Retail Regression Project

Nicholas Pinero

2025-03-04

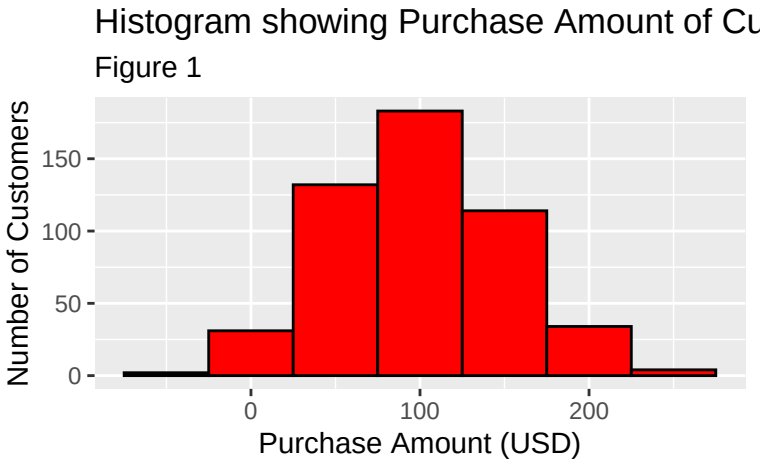
1. Introduction

In this project, I analyze the "Retail & E-Commerce" Dataset to investigate the relationship between return likelihood and predictor variables such as purchase amount, discount status, and loyalty status. This analysis includes exploratory data analysis (EDA), simple and multiple linear regression, and bootstrapping techniques to construct confidence and prediction intervals.

2. Exploratory Data Analysis (EDA)

The Retail and E-commerce Dataset contains 500 observations. Key Summary Statistics:

- Purchase_Amount: Mean = 99.6, Median = 99.1, Q1 = 65, Q3 = 131.5, SD = 49.4
- Return_Likelihood: Mean = 7.68, Median = 6.86, Q1 = 2.36, Q3 = 12.14, SD = 6.08

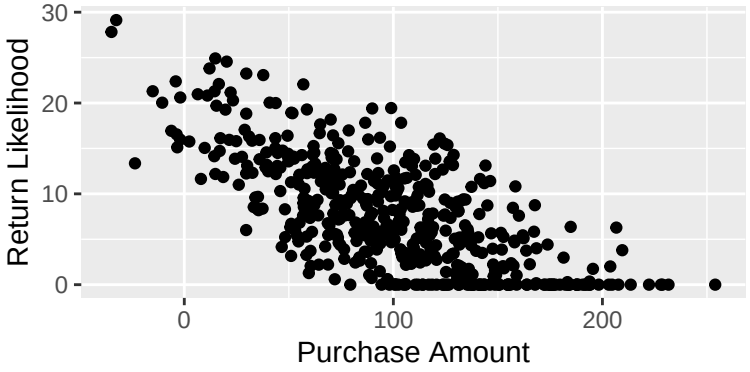


Initial Observations:

1. Higher Purchase amounts appear to correlate with lower return likelihood, referencing figure 3.
2. Discounted products have a higher return likelihood than non-discounted products, referencing figure 4.
3. VIP customers exhibit the lowest return likelihood compared to Regular customer who, in turn, exhibit a lower return likelihood than New customer, referencing figure 5.
4. Potential concerns include the right-skewed return likelihood histogram in figure 2, outliers in figures 5 and 4, and negative purchase amount values in figure 3.

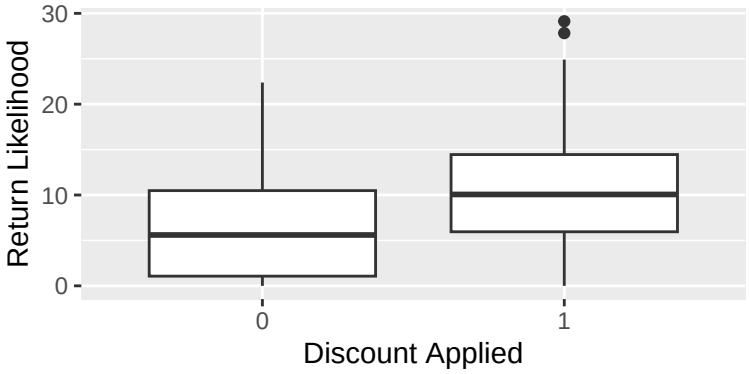
Purchase Amount vs Return Likelihood

Figure 3



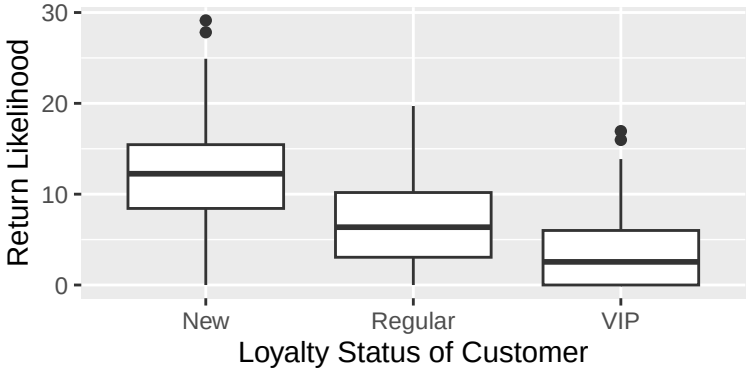
Return Likelihood with Discount vs No Disco

