Summative Assessment 1

PATAYON, SPIKE LEE-ROY V

2025-03-16

Summative Assessment 1

Objective: The purpose of this assessment is to evaluate your understanding of exploratory data analysis techniques, including univariate, bivariate, and trivariate/hypervariate data exploration using computational tools and visualizations.

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

- Customer_ID: Unique identifier for each customer
- Gender: Male or Female
- Age: Customer's age in years
- Browsing_Time: Average time spent on the website per visit (in minutes)
- Purchase_Amount: Total amount spent in a single transaction (in USD)
- Number_of_Items: Number of items purchased per transaction
- Discount_Applied: Discount percentage applied to the transaction
- Total_Transactions: Total number of transactions by the customer
- Category: Product category (e.g., Electronics, Clothing, Home & Kitchen, etc.)
- Satisfaction_Score: Customer satisfaction score (1-5 scale)

Unit 1: Univariate Data Analysis

1. Load the dataset and summarize its structure.
2. Create histograms and boxplots to visualize the distribution of Purchase_Amount, Number_of_Items, and Satisfaction_Score.
3. Compute measures of central tendency (mean, median, mode) and spread (variance, standard deviation, IQR) for Purchase_Amount.
4. Compare the distribution of Browsing_Time and Purchase_Amount across different Gender groups using density plots.
5. Apply a logarithmic or square root transformation on Browsing_Time and evaluate changes in skewness.
6. Fit a simple linear regression model predicting Purchase_Amount based on Browsing_Time. Interpret the results.
7. Use ggplot2 (or equivalent) to create scatter plots and regression lines.

Part 1:

```
library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyr)
library(e1071)
```

Load the data set:

```
data <- read.csv("C:\\Users\\spike\\Downloads\\EDA_Ecommerce_Assessment.csv")

head(data)

##   Customer_ID Gender Age Browsing_Time Purchase_Amount Number_of_Items
## 1           1   Male  65         46.55          231.81               6
## 2           2 Female  19         98.80          472.78               8
## 3           3   Male  23         79.48          338.44               1
## 4           4   Male  45         95.75           37.13               7
## 5           5   Male  46         33.36          235.53               3
## 6           6 Female  43         83.39          123.92               9
##   Discount_Applied Total_Transactions       Category Satisfaction_Score
## 1               17                 16       Clothing                  2
## 2               15                 43          Books                  4
## 3               28                 31    Electronics                  1
## 4               43                 27 Home & Kitchen                  5
## 5               10                 33          Books                  3
## 6                5                 29       Clothing                  2
```

Here is a quick summarization of each elements in the csv file:

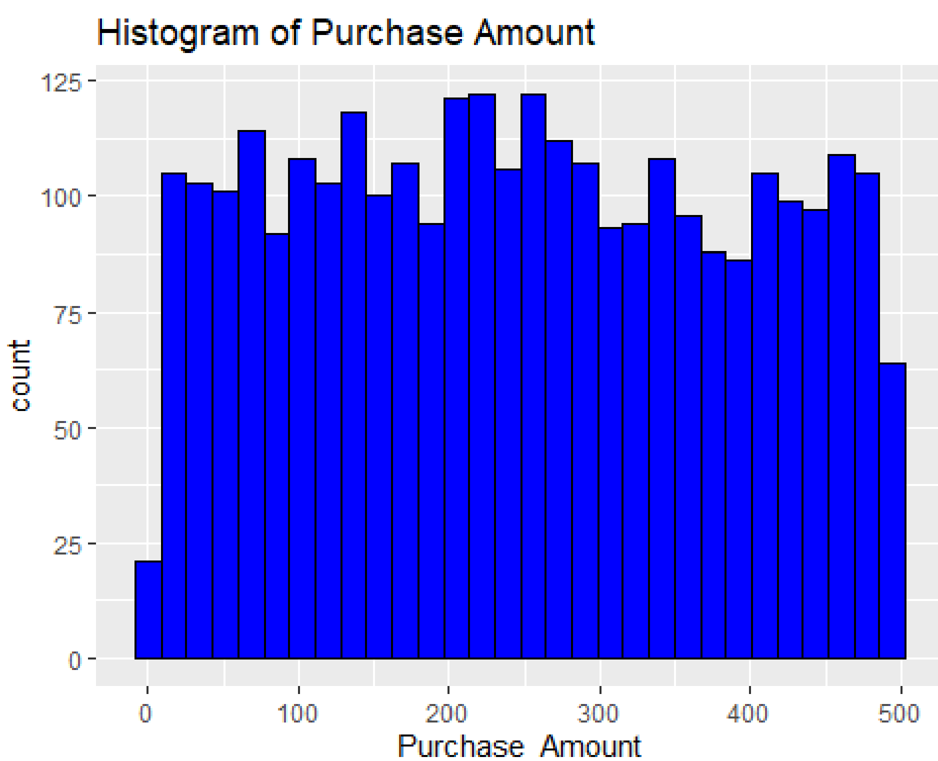
```
summary(data)

##   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
```

