      y = "count"
    ) +
    theme(
      plot.title = element_text(color = "#0099f8", size = 12, face = "bold", hjust = 0.5),
    )

  return (plot)
}
```

```r
plot_bar_chart(bank_churn_data, "Geography")
```

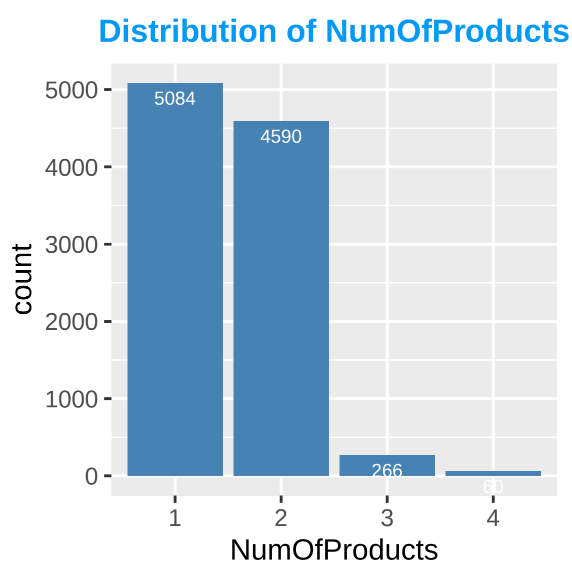
```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



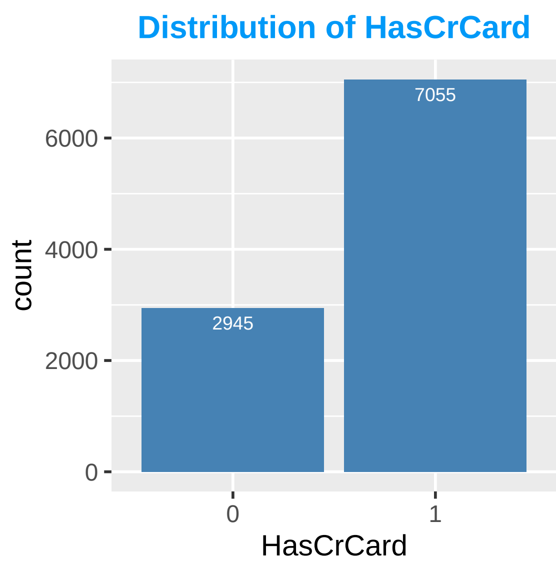

```r
plot_bar_chart(bank_churn_data, "Gender")
```



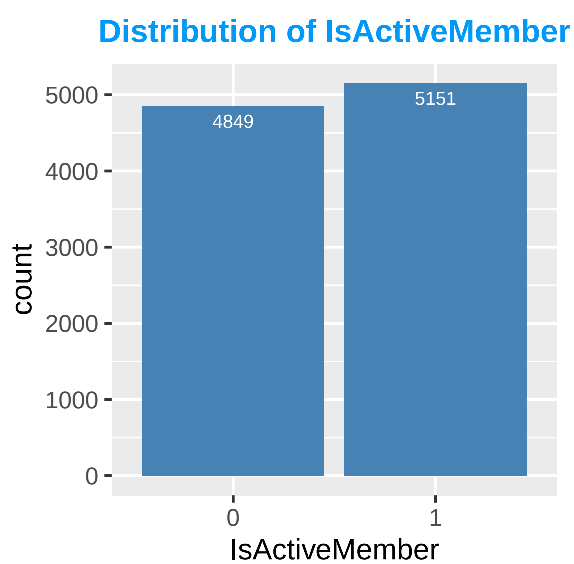

```
plot_bar_chart(bank_churn_data, "NumOfProducts")
```



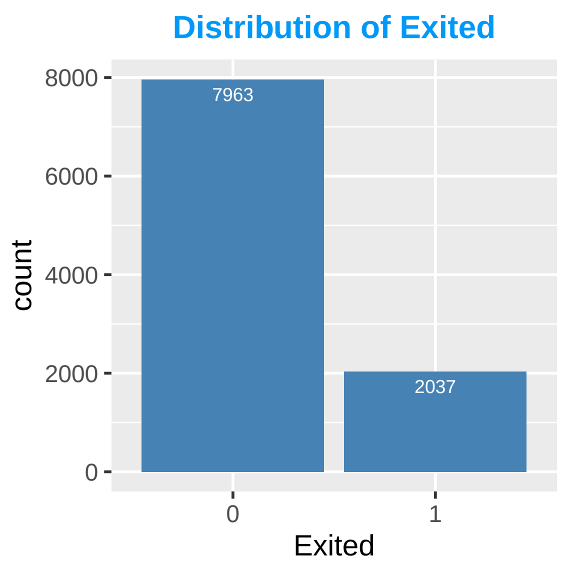

```
plot_bar_chart(bank_churn_data, "HasCrCard")
```




```
plot_bar_chart(bank_churn_data, "IsActiveMember")
```




```
plot_bar_chart(bank_churn_data, "Exited")
```