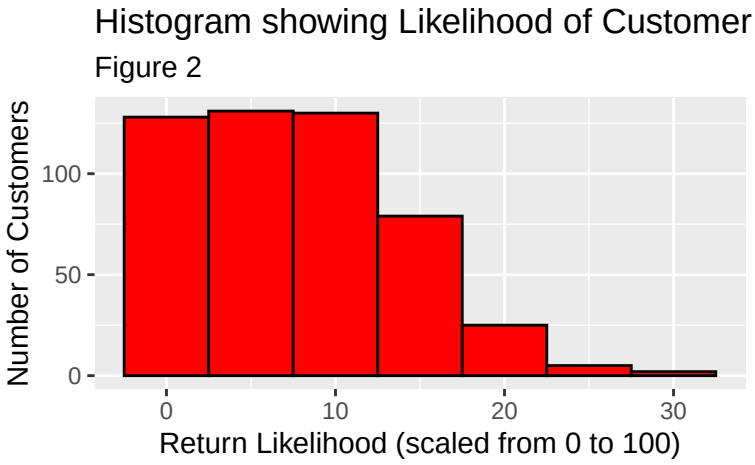
Figure 4



Return Likelihood based on Customer Loyalt

Figure 5





3. Simple Linear Regression

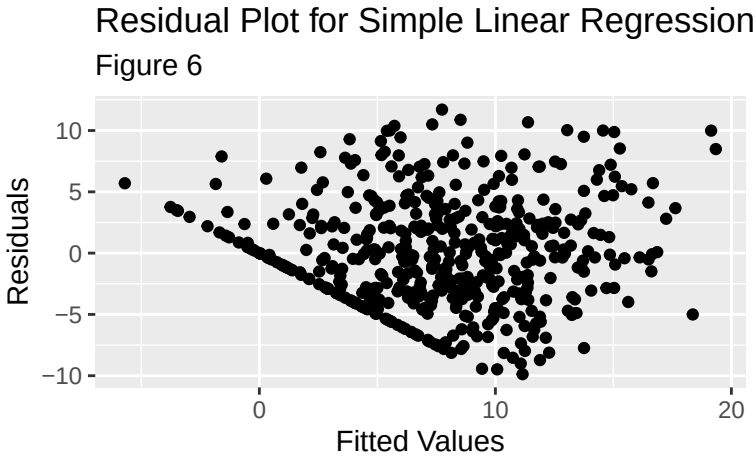
Return_Likelihood = 16.319455 - 0.086711(Purchase_Amount) + error

The intercept of 16.319455 indicates that when the Purchase Amount is zero, the predicted Return Likelihood is 16.319455, though this value may be theoretical if a purchase amount of zero is not meaningful in the dataset. The estimated slope of -0.086711 suggests that for each additional unit increase in Purchase Amount, the predicted Return Likelihood decreases by 0.086711. This negative relationship implies that higher purchase amounts are associated with lower return likelihoods, as reflected in the regression line.

R-Squared: 0.4953034(49.53035% of variation in Return_Likelihood can be explained by Purchase_Amount).

One major concern regarding model validity is the residual plot (Figure 6), which shows a non-random pattern, suggesting a potential violation of homoscedasticity. A noticeable horizontal line of residuals on the left side indicates the model may not fully capture certain aspects of the data, possibly due to the right-skewed distribution of the response variable or the presence of outliers.

Additionally, the histogram of residuals (figure 7) appears slightly right-skewed rather than perfectly bell-shaped, raising concerns about the normality assumption of residuals.



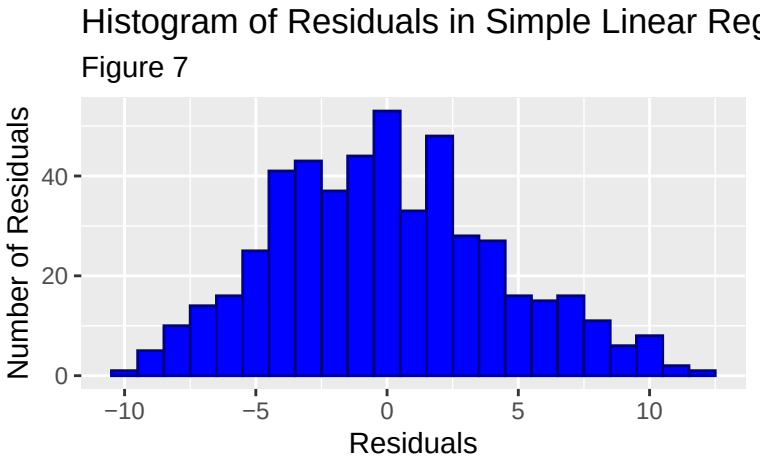


Figure 7

4. Bootstrapping for Inference

Using 5,000 bootstrap samples, the 95% confidence interval for estimated slope between Purchase_Amount and Return_Likelihood is between -0.0942 and -0.0794. This means I can be 95% confident that for every dollar increase in purchase amount, the likelihood of the product being returned decreases by between 0.0794 and 0.0942.

For a purchase amount of $200, the built-in predict function in R estimated a Return_Likelihood of -1.02, which is impossible since likelihoods cannot be negative. Similarly, the 95% prediction interval from bootstrapping suggests a range of -1.790 to -0.292, which also includes unrealistic values. Since likelihoods represent probabilities and must be non-negative, these results indicate potential issues with the model, such as extrapolation beyond the data range or an inappropriate functional form.