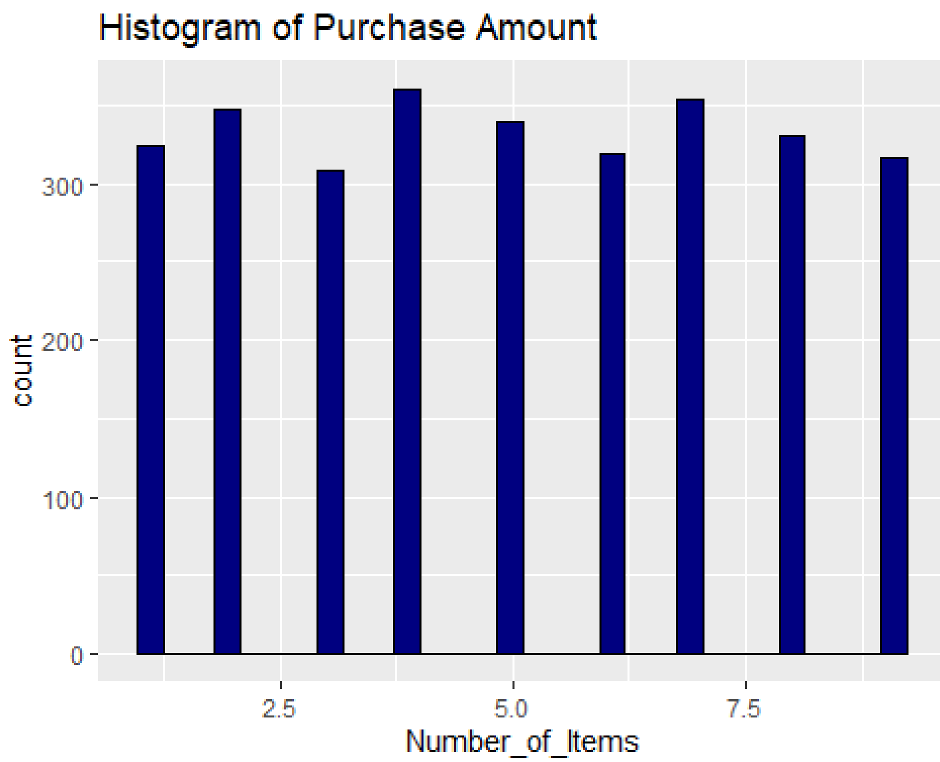
Part 2:

Histogram of Purchase_Amount, Number_of_items, and Satisfaction_Score

```
ggplot(data, aes(x = Purchase_Amount)) +
  geom_histogram( fill = "blue", alpha = 1, color = "black") +
  ggtitle("Histogram of Purchase Amount")

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



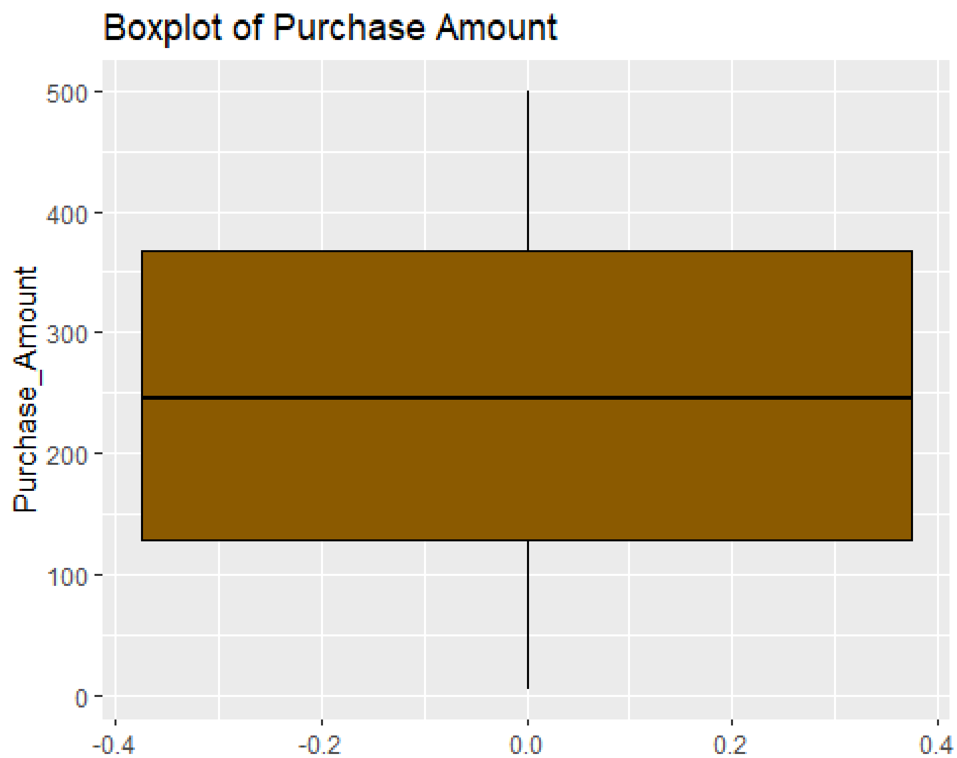