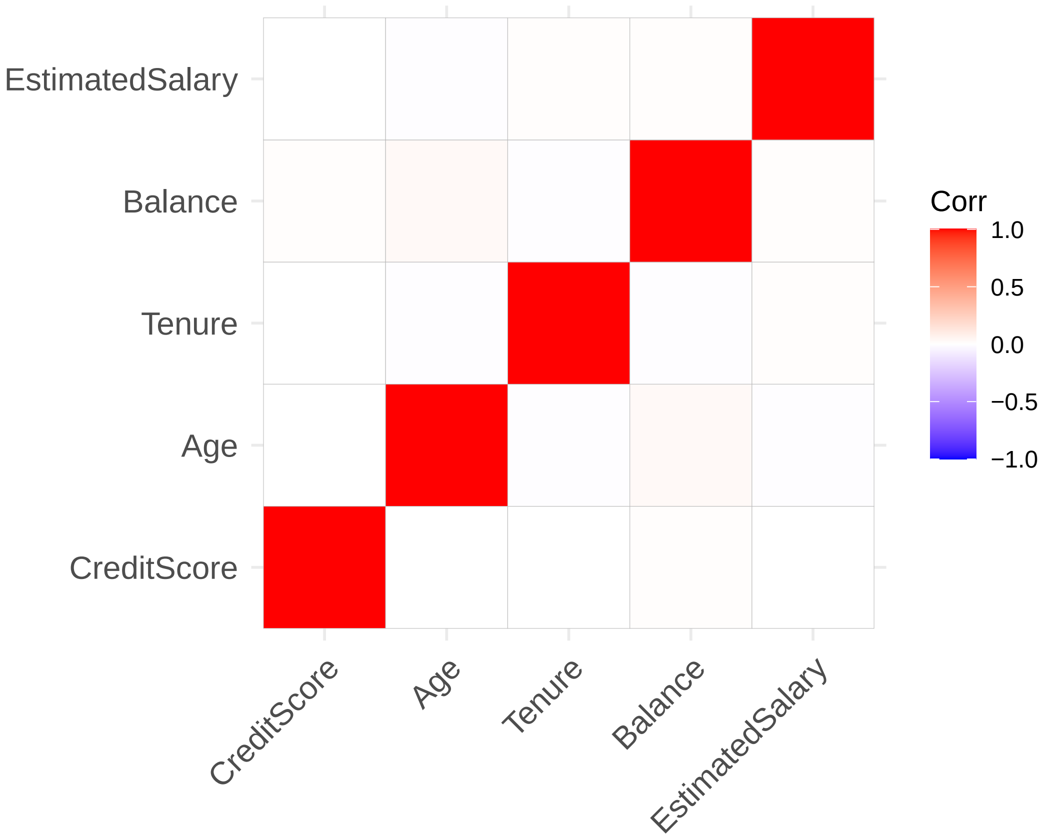
2.1.3 Identifying Numeric-Numeric Correlations

We can calculate the correlations between numeric variables.

```
numeric_only <- subset(bank_churn_data, select=c("CreditScore", "Age", "Tenure", "Balance", "EstimatedSa
correlations <- cor(numeric_only)
round(correlations, 3)
```

```
##                 CreditScore    Age Tenure Balance EstimatedSalary
## CreditScore           1.000 -0.004  0.001   0.006          -0.001
## Age                  -0.004  1.000 -0.010   0.028          -0.007
## Tenure                0.001 -0.010  1.000  -0.012           0.008
## Balance               0.006  0.028 -0.012   1.000           0.013
## EstimatedSalary      -0.001 -0.007  0.008   0.013           1.000
```

```
ggcorrplot(correlations)
```



As we can see, there does not seem to be any significant correlations between the numerical variables