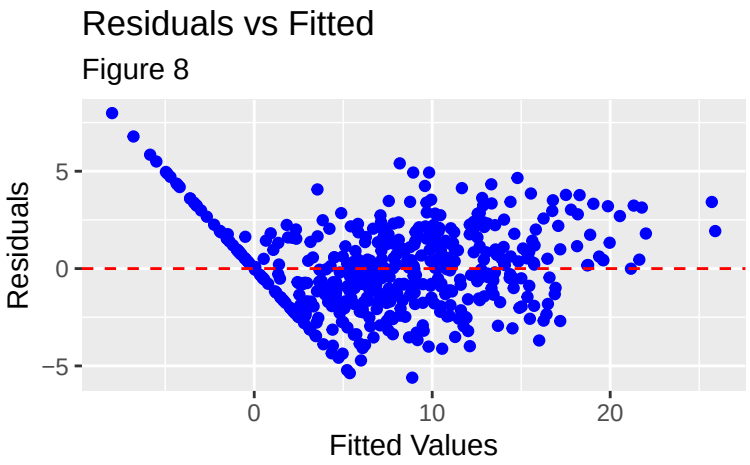
5. Multiple Linear Regression

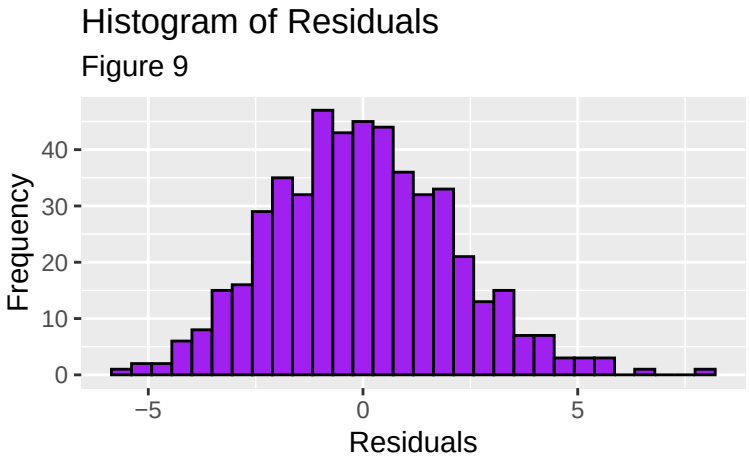
Return_Likelihood = 18.71655 - 4.98542(Regular Loyalty Status) - 7.77206(VIP Loyalty Status) + 4.29187(Discount Applied) - 0.08303(Purchase Amount)

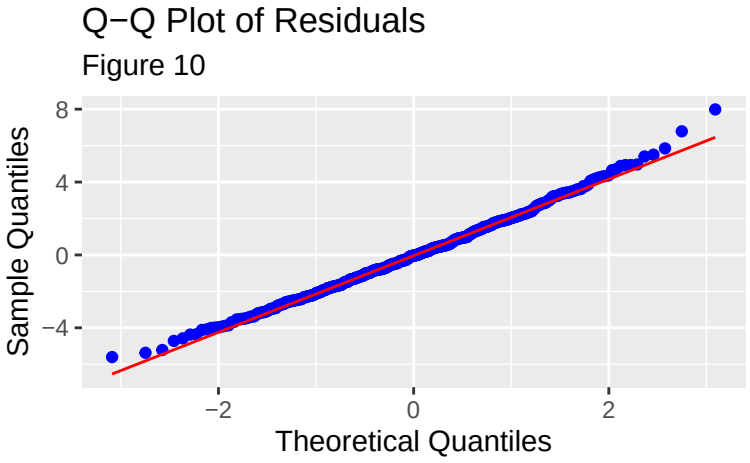
The intercept (18.72) represents the return likelihood when a customer spends $0, has no discount applied, and has New loyalty status. For each additional dollar spent, the return likelihood decreases by 0.083 units, while applying a discount increases it by 4.29 units. Compared to New customers, Regular and VIP customers have 4.99 and 7.77 units lower return likelihoods, respectively.

The multiple regression model explains 88% of the variance in return likelihood (Adjusted R-squared = 0.88), a significant improvement over the simple regression model, which had an R-squared of 0.494. This indicates that the additional predictors (discount status and loyalty status) provide valuable information for predicting return likelihood. However, the model has limitations. For example, the prediction for a new observation (Purchase_Amount = $200, Discount_Applied = No, Loyalty_Status = Regular) is -2.87, which is unrealistic since return likelihood cannot be negative.

Diagnostic plots (Figures 8, 9, and 10) reveal violations of model assumptions. The residuals vs. fitted plot (Figure 8) shows a non-random pattern, with a straight horizontal line on the left side, suggesting the model struggles to capture relationships at lower fitted values and violates the assumption of homoscedasticity. The histogram of residuals (Figure 9) is slightly right-skewed, indicating the model may underestimate higher return likelihoods, which could explain the unrealistic negative predictions. The Q-Q plot (Figure 10) shows most points close to the reference line but with slight departures at the tails, particularly on the right, further supporting the non-normality of residuals.







6. Conclusion

The analysis reveals that purchase amount, discount status, and loyalty status are significant predictors of return likelihood. Higher purchase amounts and VIP loyalty status reduce return likelihood, while discounts

increase it. The multiple regression model outperforms the simple regression model, explaining 88% of the variance in return likelihood (Adjusted R-squared = 0.88). However, the model has limitations, including unrealistic negative predictions and violations of regression assumptions due to the right-skewed distribution of return likelihood. Diagnostic plots indicate issues with homoscedasticity and normality of residuals, suggesting the model may not fully capture the underlying relationships in the data.