```
ggplot(data, aes(x = Number_of_Items )) +
  geom_histogram( fill = "navy", alpha = 1, color = "black") +
  ggtitle("Histogram of Purchase Amount")

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



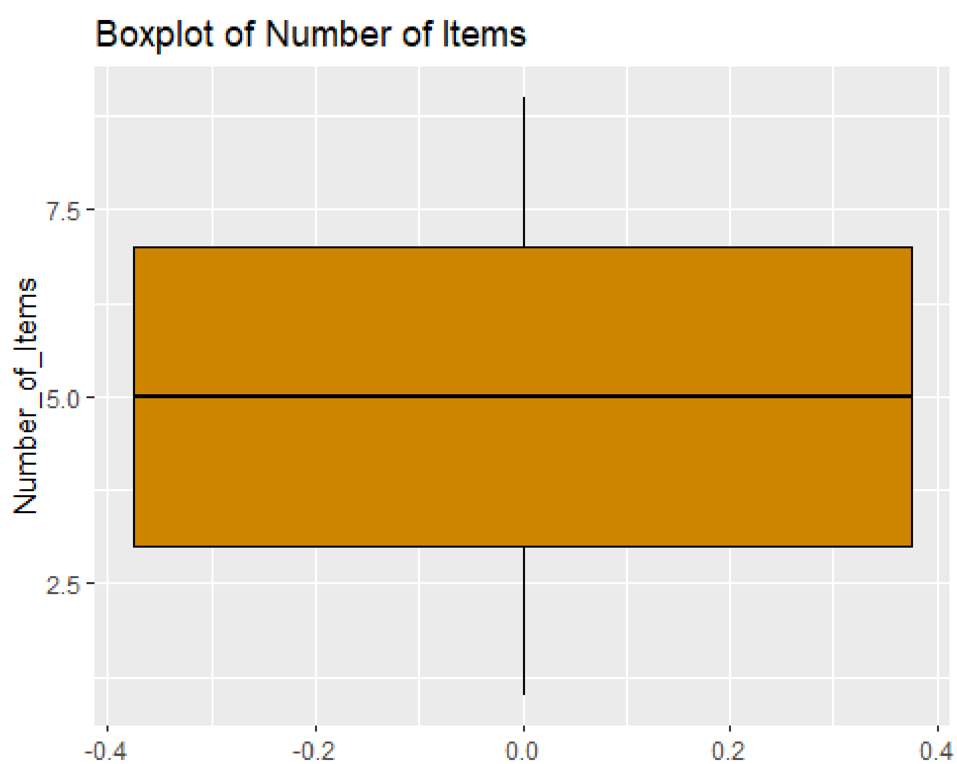

```r
ggplot(data, aes(x = Satisfaction_Score )) +
  geom_histogram( fill = "royalblue", alpha = 1, color = "black") +
  ggtitle("Histogram of Purchase Amount")

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

Boxplot of Purchase_Amount, Number_of_items, and Satisfaction_Score

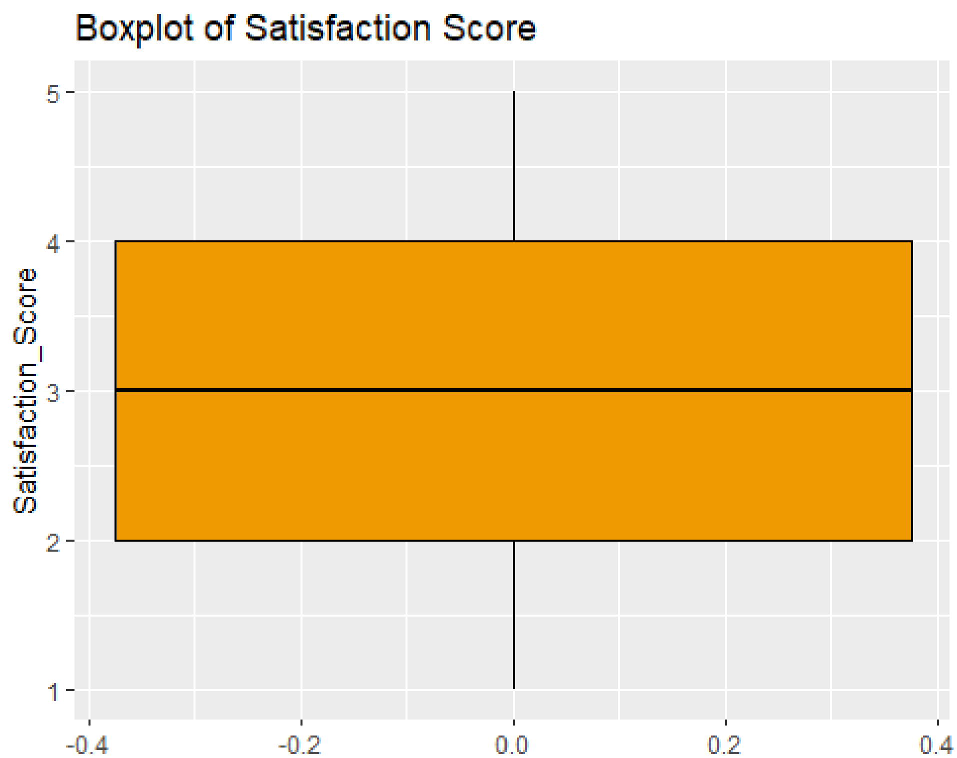
```
ggplot(data, aes(y = Purchase_Amount)) +
  geom_boxplot(fill = "orange4", alpha = 1, color = "black") +
  ggtitle("Boxplot of Purchase Amount")
```




