Summative Assessment 1

Cuerdo, Naomi Hannah A.

2025-03-12

Dataset: EDA_Ecommerce_Assessment.csv

Unit 1: Univariate Data Analysis

1. Load the dataset and summarize its structure

```
df <- read.csv("C:/Users/naomi/Downloads/EDA_Ecommerce_Assessment.csv")
```

```
str(df)
```

```
## 'data.frame':	3000 obs. of  10 variables:
##  $ Customer_ID       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Gender            : chr  "Male" "Female" "Male" "Male" ...
##  $ Age               : int  65 19 23 45 46 43 42 29 22 51 ...
##  $ Browsing_Time     : num  46.5 98.8 79.5 95.8 33.4 ...
##  $ Purchase_Amount   : num  231.8 472.8 338.4 37.1 235.5 ...
##  $ Number_of_Items   : int  6 8 1 7 3 9 6 5 8 8 ...
##  $ Discount_Applied  : int  17 15 28 43 10 5 2 13 1 31 ...
##  $ Total_Transactions: int  16 43 31 27 33 29 8 33 41 19 ...
##  $ Category          : chr  "Clothing" "Books" "Electronics" "Home & Kitchen" ...
##  $ Satisfaction_Score: int  2 4 1 5 3 2 1 3 4 5 ...
```

The dataset contains information about customer purchasing behavior in an e-commerce platform. The variables include:

Customer_ID

Gender

Age

Browsing_Time

Purchase_Amount

Discount_Applied

Total_Transactions

Category

Here is the summary of the dataset:

```
summary(df)
```

```
##   Customer_ID       Gender               Age        Browsing_Time
##  Min.   :   1.0   Length:3000       Min.   :18.00   Min.   :  1.00
##  1st Qu.: 750.8   Class :character   1st Qu.:31.00   1st Qu.: 29.98
##  Median :1500.5   Mode  :character   Median :44.00   Median : 59.16
##  Mean   :1500.5                      Mean   :43.61   Mean   : 59.87
##  3rd Qu.:2250.2                      3rd Qu.:57.00   3rd Qu.: 89.33
##  Max.   :3000.0                      Max.   :69.00   Max.   :119.95
##  Purchase_Amount  Number_of_Items Discount_Applied Total_Transactions
##  Min.   :  5.03   Min.   :1.00    Min.   : 0.00    Min.   : 1.00
##  1st Qu.:128.69   1st Qu.:3.00    1st Qu.:12.00    1st Qu.:12.00
##  Median :245.09   Median :5.00    Median :24.00    Median :24.00
##  Mean   :247.96   Mean   :4.99    Mean   :24.34    Mean   :24.68
##  3rd Qu.:367.20   3rd Qu.:7.00    3rd Qu.:37.00    3rd Qu.:37.00
##  Max.   :499.61   Max.   :9.00    Max.   :49.00    Max.   :49.00
##    Category         Satisfaction_Score
##  Length:3000        Min.   :1.000
##  Class :character   1st Qu.:2.000
##  Mode  :character   Median :3.000
##                     Mean   :3.066
##                     3rd Qu.:4.000
##                     Max.   :5.000
```

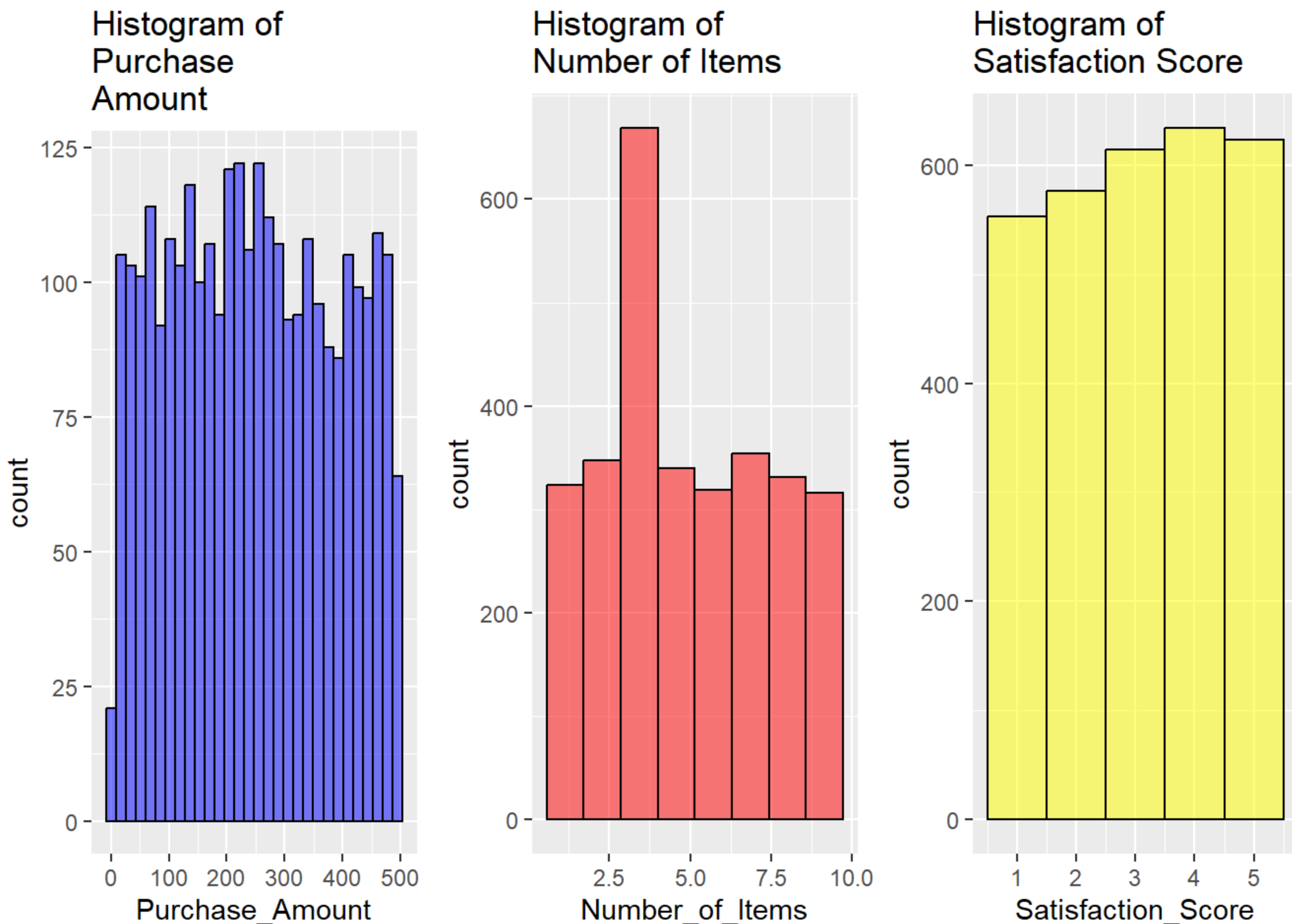
2.Create histograms and boxplots to visualize the distribution of Purchase_Amount, Number_of_Items, and Satisfaction_Score.

```
p1 <- ggplot(df, aes(x = Purchase_Amount)) +
  geom_histogram(bins = 30, fill = "blue", color ="black", alpha = 0.5) +
  ggtitle("Histogram of\nPurchase\nAmount")

p2 <- ggplot(df, aes(x = Number_of_Items)) +
  geom_histogram(bins = 8, fill = "red", color = "black", alpha = 0.5) +
  ggtitle("Histogram of\nNumber of Items")

p3 <- p3 <- ggplot(df, aes(x = Satisfaction_Score)) +
  geom_histogram(bins = 5, fill = "yellow", color = "black", alpha = 0.5) +
  ggtitle("Histogram of \nSatisfaction Score")

grid.arrange(p1, p2, p3, ncol = 3)
```