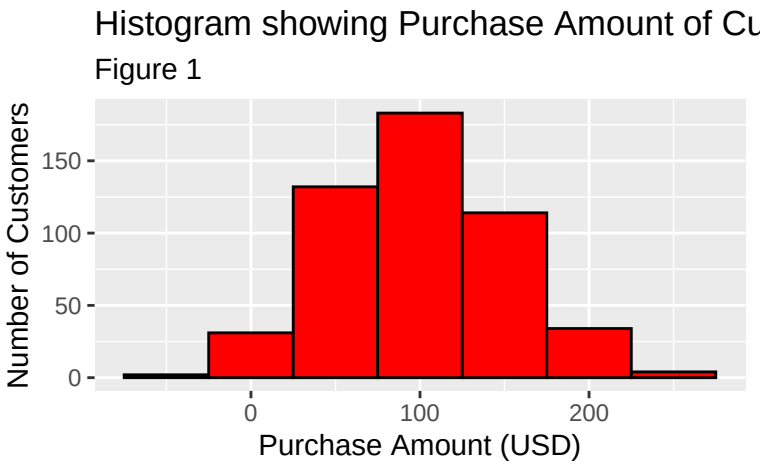
To address these limitations, future work could explore transformations (possibly log transformation) to reduce skewness or alternative models that better handle non-normal distributions. Additionally, investigating the impact of other variables, such as product category or customer demographics, could provide further insights into return likelihood and improve the prediction accuracy.

Appendix

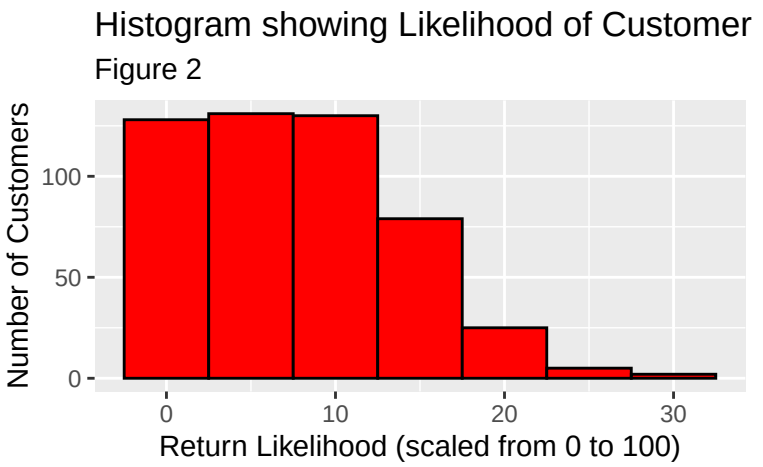
```
# Loading Libraries
library(tidyverse)

# Data Import
Retail <- read_csv("Retail_Ecommerce_Dataset.csv")
```

```
Rows: 500 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): Loyalty_Status, Product_Category
dbl (4): Customer_Age, Purchase_Amount, Discount_Applied, Return_Likelihood

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

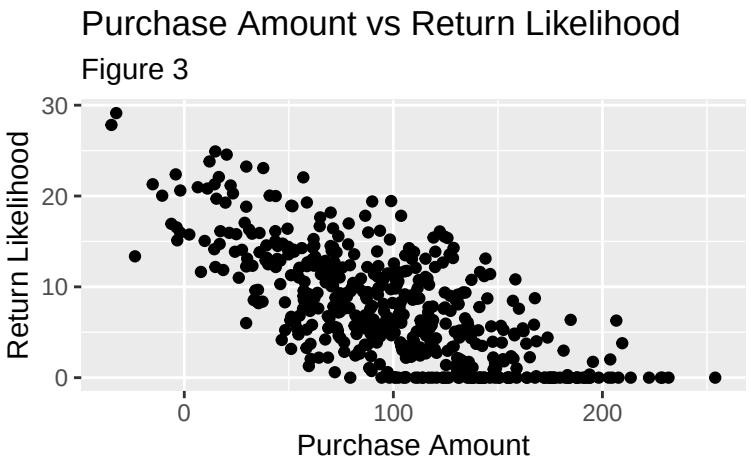
```
# Number of observations in the data set
Observations <- nrow(Retail)

# Summary Statistics
Summmary_Retail <- summary(Retail)

# Standard Deviations of Relevent Numerical Variables
SD_Customer_Age <- sd(Retail$Customer_Age)
SD_Purchase_Amount <- sd(Retail$Purchase_Amount)
SD_Return_Likelihood <- sd(Retail$Return_Likelihood)

# Figure 1
# Histogram showing Purchase Amount of Customers
ggplot(Retail, aes(x = Purchase_Amount)) +
  geom_histogram(binwidth = 50, fill = "red", color = "black") +
  labs(x = "Purchase Amount (USD)", y = "Number of Customers",
       subtitle = "Figure 1") +
  ggtitle("Histogram showing Purchase Amount of Customers")
```



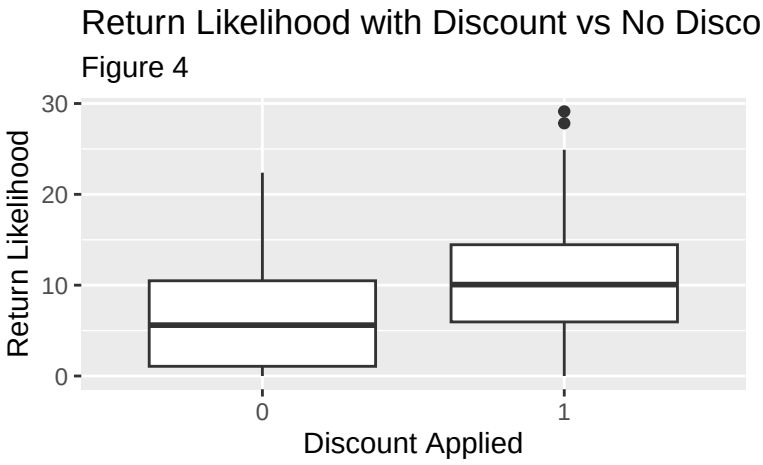