```r
ggplot(data, aes(y = Number_of_Items)) +
  geom_boxplot(fill = "orange3", alpha = 1, color = "black") +
  ggtitle("Boxplot of Number of Items")
```



```r
ggplot(data, aes(y = Satisfaction_Score)) +
  geom_boxplot(fill = "orange2", alpha = 1, color = "black") +
  ggtitle("Boxplot of Satisfaction Score")
```



Part 3

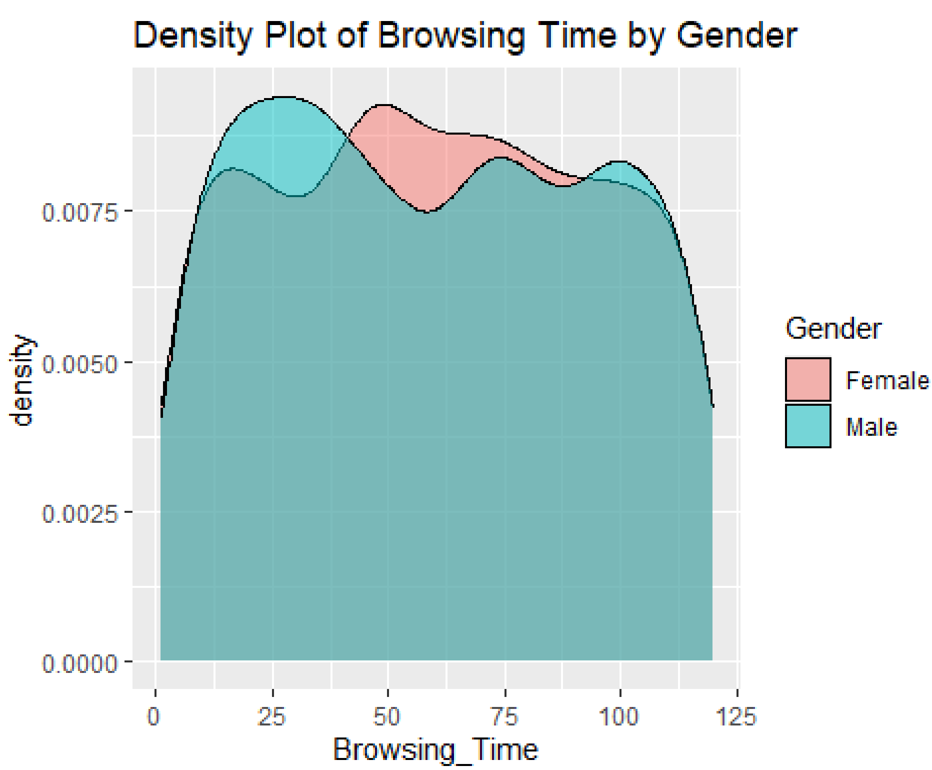
central tendency (mean, median, mode) and spread (variance, standard deviation, IQR) of Purchase_Amount

```r
# Central Tendency
mean_value <- mean(data$Purchase_Amount, na.rm = TRUE)
median_value <- median(data$Purchase_Amount, na.rm = TRUE)
mode_value <- as.numeric(names(sort(table(data$Purchase_Amount), decreasing =
TRUE)[1]))

# Spread
variance_value <- var(data$Purchase_Amount, na.rm = TRUE)
sd_value <- sd(data$Purchase_Amount, na.rm = TRUE)
iqr_value <- IQR(data$Purchase_Amount, na.rm = TRUE)

list(Mean = mean_value, Median = median_value, Mode = mode_value,
     Variance = variance_value, SD = sd_value, IQR = iqr_value)