2.1.4 Identifying Categorical-Categorical Correlations

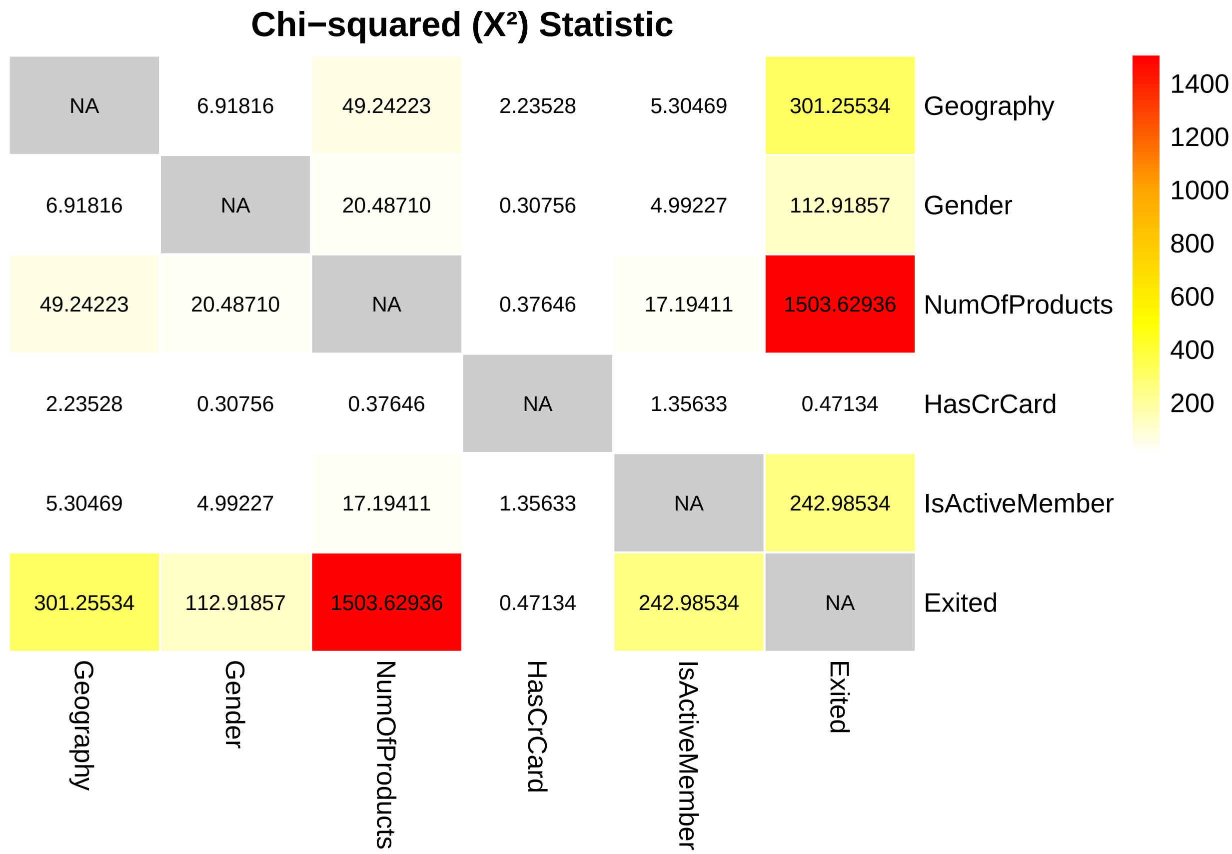
We can calculate the correlations between categorical variables using pairwise Chi-squared tests.

```
categorical_columns <- c("Geography", "Gender", "NumOfProducts", "HasCrCard", "IsActiveMember", "Exited

pairwise_p_vals <- matrix(nrow=6, ncol=6)
pairwise_chi_square <- matrix(nrow=6, ncol=6)
rownames(pairwise_p_vals) <- categorical_columns
colnames(pairwise_p_vals) <- categorical_columns
rownames(pairwise_chi_square) <- categorical_columns
colnames(pairwise_chi_square) <- categorical_columns

for (i in 1:5) {
  for (j in (i+1):6) {
    contingency_table <- table(bank_churn_data[,categorical_columns[i]], bank_churn_data[,categorical_c
    chi_square_test <- chisq.test(contingency_table)

    p_value <- chi_square_test$p.value
    total_chi_square <- chi_square_test$statistic

    pairwise_p_vals[[i, j]] <- p_value
    pairwise_p_vals[[j, i]] <- p_value

    pairwise_chi_square[[i, j]] <- total_chi_square
```

```
    pairwise_chi_square[[j, i]] <- total_chi_square
  }
}
```

The higher the Chi-squared value, the more significant the correlation.