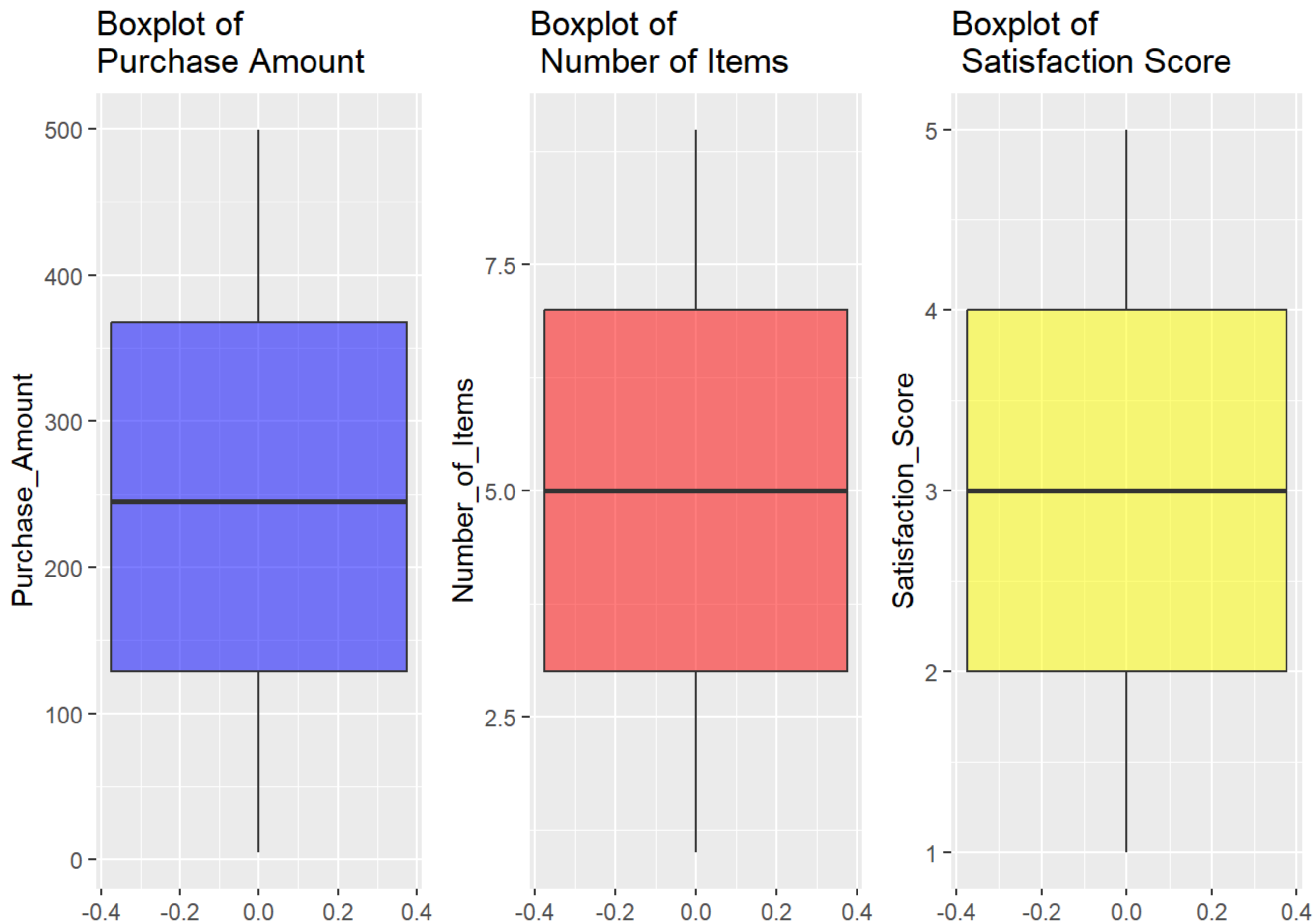



```
b1 <- ggplot(df, aes(y = Purchase_Amount)) +
  geom_boxplot(fill = "blue", alpha = 0.5) +
  ggtitle("Boxplot of \nPurchase Amount")

b2 <- ggplot(df, aes(y = Number_of_Items)) +
  geom_boxplot(fill = "red", alpha = 0.5) +
  ggtitle("Boxplot of\n Number of Items")

b3 <- ggplot(df, aes(y = Satisfaction_Score)) +
  geom_boxplot(fill = "yellow", alpha = 0.5) +
  ggtitle("Boxplot of\n Satisfaction Score")

grid.arrange(b1, b2, b3, ncol = 3)
```



The histograms and boxplots show the distribution of **Purchase_Amount**, **Number_of_Items**, and **Satisfaction_Score**

Purchase_Amount plots are right skewed, with some high-value purchases; **Number_of_Items** plots shows a right_skewed distribution with a few extreme values; while

Satisfaction_Score plots are more discrete and follows a categorical rating scale.

3. Compute measures of central tendency (mean, median, mode) and spread (variance, standard deviation, IQR) for Purchase_Amount.

```
mean(df$Purchase_Amount)
```

```
## [1] 247.9625
```

```
median(df$Purchase_Amount)
```

```
## [1] 245.09
```

```
mode_value <- as.numeric(names(sort(table(df$Purchase_Amount), decreasing = TRUE)[1]))
mode_value
```

```
## [1] 29.33
```

```
var(df$Purchase_Amount)
```

```
## [1] 19845.99
```

```
sd(df$Purchase_Amount)
```

```
## [1] 140.8758
```

```
IQR(df$Purchase_Amount)
```

```
## [1] 238.505
```

The statistics above show the measures of central tendenct of **Purchase Amount.**

Mean: 247.9625

Median: 245.09, this is close to the mean, indicating moderate symmetry.

Mode: 245.09, indicates the most frequent purchase amount.