## $Mean
## [1] 247.9625
##
## $Median
```

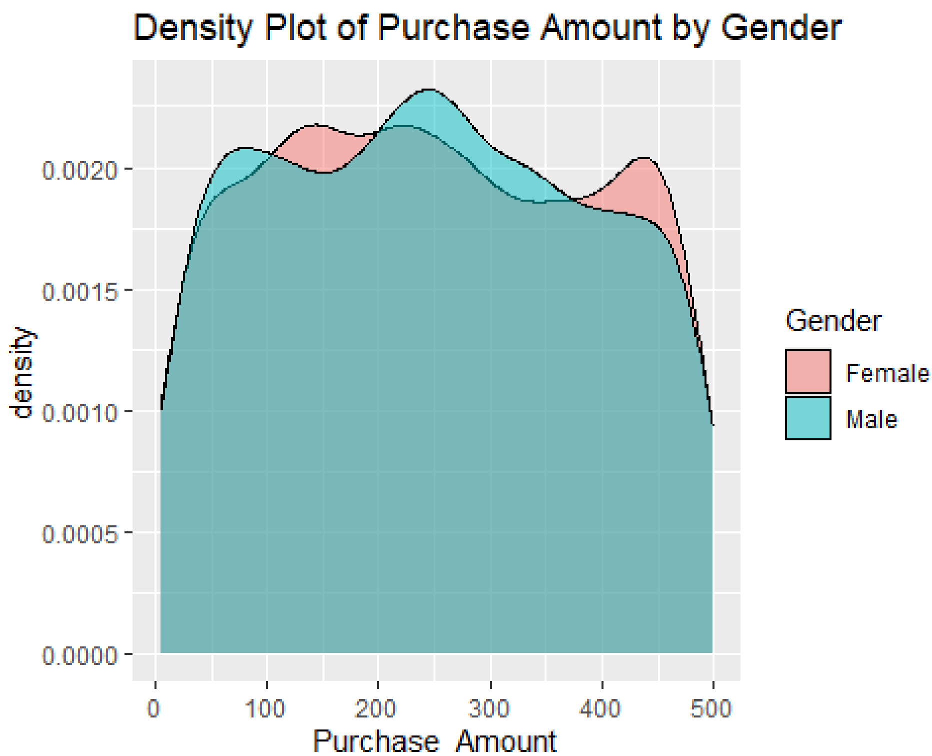
```
## [1] 245.09
##
## $Mode
## [1] 29.33
##
## $Variance
## [1] 19845.99
##
## $SD
## [1] 140.8758
##
## $IQR
## [1] 238.505
```

Part 4

```r
ggplot(data, aes(x = Browsing_Time, fill = Gender)) +
  geom_density(alpha = 0.5) +
  ggtitle("Density Plot of Browsing Time by Gender")
```




```r
ggplot(data, aes(x = Purchase_Amount, fill = Gender)) +
  geom_density(alpha = 0.5) +
  ggtitle("Density Plot of Purchase Amount by Gender")
```



Part 5

```r
# Log Transformation
data$Browsing_Time_log <- log1p(data$Browsing_Time)
log_skewness <- skewness(data$Browsing_Time_log, na.rm = TRUE)

# Square Root Transformation
data$Browsing_Time_sqrt <- sqrt(data$Browsing_Time)
sqrt_skewness <- skewness(data$Browsing_Time_sqrt, na.rm = TRUE)