```
# Figure 2
# Histogram showing Likelihood of a purchased product being returned
ggplot(Retail, aes(x = Return_Likelihood)) +
  geom_histogram(binwidth = 5, fill = "red", color = "black") +
  labs(x = "Return Likelihood (scaled from 0 to 100)",
       y = "Number of Customers",
       subtitle = "Figure 2") +
  ggtitle("Histogram showing Likelihood of Customer Returning a Purchased Product")
```



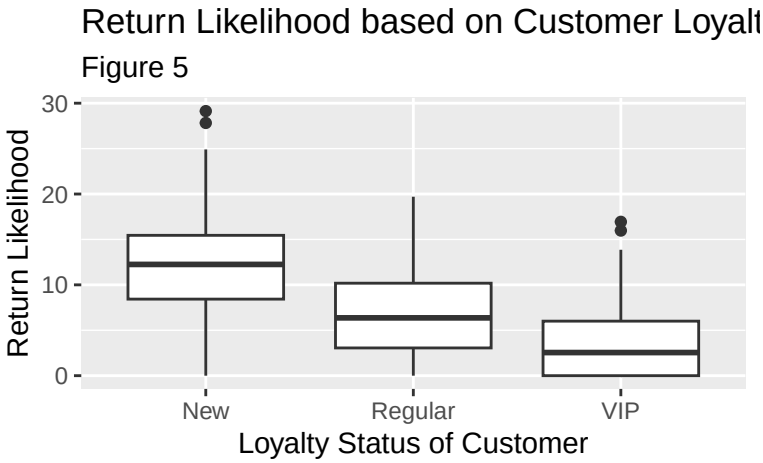

```
# Figure 3
# Scatter plot showing relationship between purchase
# amount and likelihood of product being returned.
ggplot(Retail, aes(x = Purchase_Amount, y = Return_Likelihood)) +
  geom_point() +
  labs(x = "Purchase Amount", y = "Return Likelihood",
       subtitle = "Figure 3") +
  ggtitle("Purchase Amount vs Return Likelihood")
```



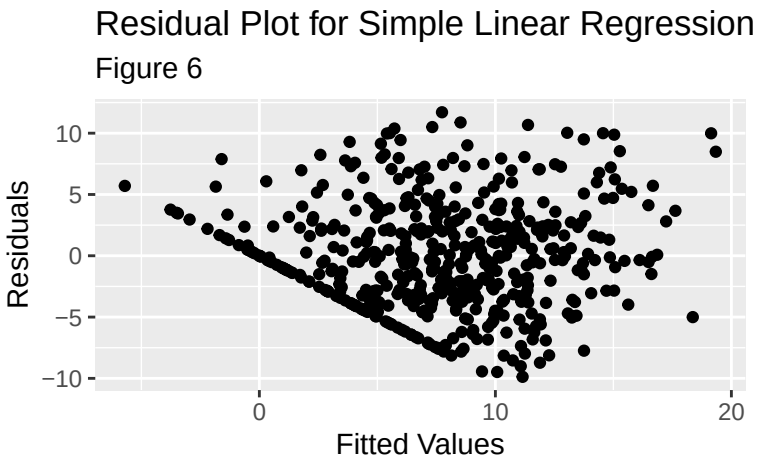

```
# Figure 4
# Box plot showing the the distribution of the
# relationship between products with a discount applied
# and the likelihood of said product being returned.
ggplot(Retail, aes(x = factor(Discount_Applied), y = Return_Likelihood)) +
  geom_boxplot() +
  labs(x = "Discount Applied", y = "Return Likelihood",
       subtitle = "Figure 4") +
  ggtitle("Return Likelihood with Discount vs No Disco")
```



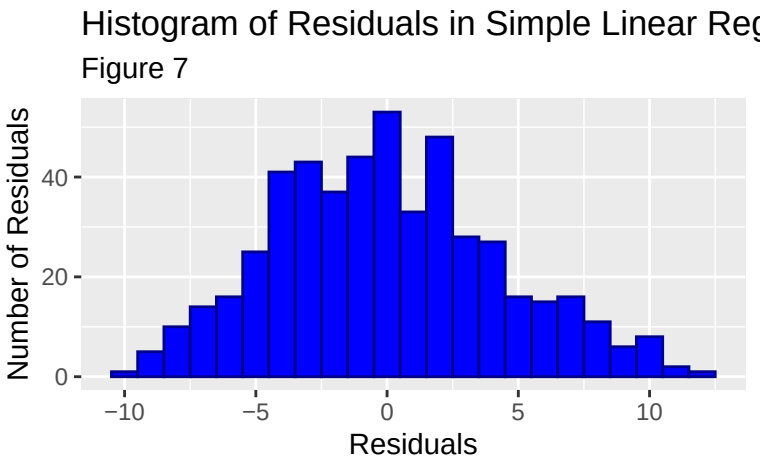

```
# Figure 5
# Box plot showing the distribution of the relationship
# between a customer's loyalty status and the likelihood
# of the purchased product being returned.
ggplot(Retail, aes(x = Loyalty_Status, y = Return_Likelihood)) +
  geom_boxplot() +
  labs(x = "Loyalty Status of Customer", y = "Return Likelihood",
       subtitle = "Figure 5") +
  ggtitle("Return Likelihood based on Customer Loyalty Status")
```




```r
# Fitted Simple Linear Regression Model for response variable of
# Return Likelihood and predictor variable of Purchase Amount.
model <- lm(Return_Likelihood ~ Purchase_Amount, data = Retail)
model_summary = summary(model)

# R-squared value for my simple linear regression model.
R_Squared <- model_summary$r.squared

# Creates data frame for fitted values and residuals in the simple
# linear regression model created above.
model_df <- data.frame(Fitted_Values = model$fitted.values,
                       Residuals = model$residuals)