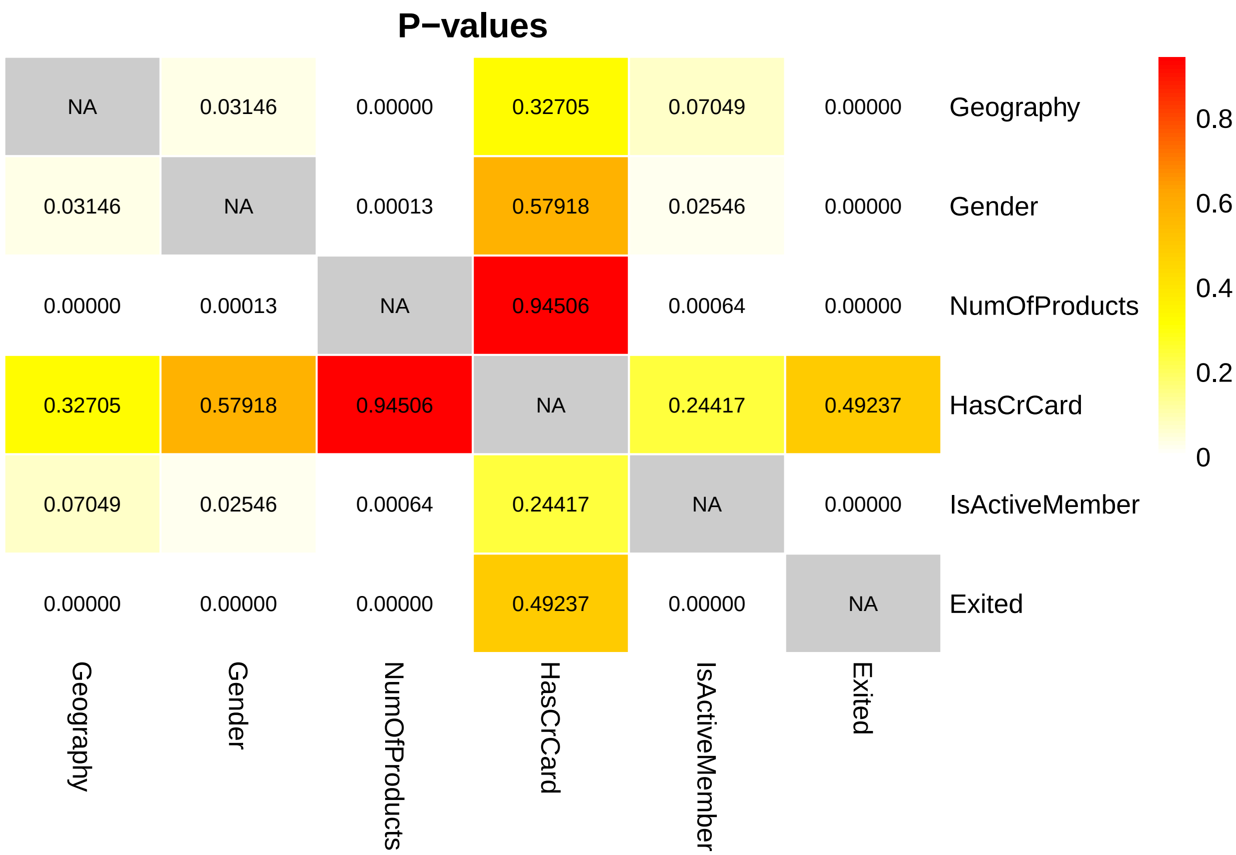
```
pheatmap(pairwise_chi_square,
         color = colorRampPalette(rev(c("red", "orange", "yellow", "white")))(100),
         display_numbers = TRUE,
         number_format = "%.5f",
         number_color = "black",
         na_col = "grey80",
         main = "Chi-squared (X²) Statistic",
         fontsize_number = 8,
         border_color = "white",
         cluster_rows = FALSE,
         cluster_cols = FALSE)
```



If the p-value is less than 0.05, we can consider it significant.

```
pheatmap(pairwise_p_vals,
         color = colorRampPalette(rev(c("red", "orange", "yellow", "white")))(100),
         display_numbers = TRUE,
         number_format = "%.5f",
         number_color = "black",
         na_col = "grey80",
         main = "P-values",
```

```
        fontsize_number = 8,
        border_color = "white",
        cluster_rows = FALSE,
        cluster_cols = FALSE)
```



2.1.5 Identifying Numerical-Categorical Correlations

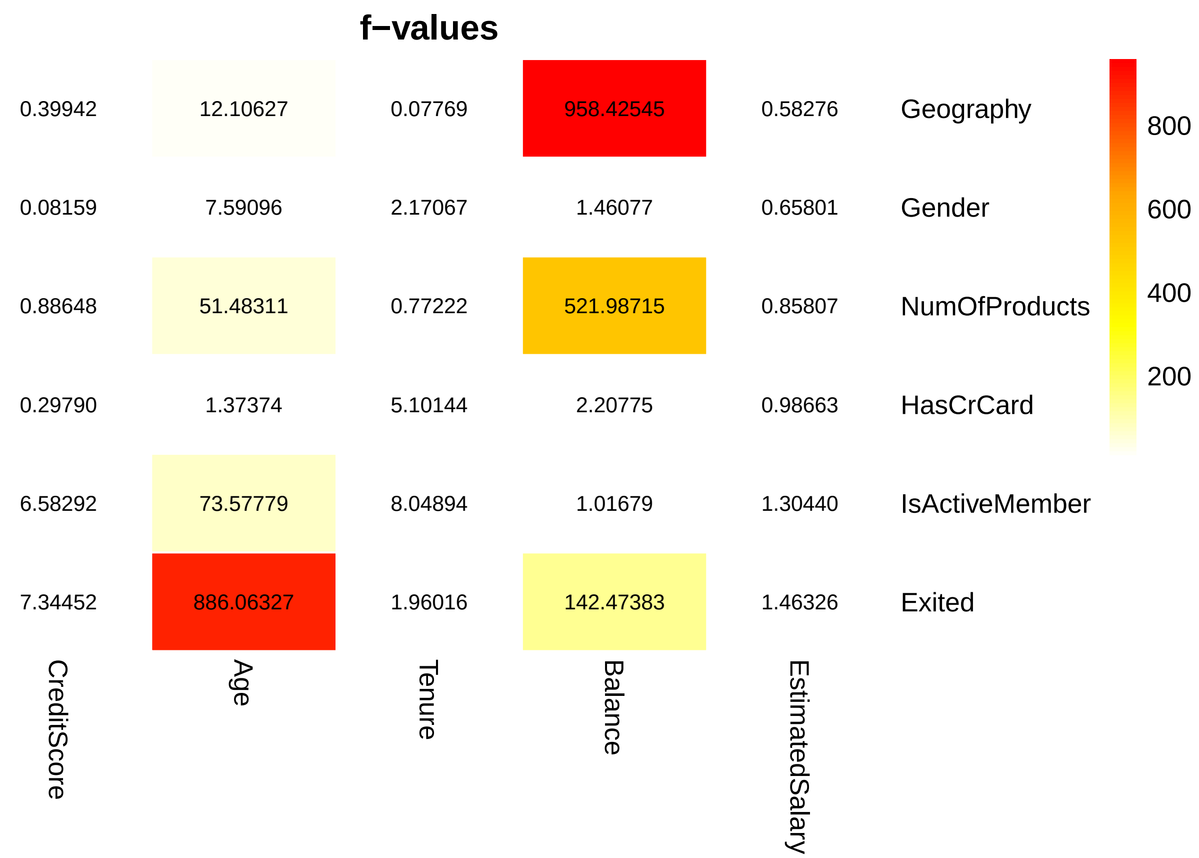
We can determine the correlation between numerical and categorical variables using pairwise ANOVA tests.

```
categorical_columns <- c("Geography", "Gender", "NumOfProducts", "HasCrCard", "IsActiveMember", "Exited
numeric_columns <- c("CreditScore", "Age", "Tenure", "Balance", "EstimatedSalary")

pairwise_p_vals <- matrix(nrow=6, ncol=5)
pairwise_f_vals <- matrix(nrow=6, ncol=5)
rownames(pairwise_p_vals) <- categorical_columns
colnames(pairwise_p_vals) <- numeric_columns
rownames(pairwise_f_vals) <- categorical_columns
colnames(pairwise_f_vals) <- numeric_columns


for (categorical_column in categorical_columns) {
  for (numeric_column in numeric_columns) {
    anova <- aov(reformulate(categorical_column, numeric_column), data=bank_churn_data)
    p_val <- summary(anova)[[1]][["Pr(>F)"]][1]
    f_val <- summary(anova)[[1]][["F value"]][1]
```

```
    pairwise_p_vals[[categorical_column, numeric_column]] <- p_val
    pairwise_f_vals[[categorical_column, numeric_column]] <- f_val
  }
}
```