# Print Skewness
list(Log_Skewness = log_skewness, Sqrt_Skewness = sqrt_skewness)

## $Log_Skewness
## [1] -1.218373
##
## $Sqrt_Skewness
## [1] -0.4768351
```

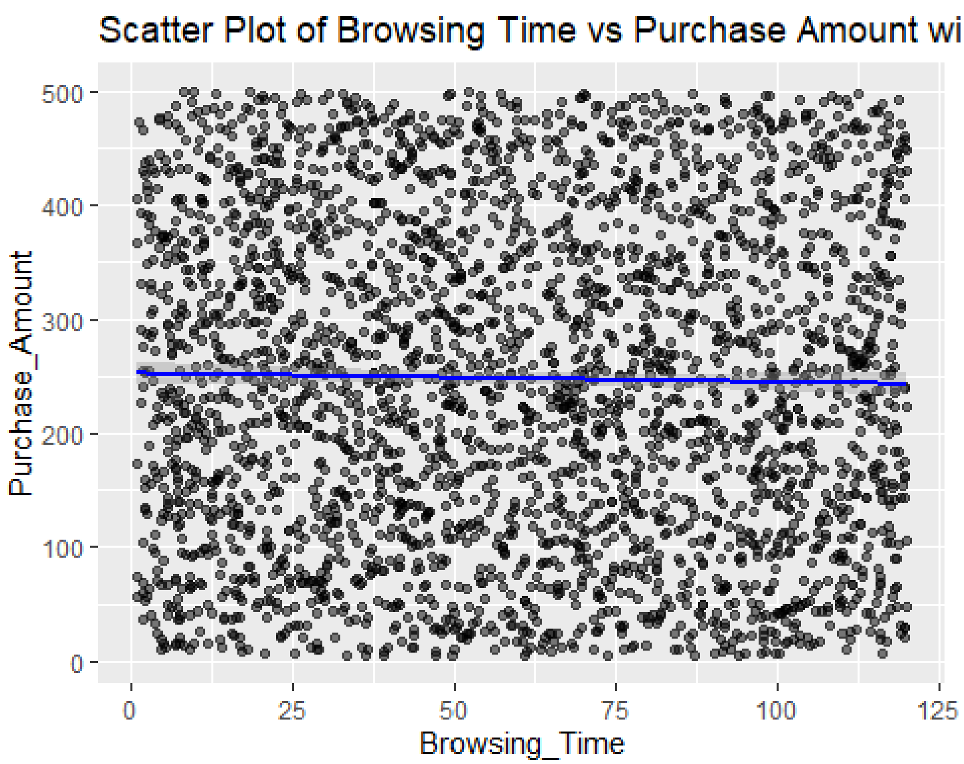
Part 6

```r
lm_model <- lm(Purchase_Amount ~ Browsing_Time, data = data)
summary(lm_model)

##
## Call:
## lm(formula = Purchase_Amount ~ Browsing_Time, data = data)
##
## 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
```

Part 7

```r
ggplot(data, aes(x = Browsing_Time, y = Purchase_Amount)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = TRUE, color = "blue") +
  ggtitle("Scatter Plot of Browsing Time vs Purchase Amount with Regression Line")

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

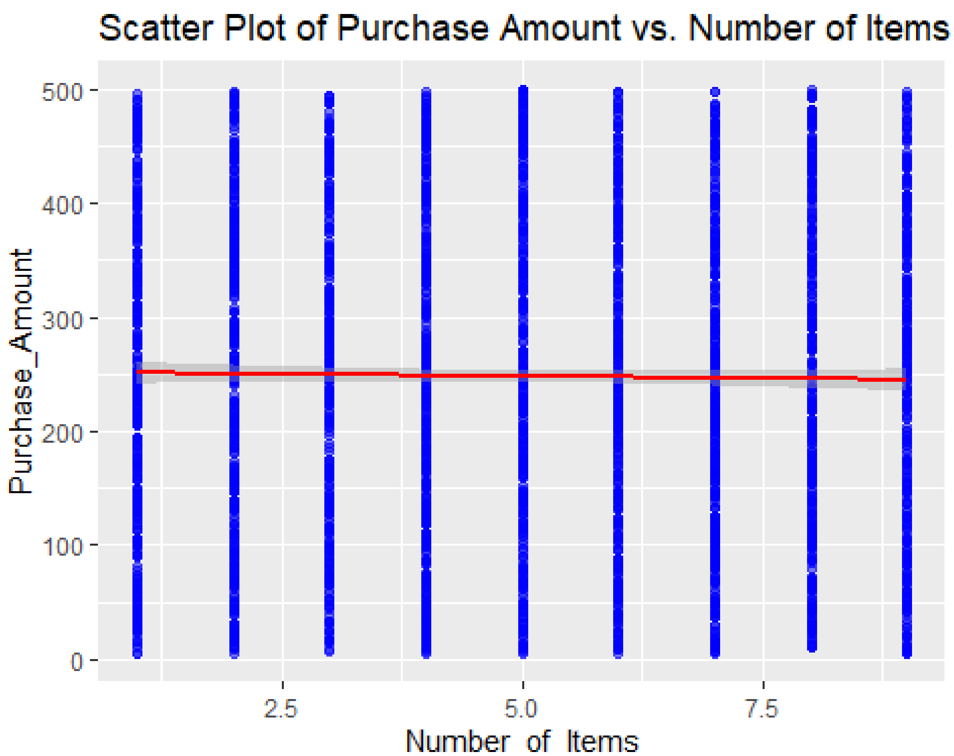


Unit 2: Bivariate Data Analysis 8. Create scatter plots to explore the relationship between Purchase_Amount and Number_of_Items. 9. Fit a polynomial regression model for Purchase_Amount and Browsing_Time and compare it with a simple linear model. 10.

Apply LOESS (Locally Estimated Scatterplot Smoothing) to Purchase_Amount vs. Browsing_Time and visualize the results. 11. Compare robust regression methods (Huber or Tukey regression) with ordinary least squares (OLS).