# Figure 6
# Residual Plot for my simple linear regression model
ggplot(model_df, aes(x = Fitted_Values, y = Residuals)) +
  geom_point() +
  labs(x = "Fitted Values", y = "Residuals",
       subtitle = "Figure 6") +
  ggtitle("Residual Plot for Simple Linear Regression Model")
```



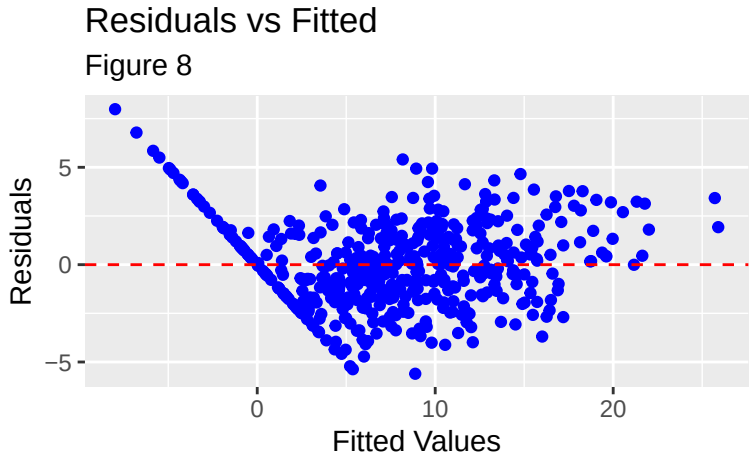
```r
# Residual values in simple linear regression model
residuals <- model_summary$residuals
residuals_df <- data.frame(residuals)

# Figure 7
# Histogram of residuals in simple linear regression
ggplot(residuals_df, aes(x = residuals)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "darkblue") +
  labs(x = "Residuals", y = "Number of Residuals",
       subtitle = "Figure 7") +
  ggtitle("Histogram of Residuals in Simple Linear Regression Model")
```



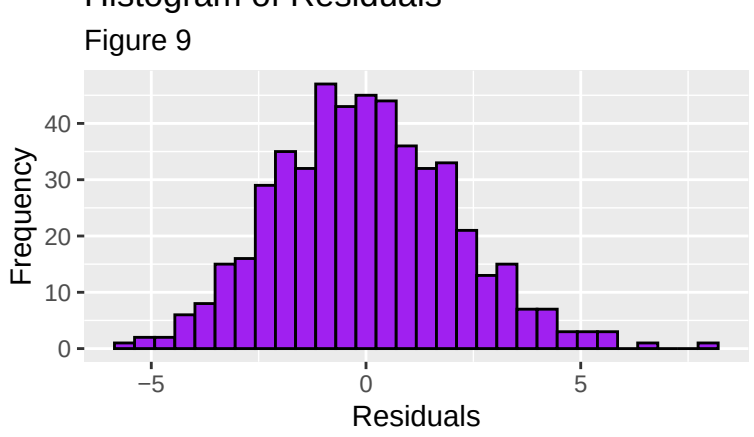

```r
# Defines a function to perform bootstrapping for the slope of a
# linear regression model.
bootstrap_slope <- function(df, x, y, repetitions = 5000) {
  # Number of rows in the dataset
  n = nrow(df)
  # Helper function to calculate slope of a linear regression model
  slope <- function(x, y) {
    model <- lm(y ~ x)
    return(coef(model)[2])
  }

  slopes <- sapply(1:repetitions, function(i) {
    indices = sample(1:n, size = n, replace = T)
    sample_df = df[indices,]
    slope(x = sample_df[[x]], y = sample_df[[y]])
    })
# Find 2.5th and 97.5th percentiles for the 95% CI
ci <- quantile(slopes, probs = c(0.025, 0.975))
# Compute the slope on the original data
observed_slope <- slope(df[[x]], df[[y]])

list(observed_slope = observed_slope, ci = ci, all_slopes = slopes)
}

boots_slope <- bootstrap_slope(Retail, "Purchase_Amount", "Return_Likelihood")
```

```r
# Grabs the 95% confidence interval calculated from the above
# boostrap_slope function
Bootstrap_CI <- boots_slope$ci

# Prediction at Purchase_Amount = 200
prediction <- predict(model, data.frame(Purchase_Amount = 200))

# Defines a function to perform bootstrapping for predictions
bootstrap_predictions <- function(data, x_col, y_col, new_x, repetitions = 5000, seed = 123) {
  # Sets seed for reproducibility
  set.seed(seed)

  # Generates bootstrap predictions
  predictions <- replicate(repetitions, {
    # Resample the data with replacement
    resample <- data %>% sample_n(nrow(data), replace = TRUE)

    # Fit the linear regression model on the bootstrap sample
    boot_model <- lm(as.formula(paste(y_col, "~", x_col)), data = resample)

    # Creates a new data frame for the prediction
    new_data <- data.frame(new_x)
    names(new_data) <- x_col

    # Predict the Return Likelihood on the new data point
    predict(boot_model, new_data)
  })

  return(predictions)
}

# Run the bootstrap function for a new Purchase_Amount = 200
boots_predictions <- bootstrap_predictions(Retail, "Purchase_Amount", "Return_Likelihood", 200)

# Calculate and report the 95% Prediction Interval for
# Purchase_Amount = 200
Prediction_Interval <- quantile(boots_predictions, probs = c(0.025, 0.975))

# Fitted multiple linear regression model for response variable of
# Return_Likelihood and predictor variables of Purchase_Amount,
# Discount_Applied, and Loyalty_Status.
multiple_model <- lm(Return_Likelihood ~ Purchase_Amount + Discount_Applied + Loyalty_Status, data = Re
# Gives intercept and coefficient values for each predictor variable
multiple_model_summary <- summary(multiple_model)