The higher the f-value is, the more significant the relationship.

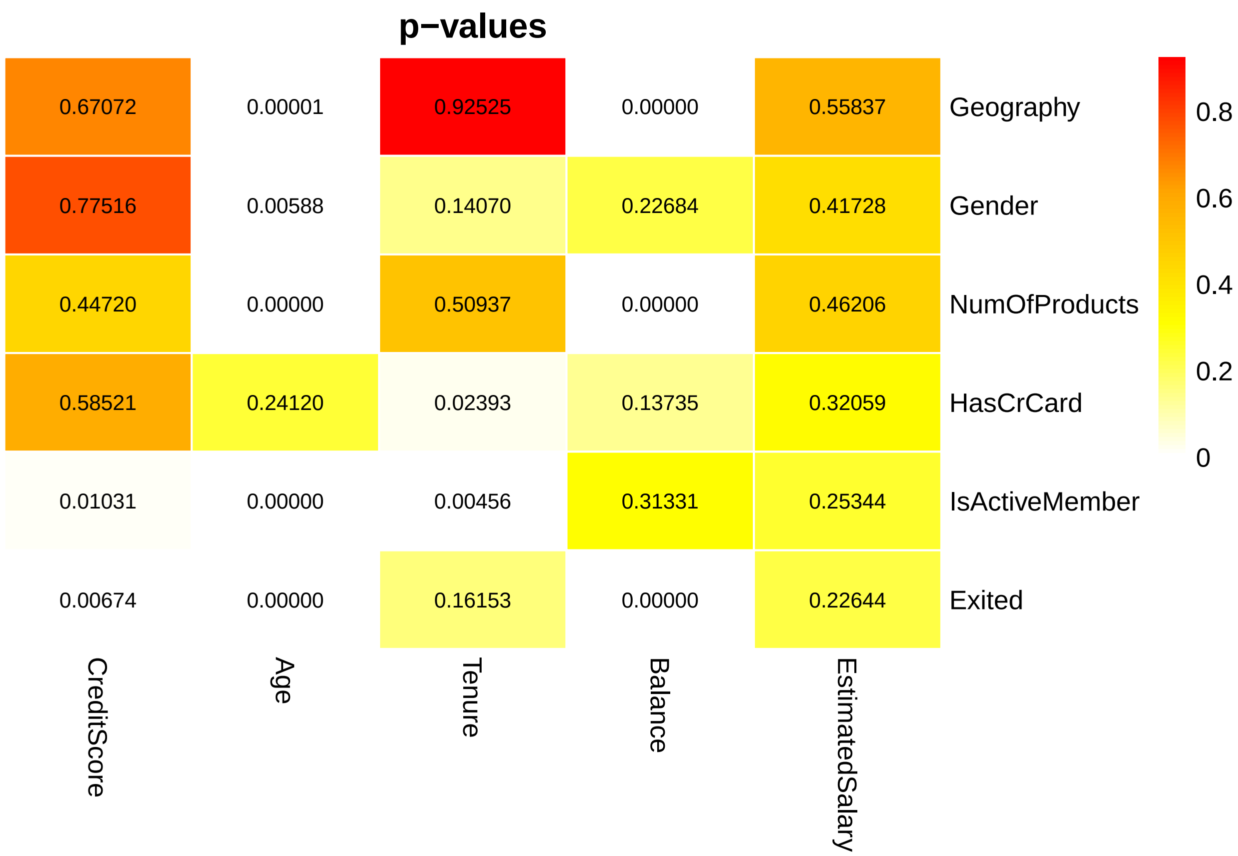
```
pheatmap(pairwise_f_vals,
         color = colorRampPalette(rev(c("red", "orange", "yellow", "white")))(100),
         display_numbers = TRUE,
         number_format = "%.5f",
         number_color = "black",
         na_col = "grey80",
         main = "f-values",
         fontsize_number = 8,
         border_color = "white",
         cluster_rows = FALSE,
         cluster_cols = FALSE)
```