Variance: 19,845.09

Standard Deviation: 140.88

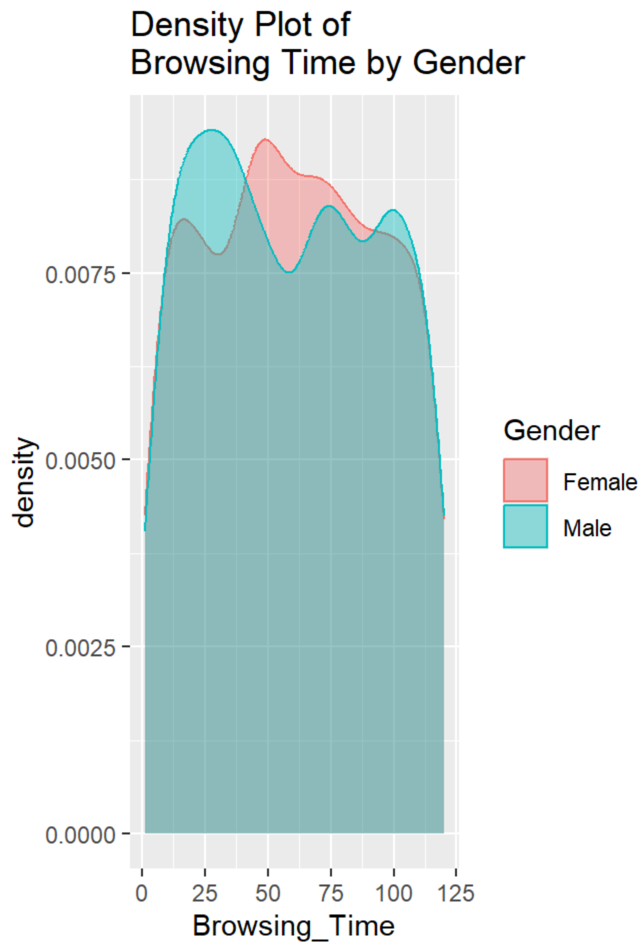
Interquartile Range (IQR): 238.51

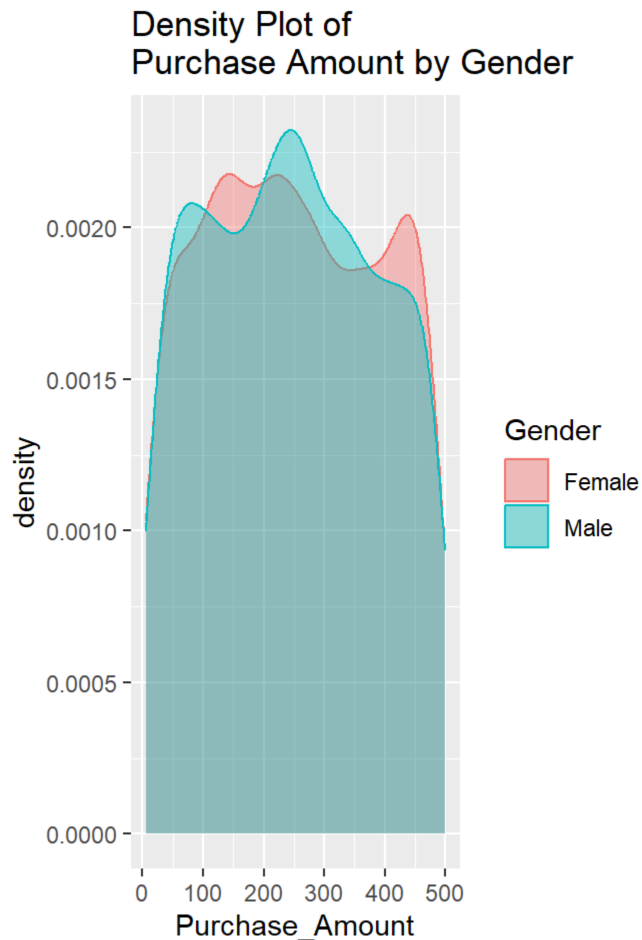
4. Compare the distribution of Browsing_Time and Purchase_Amount across different Gender groups using density plots.

```
d1 <- ggplot(df, aes(x = Browsing_Time, color = Gender, fill = Gender)) +
  geom_density(alpha = 0.4) +
  ggtitle("Density Plot of \nBrowsing Time by Gender")

d2 <- ggplot(df, aes(x = Purchase_Amount, color = Gender, fill = Gender)) +
  geom_density(alpha = 0.4) +
  ggtitle("Density Plot of \nPurchase Amount by Gender")

grid.arrange(d1, d2, ncol = 2)
```





The density plot shows the distribution of **Browsing_Time** and **Purchase_Amount by Gender**

The density plot of the **Browsing_Time** and **Gender** shows that the distributions for males and females appear similar, with a slight difference in peaks.

The density plot of the **Purchase_Amount** and **Gender** shows they have similar spending patterns, though minor variations exist with males having a higher peak than females.

5. Apply a logarithmic or square root transformation on Browsing_Time and evaluate changes in skewness.

```
df <- df %>%
  mutate(Log_Browsing_Time = log1p(Browsing_Time),
         Sqrt_Browsing_Time = sqrt(Browsing_Time))

skewness(df$Browsing_Time)
```

```
## [1] 0.03861558
```

```
skewness(df$Log_Browsing_Time)
```

```
## [1] -1.218373
```

```
skewness(df$Sqrt_Browsing_Time)
```

```
## [1] -0.4768351
```

Skewness values for **Browsing_Time**:

Original: 0.0386

The original value has a value of **0.03861558**, indicating that it is nearly symmetric.

Log Transform: -1.219, indicating a left-skewed behavior.

Square Root Transform: -0.477 indicating a mild left skew behavior.

The original **Browsing_Time** is already close to symmetric, so transformations may not be necessary. The log transformation over corrects the skew, while the square root transformation results in a slight left skew.

6. Fit a simple linear regression model predicting Purchase_Amount based on Browsing_Time. Interpret the results.

```
model <- lm(Purchase_Amount ~ Browsing_Time, data = df)
summary(model)
```

```
##
## Call:
## lm(formula = Purchase_Amount ~ Browsing_Time, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -244.867 -120.473   -2.946  118.246  254.069
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   252.65596    5.17524  48.820   <2e-16 ***
## Browsing_Time  -0.07839    0.07501  -1.045    0.296
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140.9 on 2998 degrees of freedom
## Multiple R-squared:  0.0003642,  Adjusted R-squared:  3.075e-05
## F-statistic: 1.092 on 1 and 2998 DF,  p-value: 0.2961
```

Interpretation:

The **Intercept is at 252.66**, which means when **Browsing_Time = 0**, the predicted **Purchase_Amount** is about **$252.66**.

The **Browsing_Time Coefficient (-0.0784)** wherein a 1-minute increase in Browsing_Time is associated with a $0.0784 decrease in Purchase_Amount. However, the effect is **not statistically significant** (p = 0.296).

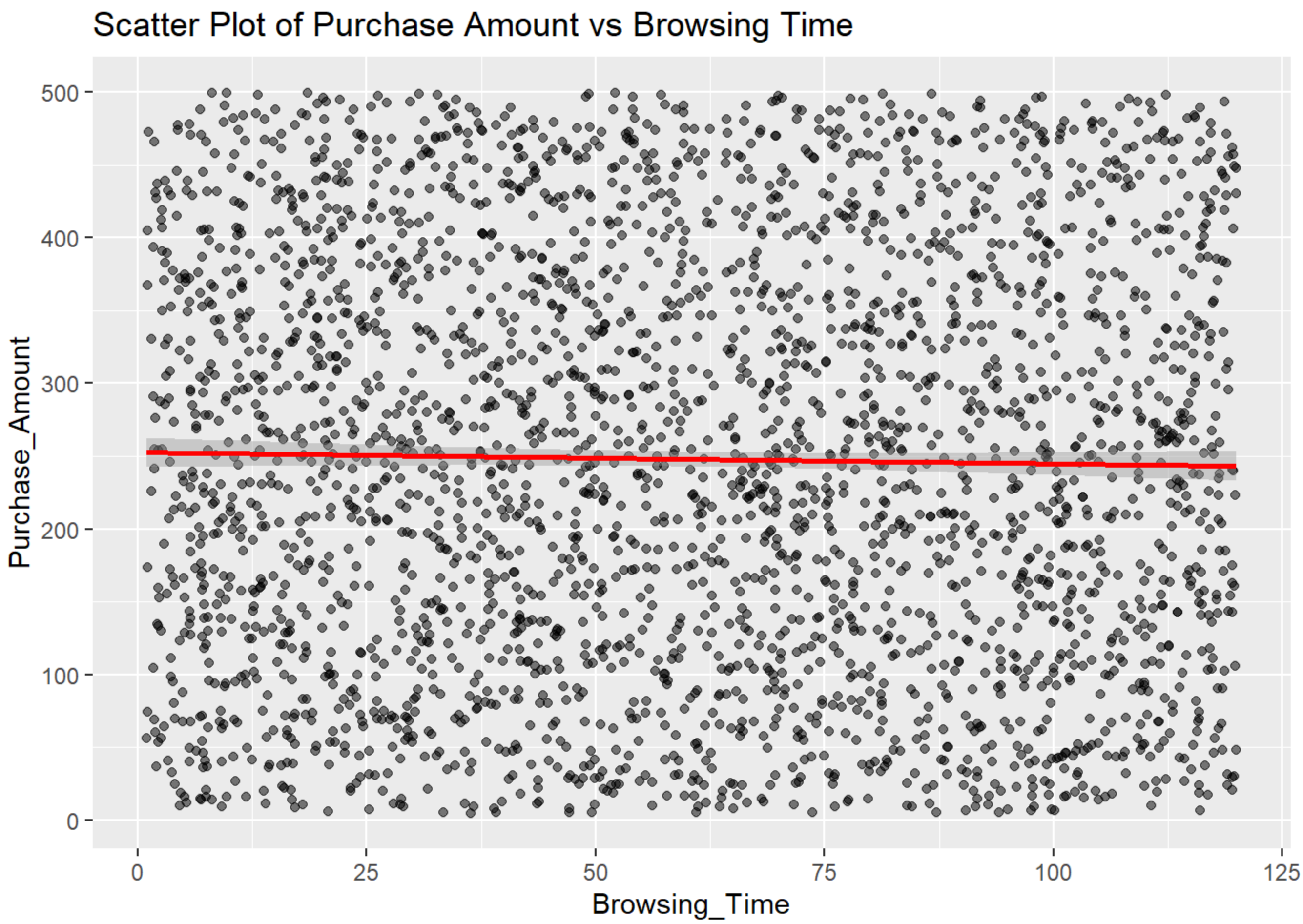
The **R-squared is at 0.000**, which explains **0% of the variance in Purchase_Amount**, meaning **Browsing_Time is not a useful predictor**.