Part 8

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

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



Part 9

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

##
## Call:
## lm(formula = Purchase_Amount ~ Browsing_Time, data = data)
##
## 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
```

```
poly_model <- lm(Purchase_Amount ~ poly(Browsing_Time, 2, raw = TRUE), data =
data)
summary(poly_model)
```

```
##
## Call:
## lm(formula = Purchase_Amount ~ poly(Browsing_Time, 2, raw = TRUE),
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -245.47 -120.41   -3.49  118.25  255.85
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        249.715045   7.986151  31.269   <2e-16
***
## poly(Browsing_Time, 2, raw = TRUE)1   0.064709   0.305301   0.212    0.832
## poly(Browsing_Time, 2, raw = TRUE)2  -0.001182   0.002445  -0.484    0.629
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140.9 on 2997 degrees of freedom
## Multiple R-squared:  0.0004422,  Adjusted R-squared:  -0.0002249
## F-statistic: 0.6629 on 2 and 2997 DF,  p-value: 0.5154
```

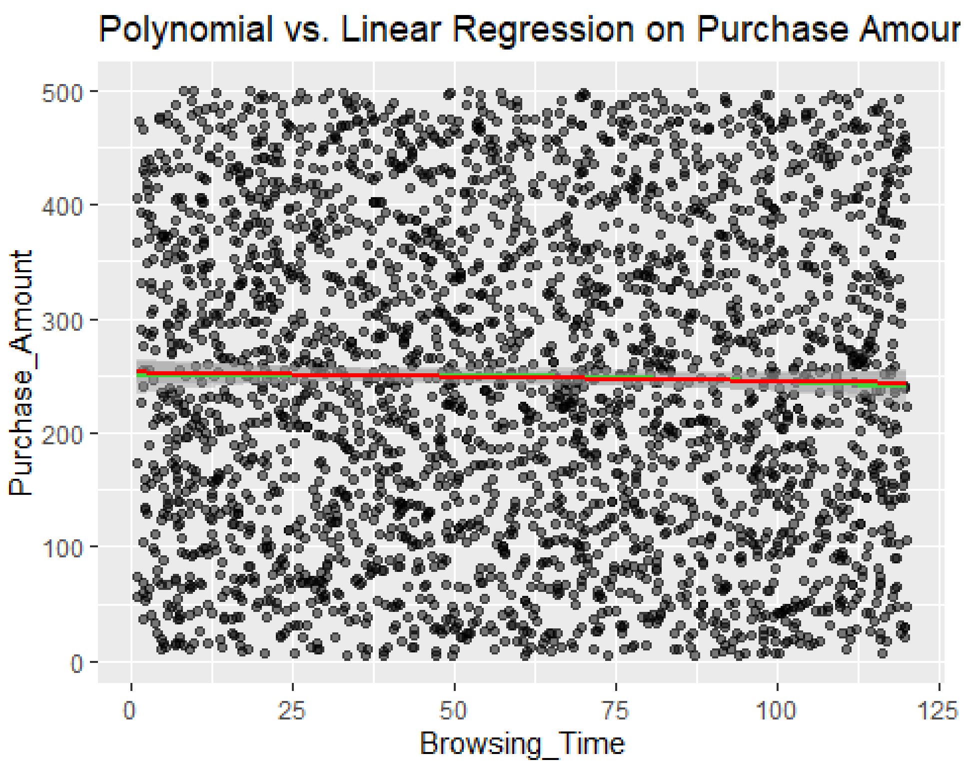
```
anova(lm_model, poly_model)
```