If the p-value is less than 0.05, we can consider it significant

```
pheatmap(pairwise_p_vals,
         color = colorRampPalette(rev(c("red", "orange", "yellow", "white")))(100),
         display_numbers = TRUE,
         number_format = "%.5f",
         number_color = "black",
         na_col = "grey80",
```

```
         main = "p-values",
         fontsize_number = 8,
         border_color = "white",
         cluster_rows = FALSE,
         cluster_cols = FALSE)
```



2.1 Predictive Logistic Regression Model for Churn (Exited)

We first split the data into training and testing datasets at an 80/20 ratio.

```
split <- sample.split(bank_churn_data$Exited, SplitRatio = 0.8)
train_data <- subset(bank_churn_data, split == TRUE)
test_data <- subset(bank_churn_data, split == FALSE)
```

Next we develop the model. From our earlier analysis, we determined that the following variables have the most significant impact on `Exited`
- `Balance`
- `NumOfProducts`
- `Geography`
- `Gender`
- `Age`
- `IsActiveMember`

```
model <- glm(Exited ~ Balance + NumOfProducts + Geography + Gender + Age + IsActiveMember, data = train_
```

We can view the summary of the model.

```r
summary(model)
```

```
##
## Call:
## glm(formula = Exited ~ Balance + NumOfProducts + Geography +
##     Gender + Age + IsActiveMember, family = binomial(link = "logit"),
##     data = train_data)
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.366e+00  1.507e-01 -22.335  < 2e-16 ***
## Balance          -5.800e-07  6.391e-07  -0.908    0.364
## NumOfProducts2   -1.551e+00  8.002e-02 -19.383  < 2e-16 ***
## NumOfProducts3    2.597e+00  2.027e-01  12.809  < 2e-16 ***
## NumOfProducts4    1.619e+01  2.116e+02   0.077    0.939
## GeographyGermany  9.069e-01  8.081e-02  11.223  < 2e-16 ***
## GeographySpain    2.541e-02  8.556e-02   0.297    0.766
## GenderMale       -5.209e-01  6.596e-02  -7.896 2.88e-15 ***
## Age               7.235e-02  3.097e-03  23.361  < 2e-16 ***
## IsActiveMember1  -1.123e+00  6.952e-02 -16.155  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8088.9  on 7999  degrees of freedom
## Residual deviance: 5941.3  on 7990  degrees of freedom
## AIC: 5961.3
##
## Number of Fisher Scoring iterations: 14
```

2.2 Getting predictions for test dataset

We use the threshold value as 0.5, and make a set of prediction on our test dataset

```r
test_probabilities <- predict(model, newdata = test_data, type = "response")
predicted_exited <- ifelse(test_probabilities > 0.5, 1, 0)
```

We can compare the predicted values against our actual values and build the confusion matrix.

```r
actual_exited <- test_data$Exited
conf_matrix <- table(Actual = actual_exited, Predicted = predicted_exited)
print(conf_matrix)
```

```
##       Predicted
## Actual    0    1
##      0 1522   71
##      1  250  157
```

Here, we see that our model seems to have an issue with misclassificaion of false values (high number of false negatives). And we can plot the performance metrics to analyze the performance of our model.

```r
TP <- conf_matrix[2, 2]
TN <- conf_matrix[1, 1]
FP <- conf_matrix[1, 2]
FN <- conf_matrix[2, 1]
```

```r
accuracy <- (TP + TN) / sum(conf_matrix)
precision <- TP / (TP + FP)
sensitivity <- TP / (TP + FN)
specificity <- TN / (TN + FP)
f1_score <- 2 * (precision * sensitivity) / (precision + sensitivity)

print(glue("Accuracy: {accuracy}"))
```

```
## Accuracy: 0.8395
```

```r
print(glue("Precision: {precision}"))
```

```
## Precision: 0.68859649122807
```

```r
print(glue("Sensitivity: {sensitivity}"))
```

```
## Sensitivity: 0.385749385749386
```

```r
print(glue("Specificity: {specificity}"))
```

```
## Specificity: 0.955430006277464
```

```r
print(glue("F1 score: {f1_score}"))
```

```
## F1 score: 0.494488188976378
```

Even though the accuracy of the model may be high, it still does not seem to perform that well when the Churn is false (as indicated by the slow Sensitivity). We can also plot the ROC curve and calculate the area under it.