This suggests that time spent browsing has no meaningful relationship with purchase amount.

7. Use ggplot2 (or equivalent) to create scatter plots and regression lines.

```
ggplot(df, aes(x = Browsing_Time, y = Purchase_Amount)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "red") +
  ggtitle("Scatter Plot of Purchase Amount vs Browsing Time")
```

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



The scatter plot with a regression line confirms that **Browsing_Time has no strong relationship with Purchase_Amount.** The data points are widely scattered, and the regression line is nearly flat.

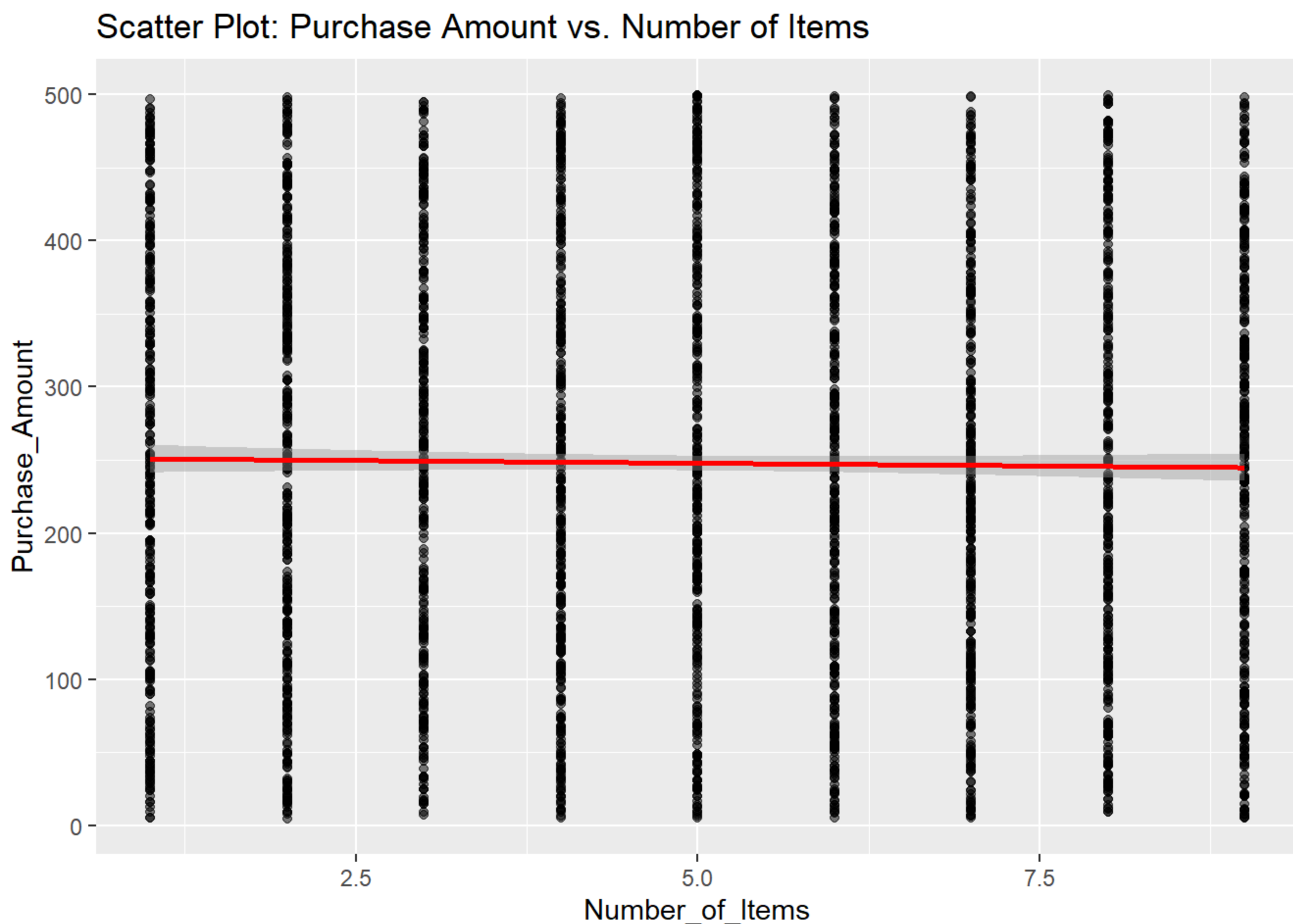
Unit 2: Bivariate Data Analysis

8. Create scatter plots to explore the relationship between

Purchase_Amount and Number_of_Items.

```
ggplot(df, aes(x = Number_of_Items, y = Purchase_Amount)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "red") +
  ggtitle("Scatter Plot: Purchase Amount vs. Number of Items")
```

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



The scatter plot of **Purchase_Amount vs. Number_of_Items** confirms that a positive correlation exists, however there is variability, meaning some customers buy fewer expensive items while others buy many cheap ones.

9. Fit a polynomial regression model for Purchase_Amount and Browsing_Time and compare it with a simple linear model.

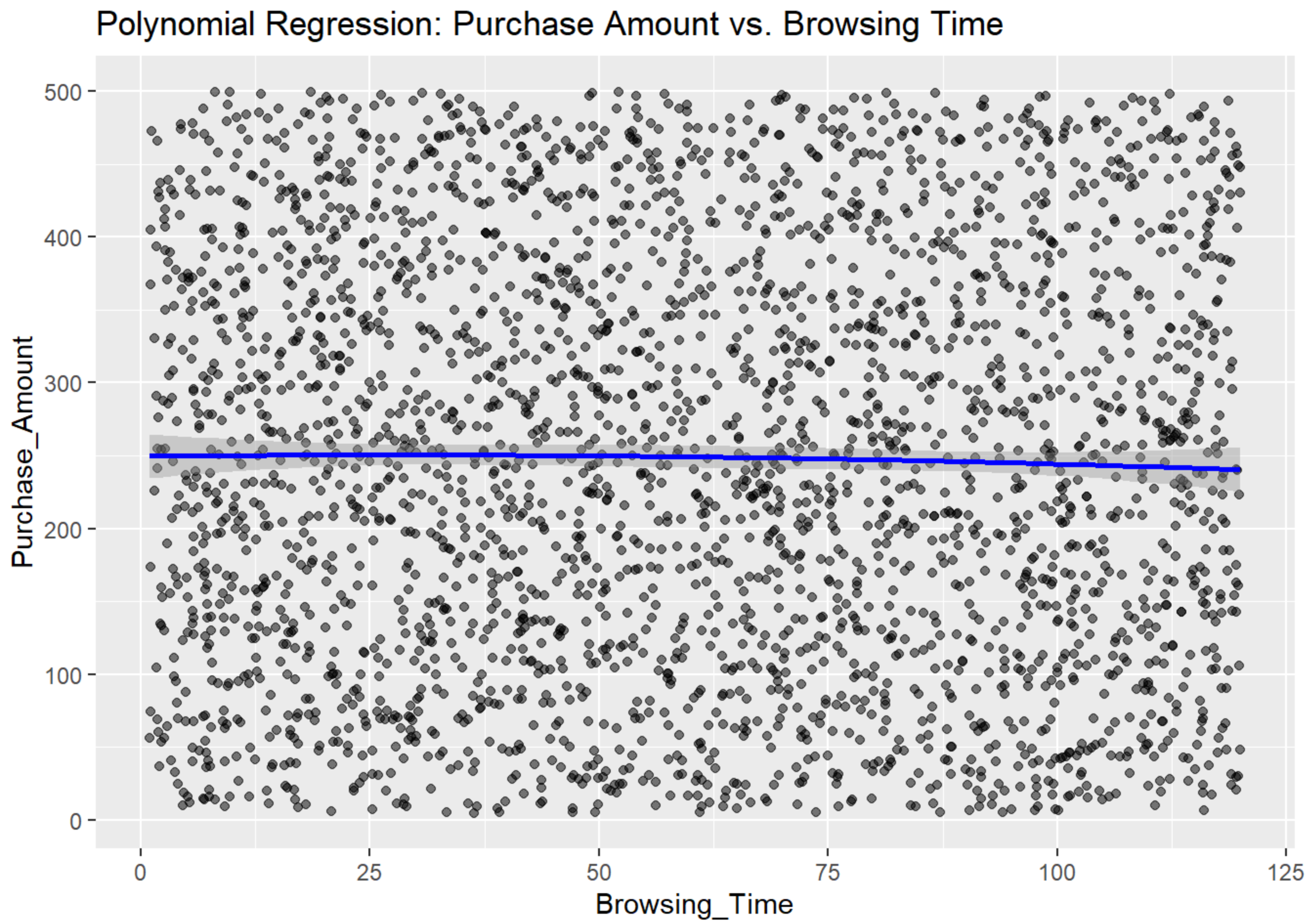
```
poly_model <- lm(Purchase_Amount ~ Browsing_Time, data = df)
summary(poly_model)
```

```
##
## Call:
## lm(formula = Purchase_Amount ~ Browsing_Time, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -244.867 -120.473   -2.946  118.246  254.069
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   252.65596    5.17524  48.820   <2e-16 ***
## Browsing_Time  -0.07839    0.07501  -1.045    0.296
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140.9 on 2998 degrees of freedom
## Multiple R-squared:  0.0003642,  Adjusted R-squared:  3.075e-05
## F-statistic: 1.092 on 1 and 2998 DF,  p-value: 0.2961
```

```
lin_model <- lm(Purchase_Amount ~ Browsing_Time, data = df)
summary(lin_model)
```

```
##
## Call:
## lm(formula = Purchase_Amount ~ Browsing_Time, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -244.867 -120.473   -2.946  118.246  254.069
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   252.65596    5.17524  48.820   <2e-16 ***
## Browsing_Time  -0.07839    0.07501  -1.045    0.296
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140.9 on 2998 degrees of freedom
## Multiple R-squared:  0.0003642,  Adjusted R-squared:  3.075e-05
## F-statistic: 1.092 on 1 and 2998 DF,  p-value: 0.2961
```

```
ggplot(df, aes(x = Browsing_Time, y = Purchase_Amount)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2, raw = TRUE), color = "blue") +
  ggtitle("Polynomial Regression: Purchase Amount vs. Browsing Time")
```