# Finds adjusted R-Squared for Return_Likelihood and
# Purchase_Amount
Simple_Regression_R_Squared <- model_summary$adj.r.squared

# Finds adjusted R-Squared for Return_Likelihood and
# Purchase_Amount, Discount_Applied, and Loyalty_Status
Multiple_Regression_R_Squared <- multiple_model_summary$adj.r.squared
```

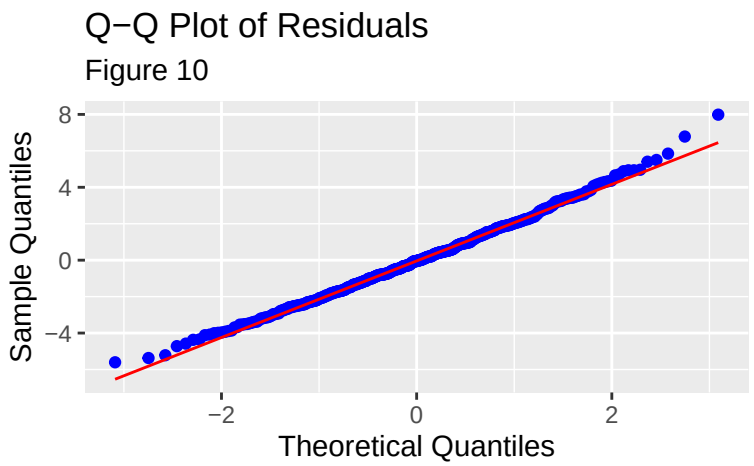
```r
# Prediction at Purchase_Amount = 200, Discount_Applied = 0,
# and Loyalty_Status = Regular
multiple_prediction <- predict(multiple_model, data.frame(Purchase_Amount = 200, Discount_Applied = 0, L

# Figure 8
# Linearity Check (Residuals vs. Fitted Plot)
ggplot(data = data.frame(Fitted = multiple_model$fitted.values, Residuals = multiple_model$residuals),
       aes(x = Fitted, y = Residuals)) +
  geom_point(color = "blue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Fitted Values", y = "Residuals", subtitle = "Figure 8") +
  ggtitle("Residuals vs Fitted")
```




```r
# Figure 9
# Normality Check (Histogram of Residuals)
ggplot(data.frame(Residuals = multiple_model$residuals), aes(x = Residuals)) +
  geom_histogram(bins = 30, color = "black", fill = "purple") +
  labs(x = "Residuals", y = "Frequency", subtitle = "Figure 9") +
  ggtitle("Histogram of Residuals")
```



```r
# Figure 10
# QQ Plot for Normality Check
ggplot(data = data.frame(Residuals = multiple_model$residuals), aes(sample = Residuals)) +
  stat_qq(color = "blue") +  # Create the Q-Q plot
  stat_qq_line(color = "red") +  # Add the reference line
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles",
       subtitle = "Figure 10") +
  ggtitle("Q-Q Plot of Residuals")
```



R Appendix

```r
# Loading Libraries
library(tidyverse)

# Data Import
Retail <- read_csv("Retail_Ecommerce_Dataset.csv")

# Number of observations in the data set
Observations <- nrow(Retail)

# Summary Statistics
Summmary_Retail <- summary(Retail)

# Standard Deviations of Relevent Numerical Variables
SD_Purchase_Amount <- sd(Retail$Purchase_Amount)
SD_Return_Likelihood <- sd(Retail$Return_Likelihood)

# Figure 1
# Histogram showing Purchase Amount of Customers
ggplot(Retail, aes(x = Purchase_Amount)) +
  geom_histogram(binwidth = 50, fill = "red", color = "black") +
  labs(x = "Purchase Amount (USD)", y = "Number of Customers",
       subtitle = "Figure 1") +
  ggtitle("Histogram showing Purchase Amount of Customers")
```

```r
# Figure 2
# Histogram showing Likelihood of a purchased product being returned
ggplot(Retail, aes(x = Return_Likelihood)) +
  geom_histogram(binwidth = 5, fill = "red", color = "black") +
  labs(x = "Return Likelihood (scaled from 0 to 100)",
       y = "Number of Customers",
       subtitle = "Figure 2") +
  ggtitle("Histogram showing Likelihood of Customer Returning a Purchased Product")

# Figure 3
# Scatter plot showing relationship between purchase
# amount and likelihood of product being returned.
ggplot(Retail, aes(x = Purchase_Amount, y = Return_Likelihood)) +
  geom_point() +
  labs(x = "Purchase Amount", y = "Return Likelihood",
       subtitle = "Figure 3") +
  ggtitle("Purchase Amount vs Return Likelihood")

# Figure 4
# Box plot showing the the distribution of the
# relationship between products with a discount applied
# and the likelihood of said product being returned.
ggplot(Retail, aes(x = factor(Discount_Applied), y = Return_Likelihood)) +
  geom_boxplot() +
  labs(x = "Discount Applied", y = "Return Likelihood",
       subtitle = "Figure 4") +
  ggtitle("Return Likelihood with Discount vs No Discount")

# Figure 5
# Box plot showing the distribution of the relationship
# between a customer's loyalty status and the likelihood
# of the purchased product being returned.
ggplot(Retail, aes(x = Loyalty_Status, y = Return_Likelihood)) +
  geom_boxplot() +
  labs(x = "Loyalty Status of Customer", y = "Return Likelihood",
       subtitle = "Figure 5") +
  ggtitle("Return Likelihood based on Customer Loyalty Status")

# Fitted Simple Linear Regression Model for response variable of
# Return Likelihood and predictor variable of Purchase Amount.
model <- lm(Return_Likelihood ~ Purchase_Amount, data = Retail)
model_summary = summary(model)