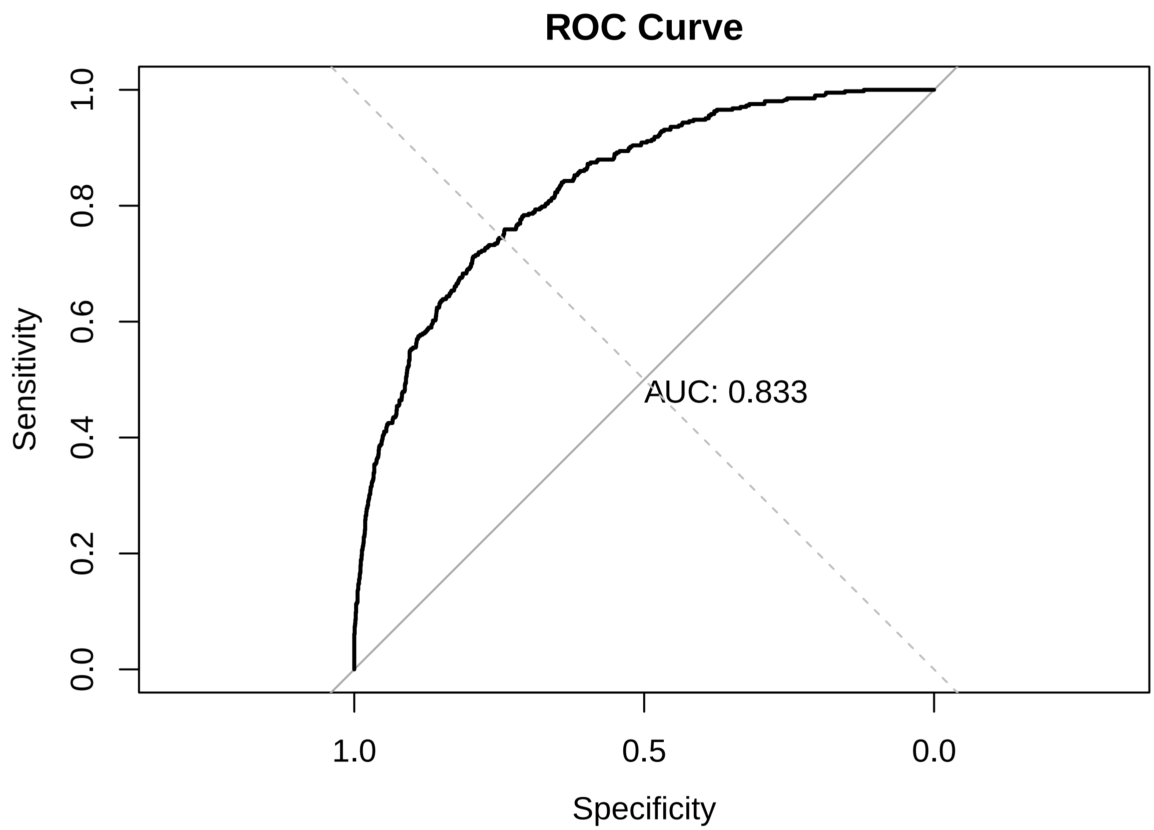
```r
roc_curve <- roc(response = actual_exited, predictor = test_probabilities)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```r
auc_value <- auc(roc_curve)
print(glue("AUC: {round(auc_value, 4)}"))
```

```
## AUC: 0.8326
```

```r
plot(roc_curve, main = "ROC Curve", print.auc = TRUE)
abline(a=0, b=1, lty=2, col="gray")
```



2.4 Predicting Tenure

From our earlier analysis, we can see that there are few to no variables which show significant correlation with `Tenure`. Hence we will use all variables in the dataset for our model.

```
tenure_model = lm(Tenure ~ CreditScore + Geography + Gender + Age + Balance + NumOfProducts + HasCrCard
summary(tenure_model)
```

```
##
## Call:
## lm(formula = Tenure ~ CreditScore + Geography + Gender + Age +
##     Balance + NumOfProducts + HasCrCard + +IsActiveMember + EstimatedSalary,
##     data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3189 -2.2364 -0.0358  2.7320  5.3720
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       4.714e+00  2.744e-01  17.180   <2e-16 ***
## CreditScore       2.854e-04  3.334e-04   0.856   0.3920
## GeographyGermany  5.973e-02  8.595e-02   0.695   0.4871
## GeographySpain    2.737e-02  7.935e-02   0.345   0.7302
## GenderMale        1.081e-01  6.499e-02   1.664   0.0962 .
## Age              -7.153e-04  3.137e-03  -0.228   0.8197
## Balance          -2.404e-07  6.122e-07  -0.393   0.6945
```

```
## NumOfProducts2    2.874e-02  7.169e-02   0.401   0.6885
## NumOfProducts3   -9.737e-02  2.035e-01  -0.479   0.6322
## NumOfProducts4    4.243e-01  4.487e-01   0.946   0.3443
## HasCrCard1        1.566e-01  7.064e-02   2.217   0.0266 *
## IsActiveMember1  -1.300e-01  6.492e-02  -2.003   0.0452 *
## EstimatedSalary   4.727e-07  5.639e-07   0.838   0.4019
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.887 on 7987 degrees of freedom
## Multiple R-squared:  0.001888,   Adjusted R-squared:  0.0003886
## F-statistic: 1.259 on 12 and 7987 DF,  p-value: 0.2358
```