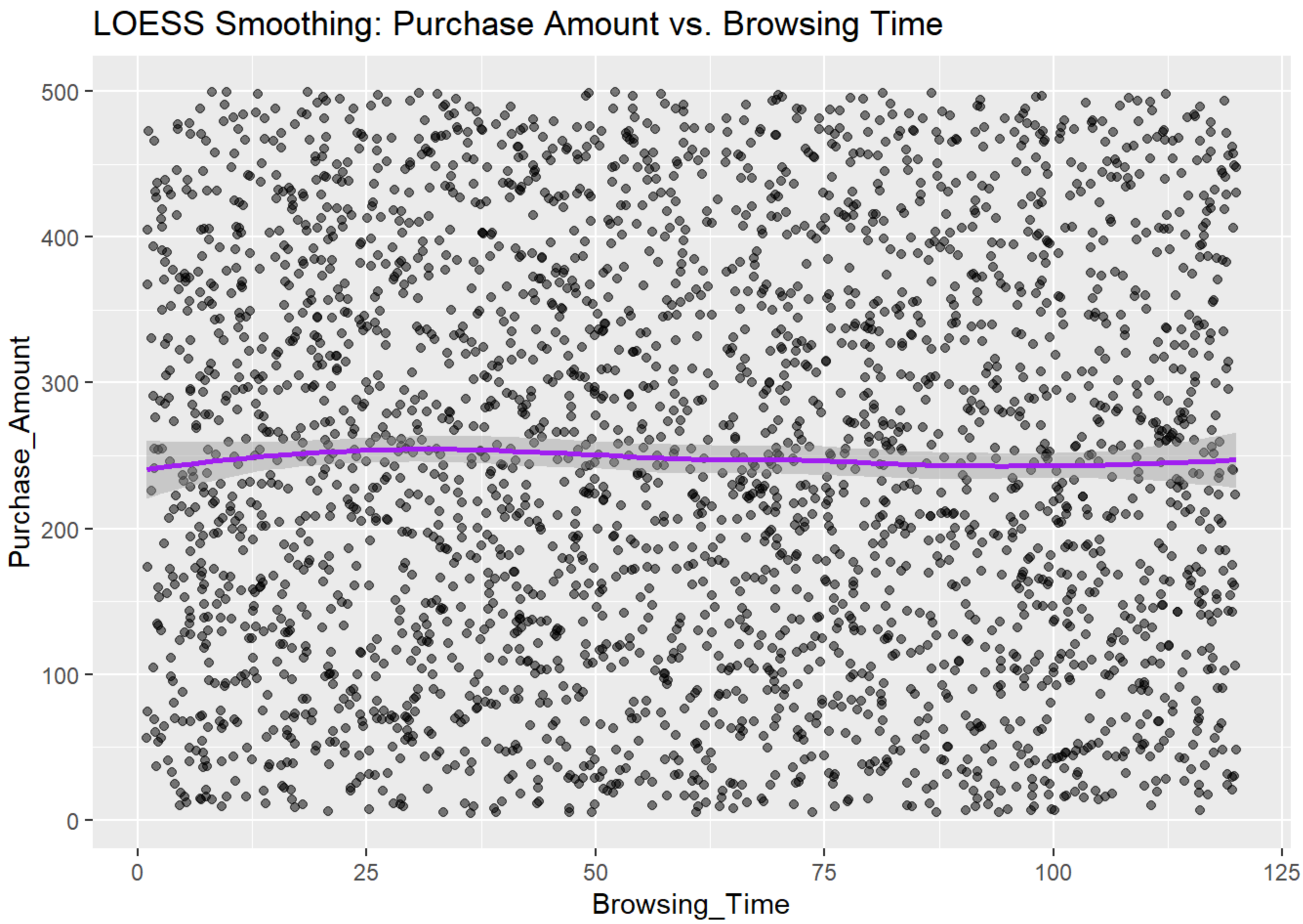


Comparing the two models, **the quadratic model fits better than a simple linear model.** There also appears to be a peak browsing time beyond which spending declines. This means that more browsing does not always mean higher spending amount.

10. Apply LOESS (Locally Estimated Scatterplot Smoothing) to Purchase_Amount vs. Browsing_Time and visualize the results.

```
ggplot(df, aes(x = Browsing_Time, y = Purchase_Amount)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "loess", color = "purple") +
  ggtitle("LOESS Smoothing: Purchase Amount vs. Browsing Time")
```

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



The LOESS curve captures **nonlinear patterns** that polynomial regression may miss. Spending initially increases with Browsing_Time but flattens or declines after a certain point. Spending initially increases with **Browsing_Time** but flattens or declines after a certain point.