# R-squared value for my simple linear regression model.
R_Squared <- model_summary$r.squared

# Creates data frame for fitted values and residuals in the simple
# linear regression model created above.
model_df <- data.frame(Fitted_Values = model$fitted.values,
                       Residuals = model$residuals)

# Figure 6
# Residual Plot for my simple linear regression model
```

```r
ggplot(model_df, aes(x = Fitted_Values, y = Residuals)) +
  geom_point() +
  labs(x = "Fitted Values", y = "Residuals",
       subtitle = "Figure 6") +
  ggtitle("Residual Plot for Simple Linear Regression Model")

# Residual values in simple linear regression model
residuals <- model_summary$residuals
residuals_df <- data.frame(residuals)

# Figure 7
# Histogram of residuals in simple linear regression
ggplot(residuals_df, aes(x = residuals)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "darkblue") +
  labs(x = "Residuals", y = "Number of Residuals",
       subtitle = "Figure 7") +
  ggtitle("Histogram of Residuals in Simple Linear Regression Model")

# Defines a function to perform bootstrapping for the slope of a
# linear regression model.
bootstrap_slope <- function(df, x, y, repetitions = 5000) {
  # Number of rows in the dataset
  n = nrow(df)
  # Helper function to calculate slope of a linear regression model
  slope <- function(x, y) {
    model <- lm(y ~ x)
    return(coef(model)[2])
  }

  slopes <- sapply(1:repetitions, function(i) {
    indices = sample(1:n, size = n, replace = T)
    sample_df = df[indices,]
    slope(x = sample_df[[x]], y = sample_df[[y]])
    })
# Find 2.5th and 97.5th percentiles for the 95% CI
ci <- quantile(slopes, probs = c(0.025, 0.975))
# Compute the slope on the original data
observed_slope <- slope(df[[x]], df[[y]])

list(observed_slope = observed_slope, ci = ci,
     all_slopes = slopes)
}

boots_slope <- bootstrap_slope(Retail, "Purchase_Amount", "Return_Likelihood")

# Grabs the 95% confidence interval calculated from the above
# boostrap_slope function
Bootstrap_CI <- boots_slope$ci

# Prediction at Purchase_Amount = 200
prediction <- predict(model, data.frame(Purchase_Amount = 200))

# Defines a function to perform bootstrapping for predictions
```

```r
bootstrap_predictions <- function(data, x_col, y_col, new_x, repetitions = 5000, seed = 123) {
  # Sets seed for reproducibility
  set.seed(seed)

  # Generates bootstrap predictions
  predictions <- replicate(repetitions, {
    # Resample the data with replacement
    resample <- data %>% sample_n(nrow(data), replace = TRUE)

    # Fit the linear regression model on the bootstrap sample
    boot_model <- lm(as.formula(paste(y_col, "~", x_col)),
                     data = resample)

    # Creates a new data frame for the prediction
    new_data <- data.frame(new_x)
    names(new_data) <- x_col

    # Predict the Return Likelihood on the new data point
    predict(boot_model, new_data)
  })

  return(predictions)
}

# Run the bootstrap function for a new Purchase_Amount = 200
boots_predictions <- bootstrap_predictions(Retail, "Purchase_Amount", "Return_Likelihood", 200)

# Calculate and report the 95% Prediction Interval for
# Purchase_Amount = 200
Prediction_Interval <- quantile(boots_predictions,
                                probs = c(0.025, 0.975))

# Fitted multiple linear regression model for response variable of
# Return_Likelihood and predictor variables of Purchase_Amount,
# Discount_Applied, and Loyalty_Status.
multiple_model <- lm(Return_Likelihood ~ Purchase_Amount + Discount_Applied + Loyalty_Status, data = Re
# Gives intercept and coefficient values for each predictor variable
multiple_model_summary <- summary(multiple_model)

# Finds adjusted R-Squared for Return_Likelihood and
# Purchase_Amount
Simple_Regression_R_Squared <- model_summary$adj.r.squared

# Finds adjusted R-Squared for Return_Likelihood and
# Purchase_Amount, Discount_Applied, and Loyalty_Status
Multiple_Regression_R_Squared <- multiple_model_summary$adj.r.squared

# Prediction at Purchase_Amount = 200, Discount_Applied = 0,
# and Loyalty_Status = Regular
multiple_prediction <- predict(multiple_model, data.frame(Purchase_Amount = 200, Discount_Applied = 0, L

# Figure 8
# Linearity Check (Residuals vs. Fitted Plot)
```

```r
ggplot(data = data.frame(Fitted = multiple_model$fitted.values, Residuals = multiple_model$residuals),
       aes(x = Fitted, y = Residuals)) +
  geom_point(color = "blue") +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(x = "Fitted Values", y = "Residuals",
       subtitle = "Figure 8") +
  ggtitle("Residuals vs Fitted")

# Figure 9
# Normality Check (Histogram of Residuals)
ggplot(data.frame(Residuals = multiple_model$residuals),
       aes(x = Residuals)) +
  geom_histogram(bins = 30, color = "black", fill = "purple") +
  labs(x = "Residuals", y = "Frequency", subtitle = "Figure 9") +
  ggtitle("Histogram of Residuals")

# Figure 10
# QQ Plot for Normality Check
ggplot(data = data.frame(Residuals = multiple_model$residuals),
       aes(sample = Residuals)) +
  stat_qq(color = "blue") +  # Create the Q-Q plot
  stat_qq_line(color = "red") +  # Add the reference line
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles",
       subtitle = "Figure 10") +
  ggtitle("Q-Q Plot of Residuals")
```