After training the model on our training dataset, we can evaluate it against our test dataset.

```
test_predictions <- predict(tenure_model, newdata = test_data)
actual_tenure = test_data$Tenure

rmse <- sqrt(mean((actual_tenure - test_predictions)^2))
print(glue("RMSE = {rmse}"))
```

```
## RMSE = 2.90873855815441
```

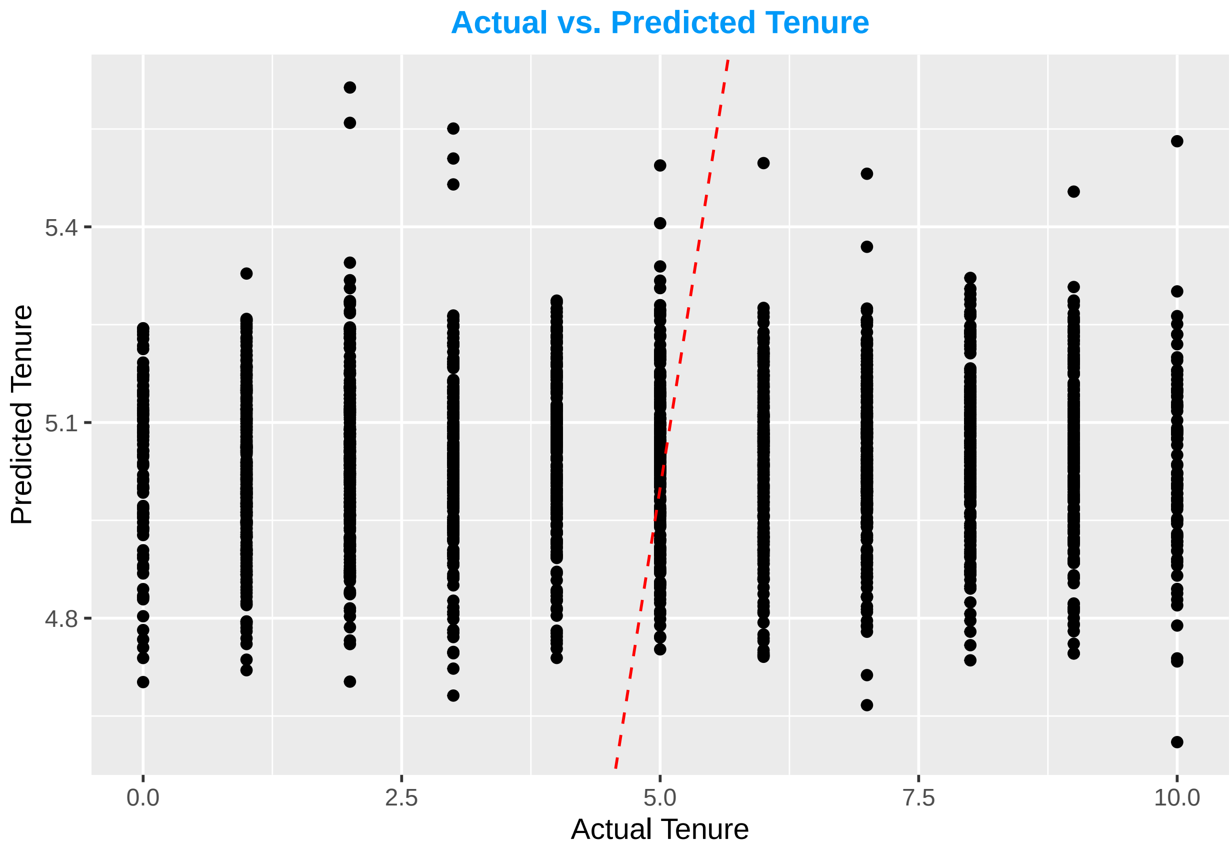
The root mean squared error we obtain is 2.9, which is not great considering that this would cover over 50% of the values in `Tenure`.

```
rss <- sum((test_predictions - actual_tenure)^2)
tss <- sum((actual_tenure - mean(actual_tenure))^2)
rsq_test <- 1 - (rss / tss)
print(glue("R² = {round(rsq_test, 4)}"))
```

```
## R² = -9e-04
```

The value we obtain for R^2 is also very low. This indicates that this model does not perform well.

```
ggplot(data = test_data, aes(x = Tenure, y = test_predictions)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
  labs(title = "Actual vs. Predicted Tenure",
       x = "Actual Tenure",
       y = "Predicted Tenure") +
  theme(
    plot.title = element_text(color = "#0099f8", size = 12, face = "bold", hjust = 0.5),
  )
```



We can also plot our predicted values against the actual values. Here we can see that our model is mostly predicting values between 4.6 and 5.4. But the complete range of values fall between 0 and 10. We can infer from this, that our model does not have sufficient data to make accurate predictions.