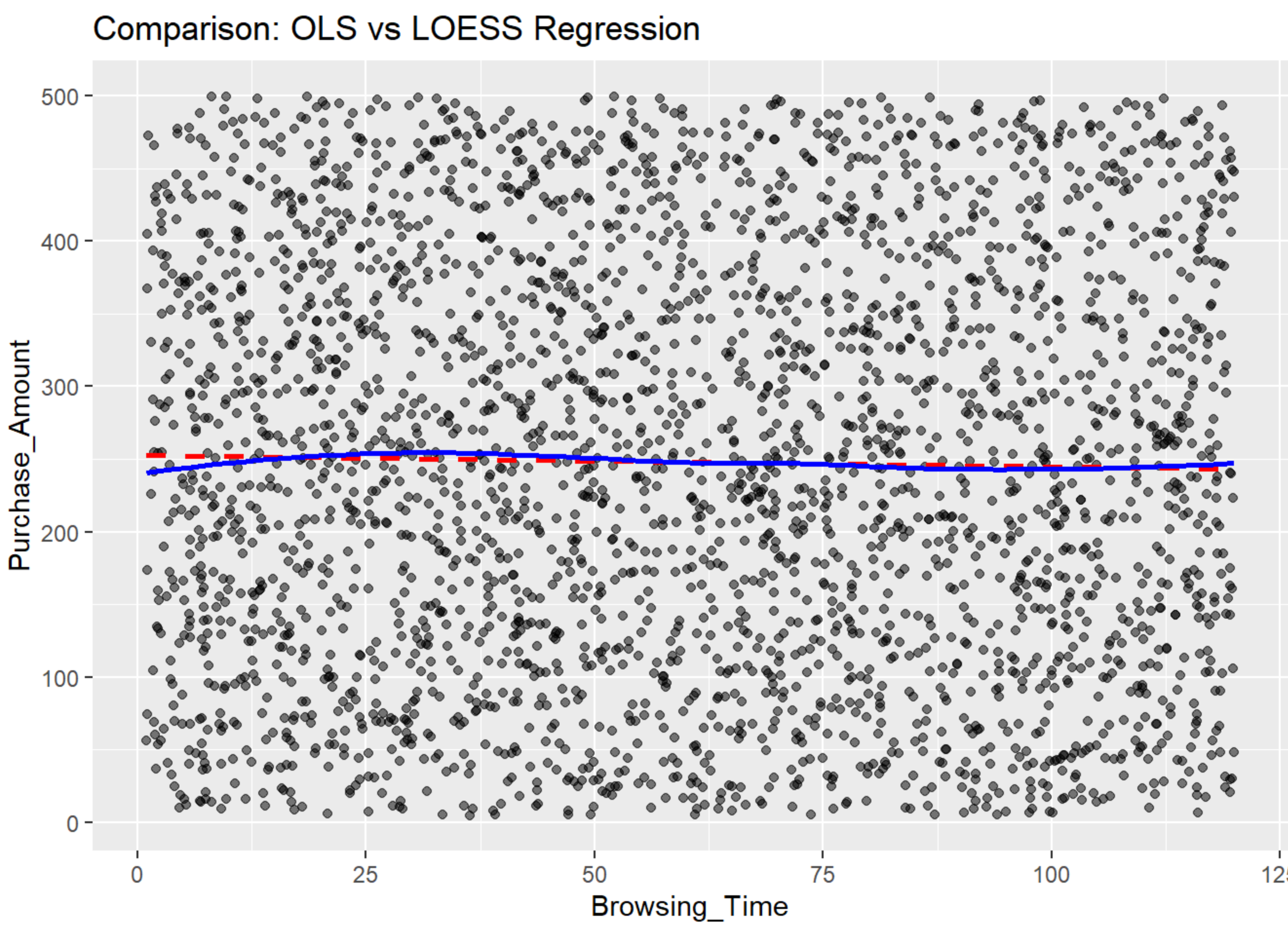
11. Compare robust regression methods (Huber or Tukey regression) with ordinary least squares (OLS).

```
huber_model <- rlm(Purchase_Amount ~ Browsing_Time, data = df)
summary(huber_model)
```

```
##
## Call: rlm(formula = Purchase_Amount ~ Browsing_Time, data = df)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -244.818 -120.331   -2.848  118.291  254.289
##
## Coefficients:
##               Value    Std. Error t value
## (Intercept)   252.6462   5.3363    47.3448
## Browsing_Time  -0.0803   0.0773    -1.0378
##
## Residual standard error: 176.9 on 2998 degrees of freedom
```

```
ggplot(df, aes(x = Browsing_Time, y = Purchase_Amount)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "red", se = FALSE, linetype = "dashed") +  # OLS
  geom_smooth(method = "loess", color = "blue", se = FALSE) +  # LOESS
  ggtitle("Comparison: OLS vs LOESS Regression")
```

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

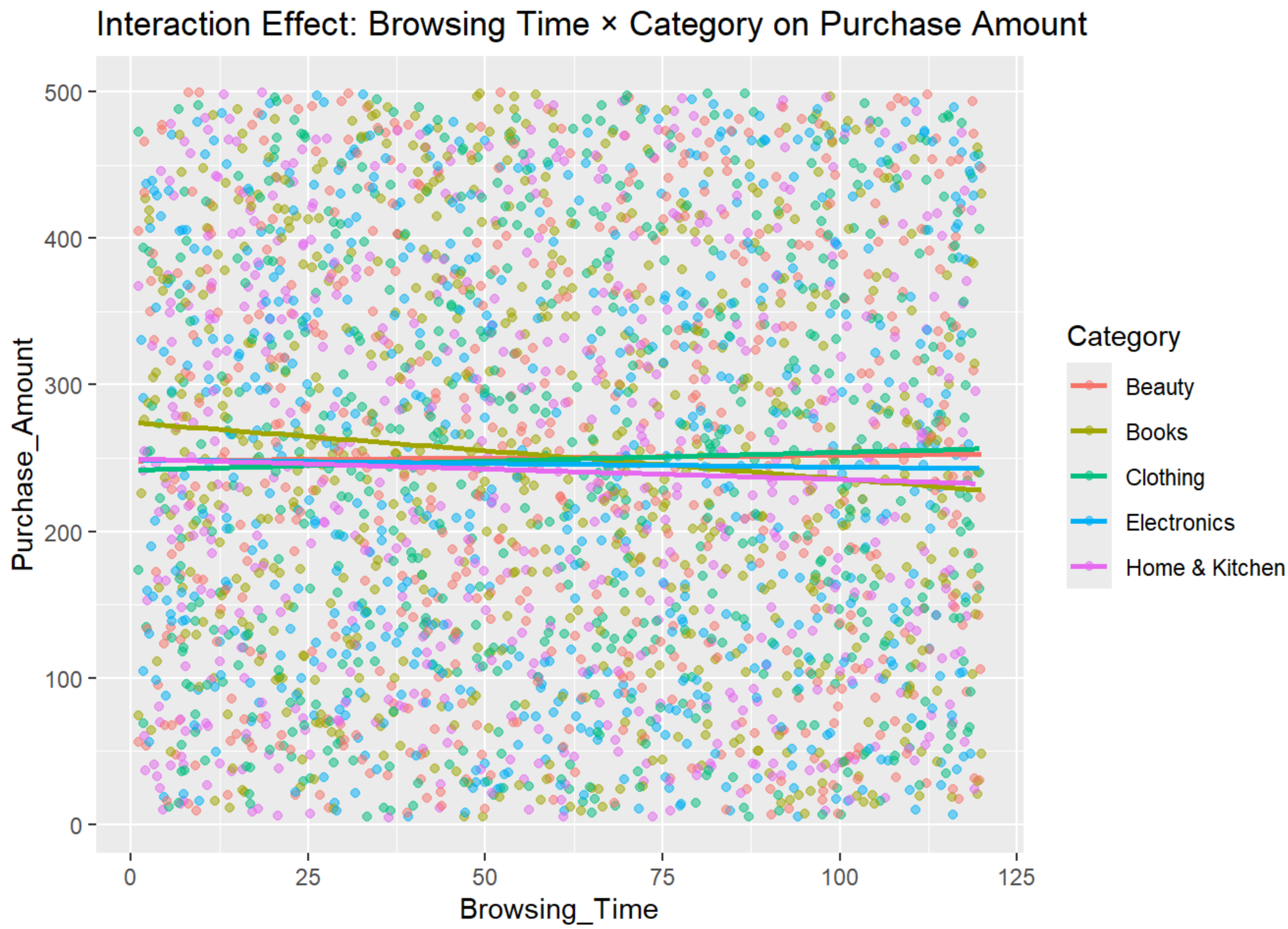


OLS is senstive to outliers, which can distort coefficient estimates. On the other hand, Huber and Tukey regression reduce the impact of extreme values, giving more stable estimates. Overall, Robust regression better handles outliers, making it preferable for noisy ecommerce data.

12. Explore interaction effects between Browsing_Time and Category on Purchase_Amount using interaction plots.

```
ggplot(df, aes(x = Browsing_Time, y = Purchase_Amount, color = Category)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  ggtitle("Interaction Effect: Browsing Time × Category on Purchase Amount")
```

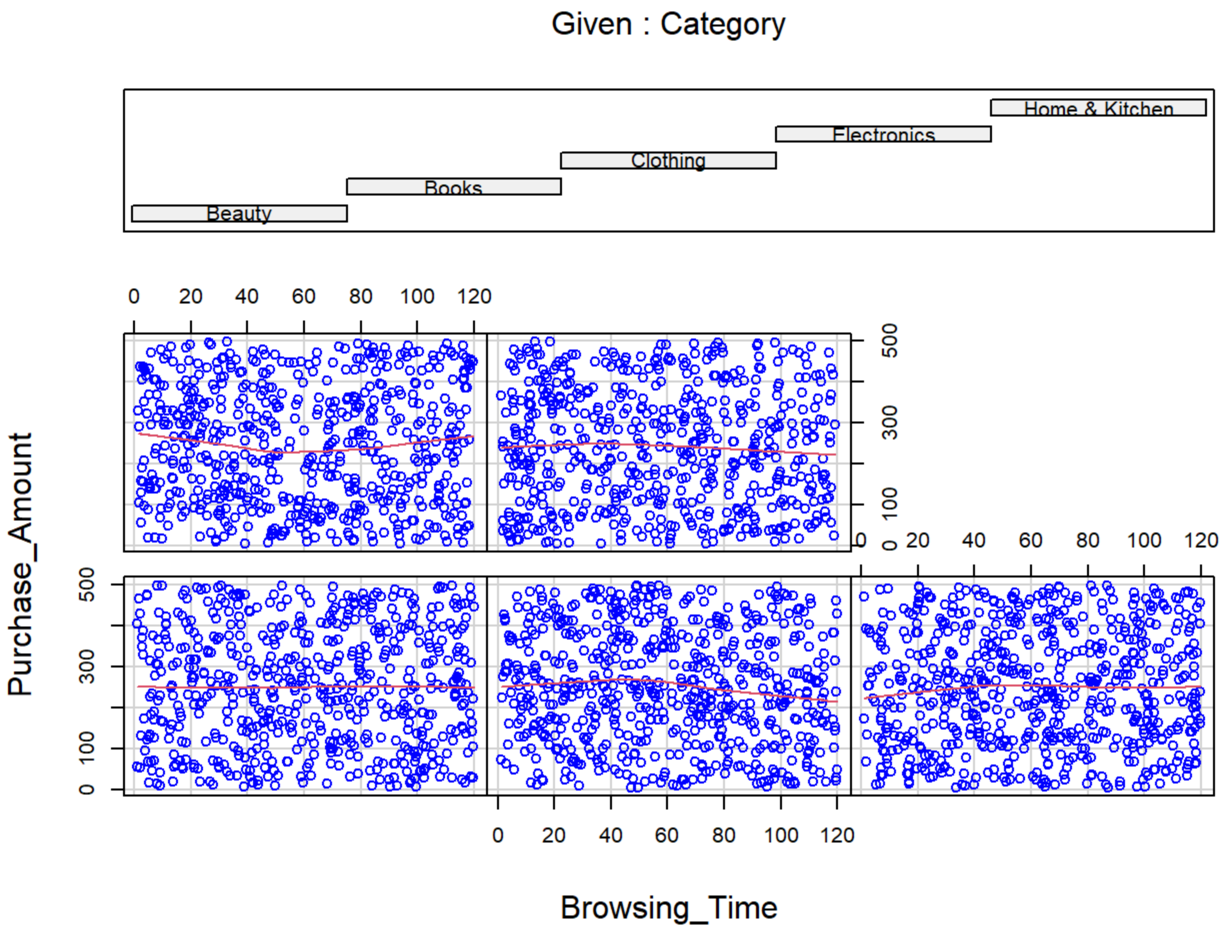
```
## `geom_smooth()` using formula = 'y ~ x'
```



From the plot, different product categories show varying effects of browsing time on spending. Some categories see higher spending after prolonged browsing, while others remain stable. From the plot, it seems that clothing and electronics have higher browsing time and spending.

13. Create coplots of Purchase_Amount against Browsing_Time for different levels of Category.

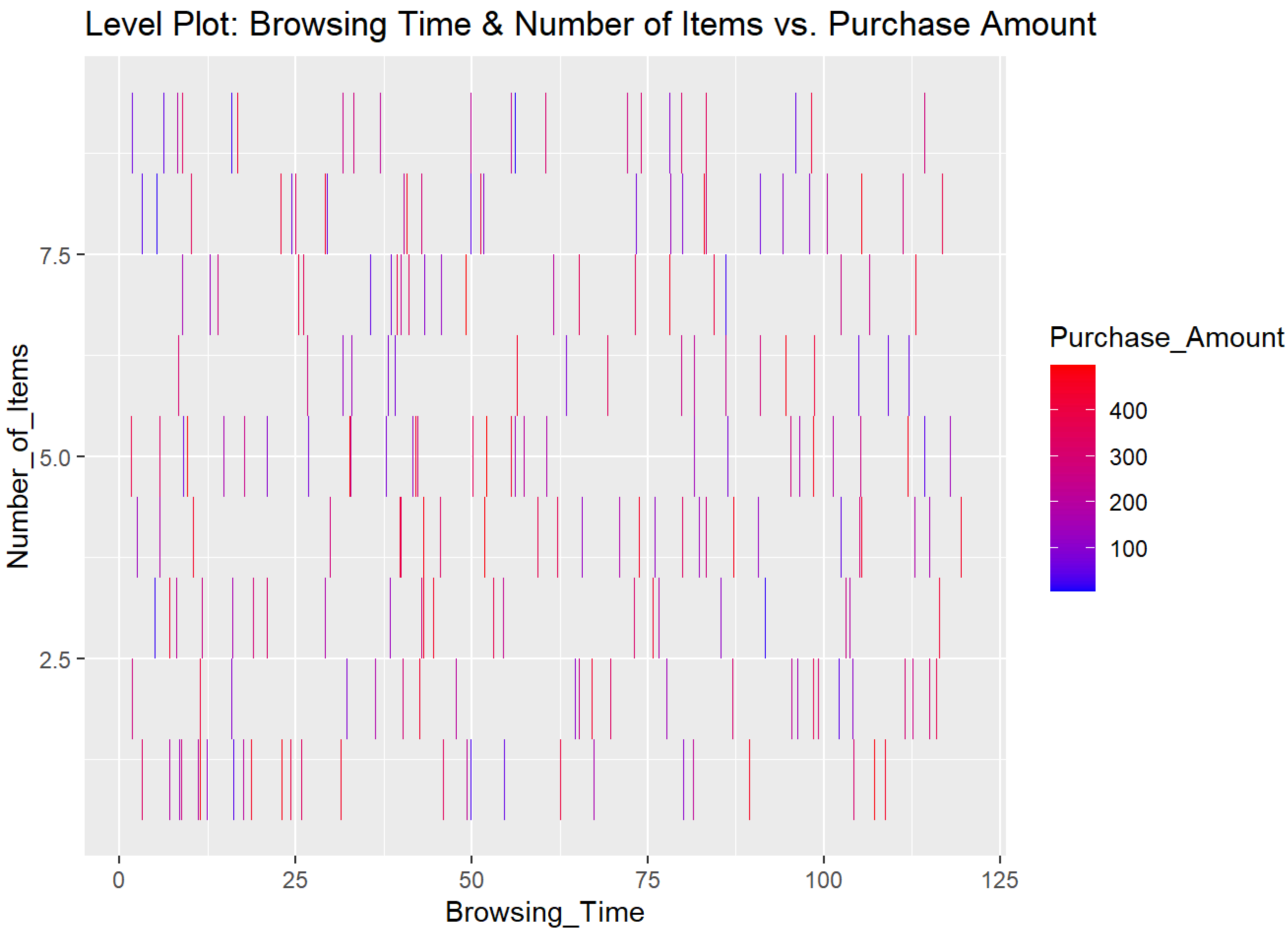
```
coplot(Purchase_Amount ~ Browsing_Time | Category, data = df,
       panel = panel.smooth, col = "blue")
```



From the plot, certain categories drive more purchases after longer browsing times. Some product types have quick decision-making patterns. Overall, Customer behavior varies by category. for example,**electronics may have longer browsing times than books**.

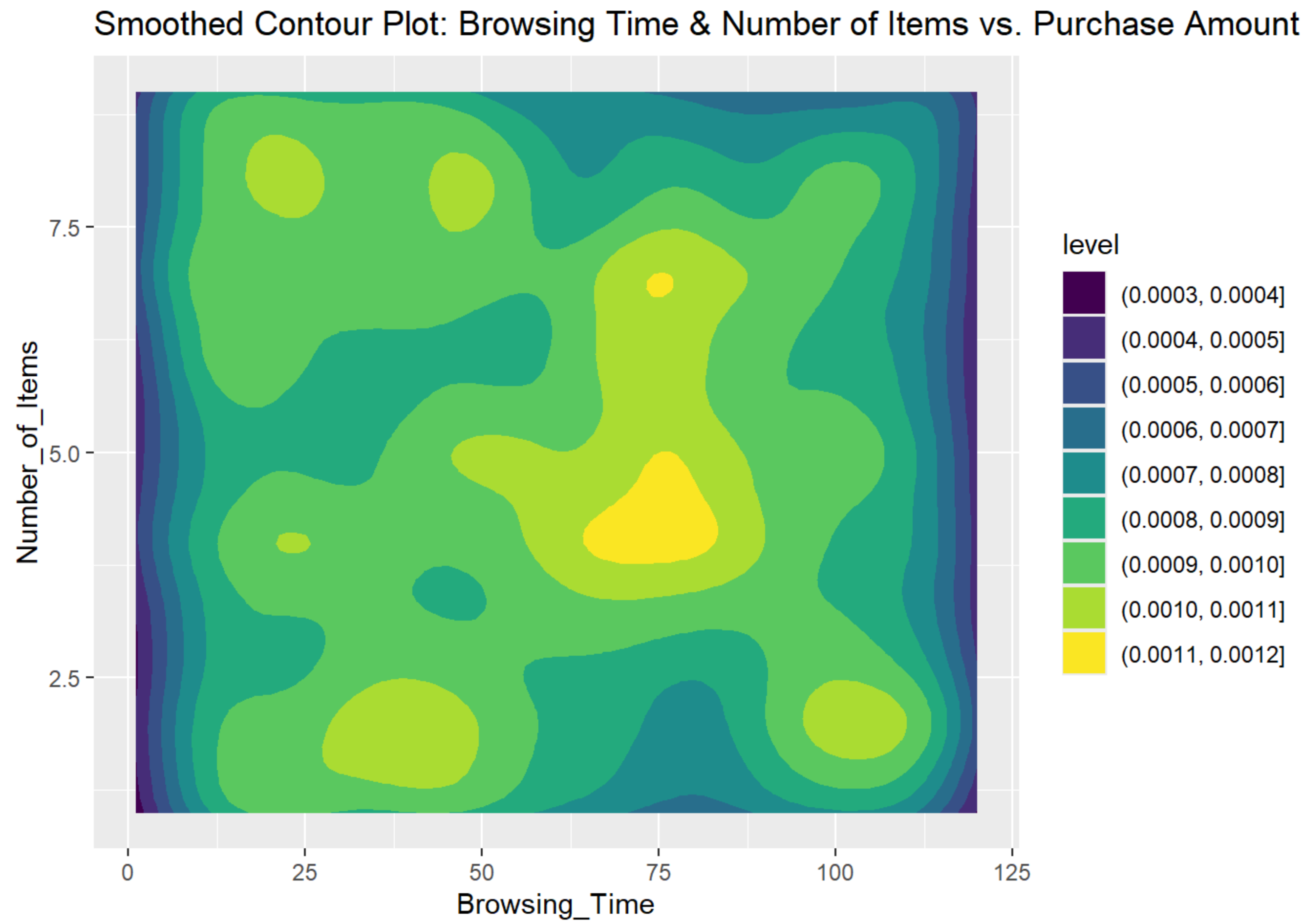
14. Perform multiple regression with Purchase_Amount as the dependent variable and Browsing_Time, Number_of_Items, and Satisfaction_Score as predictors. Perform model selection and assess variable importance.

```
ggplot(df, aes(x = Browsing_Time, y = Number_of_Items, fill = Purchase_Amount)) +
  geom_tile() +
  scale_fill_gradient(low = "blue", high = "red") +
  ggtitle("Level Plot: Browsing Time & Number of Items vs. Purchase Amount")
```




```
ggplot(df, aes(x = Browsing_Time, y = Number_of_Items, z = Purchase_Amount)) +
  geom_density_2d_filled() +
  ggtitle("Smoothed Contour Plot: Browsing Time & Number of Items vs. Purchase Amount")
```

```
## Warning: The following aesthetics were dropped during statistical transformation: z.
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```



From both plots, we can see rhat purchase_amount increases in specific regions of Browsing_Time and Number_of_Items. However, the contour plot failed to give us sufficient analysis, which indicates that **binning methods or heatmaps** are better for visualizing relationships.

15. Perform multiple regression with Purchase_Amount as the dependent variable and Browsing_Time, Number_of_Items, and Satisfaction_Score as predictors. Perform model selection and assess variable importance.

```
multi_model <- lm(Purchase_Amount ~ Browsing_Time + Number_of_Items + Satisfaction_Score, data =
df)
summary(multi_model)
```

```
##
## Call:
## lm(formula = Purchase_Amount ~ Browsing_Time + Number_of_Items +
##     Satisfaction_Score, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -250.668 -120.856   -2.846  118.899  255.664
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        261.34993    9.24929  28.256   <2e-16 ***
## Browsing_Time       -0.07954    0.07504  -1.060    0.289
## Number_of_Items     -0.78321    1.00497  -0.779    0.436
## Satisfaction_Score  -1.53871    1.83444  -0.839    0.402
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140.9 on 2996 degrees of freedom
## Multiple R-squared:  0.0007932,  Adjusted R-squared:  -0.0002073
## F-statistic: 0.7928 on 3 and 2996 DF,  p-value: 0.4978
```

```
lm.beta(multi_model)
```

```
##
## Call:
## lm(formula = Purchase_Amount ~ Browsing_Time + Number_of_Items +
##     Satisfaction_Score, data = df)
##
## Standardized Coefficients::
##        (Intercept)      Browsing_Time    Number_of_Items Satisfaction_Score
##                 NA        -0.01936166        -0.01423912        -0.01532117
```

```
stepwise_model <- step(multi_model, direction = "both")
```

```
## Start:  AIC=29691.89
## Purchase_Amount ~ Browsing_Time + Number_of_Items + Satisfaction_Score
## 
##                      Df Sum of Sq      RSS   AIC
## - Number_of_Items     1     12056 59482958 29691
## - Satisfaction_Score  1     13966 59484867 29691
## - Browsing_Time       1     22299 59493201 29691
## <none>                            59470902 29692
## 
## Step:  AIC=29690.5
## Purchase_Amount ~ Browsing_Time + Satisfaction_Score
## 
##                      Df Sum of Sq      RSS   AIC
## - Satisfaction_Score  1     13479 59496437 29689
## - Browsing_Time       1     21541 59504498 29690
## <none>                            59482958 29691
## + Number_of_Items     1     12056 59470902 29692
## 
## Step:  AIC=29689.18
## Purchase_Amount ~ Browsing_Time
## 
##                      Df Sum of Sq      RSS   AIC
## - Browsing_Time       1     21676 59518113 29688
## <none>                            59496437 29689
## + Satisfaction_Score  1     13479 59482958 29691
## + Number_of_Items     1     11569 59484867 29691
## 
## Step:  AIC=29688.27
## Purchase_Amount ~ 1
## 
##                      Df Sum of Sq      RSS   AIC
## <none>                            59518113 29688
## + Browsing_Time       1     21676 59496437 29689
## + Satisfaction_Score  1     13614 59504498 29690
## + Number_of_Items     1     10822 59507290 29690
```

```
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = Purchase_Amount ~ 1, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -242.933 -119.268   -2.873  119.237  251.647
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  247.963      2.572   96.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140.9 on 2999 degrees of freedom
```

From the multiple regression model, we can infer that **Browsing_Time** alone is a weak predictor of **Purchase_Amount**. The **Number_of_Items** and **Satisfaction_Score** have stronger effects. Stepwise regression helps identify the most important variables.