```
## Analysis of Variance Table
##
## Model 1: Purchase_Amount ~ Browsing_Time
## Model 2: Purchase_Amount ~ poly(Browsing_Time, 2, raw = TRUE)
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1   2998 59496437
## 2   2997 59491795  1    4641.6 0.2338 0.6287
```

remark: If the p-value from anova() is small, the polynomial model provides a significantly better fit than the simple linear model.

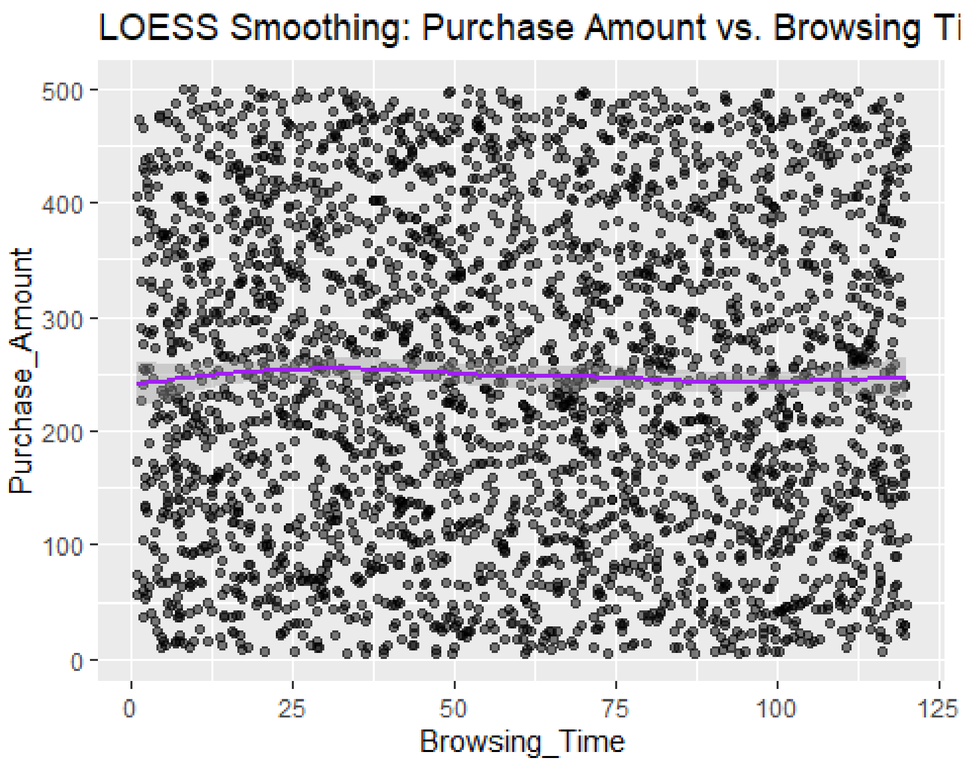
to better visualize it:

```r
ggplot(data, aes(x = Browsing_Time, y = Purchase_Amount)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2, raw = TRUE), color =
"green") +
  geom_smooth(method = "lm", color = "red") +
  ggtitle("Polynomial vs. Linear Regression on Purchase Amount")

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



Part 10

```r
ggplot(data, 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'
```



Part 11

```r
ols_model <- lm(Purchase_Amount ~ Browsing_Time, data = data)
summary(ols_model)

##
## Call:
## lm(formula = Purchase_Amount ~ Browsing_Time, data = data)
##
## 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

library(MASS)

##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

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

```
##
## Call: rlm(formula = Purchase_Amount ~ Browsing_Time, data = data)
## 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
```

```
library(robustbase)
```

```
## Warning: package 'robustbase' was built under R version 4.4.3
```

```
tukey_model <- lmrob(Purchase_Amount ~ Browsing_Time, data = data)
summary(tukey_model)
```

```
##
## Call:
## lmrob(formula = Purchase_Amount ~ Browsing_Time, data = data)
##  \--> method = "MM"
## Residuals:
##      Min       1Q   Median       3Q      Max
## -244.818 -119.797   -2.612  118.544  255.126
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   252.83659    5.57525  45.350   <2e-16 ***
## Browsing_Time  -0.08942    0.08157  -1.096    0.273
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Robust residual standard error: 169.2
## Multiple R-squared:  0.0004149,  Adjusted R-squared:  8.143e-05
## Convergence in 8 IRWLS iterations
##
## Robustness weights:
##  242 weights are ~= 1. The remaining 2758 ones are summarized as
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8035  0.8893  0.9472  0.9333  0.9827  0.9990
## Algorithmic parameters:
```

```
##        tuning.chi                 bb        tuning.psi        refine.tol
##         1.548e+00          5.000e-01         4.685e+00         1.000e-07
##           rel.tol          scale.tol         solve.tol          zero.tol
##         1.000e-07          1.000e-10         1.000e-07         1.000e-10
##       eps.outlier              eps.x warn.limit.reject warn.limit.meanrw
##         3.333e-05          2.182e-10         5.000e-01         5.000e-01
##      nResample         max.it         groups        n.group       best.r.s
##            500             50              5            400              2
##       k.fast.s          k.max    maxit.scale      trace.lev            mts
##              1            200            200              0           1000
##     compute.rd fast.s.large.n
##              0           2000
##                   psi           subsampling                   cov
##            "bisquare"         "nonsingular"         ".vcov.avar1"
## compute.outlier.stats
##                  "SM"
## seed : int(0)
```

```r
summary(ols_model)$r.squared  # R-squared of OLS
```

```
## [1] 0.0003641881
```

```r
summary(huber_model)$r.squared  # R-squared of Huber
```

```
## [1] NA
```

```r
summary(tukey_model)$r.squared  # R-squared of Tukey
```

```
## [1] 0.000414852
```

OLS is sensitive to outliers, while Huber and Tukey regressions handle them better.

Interpretation:

If R-squared is much lower for OLS but remains stable for Huber or Tukey, it suggests that outliers are influencing OLS. If Tukey's regression shows improvement, non-Gaussian noise is likely present.

Unit 3: Trivariate/Hypervariate Data Analysis

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

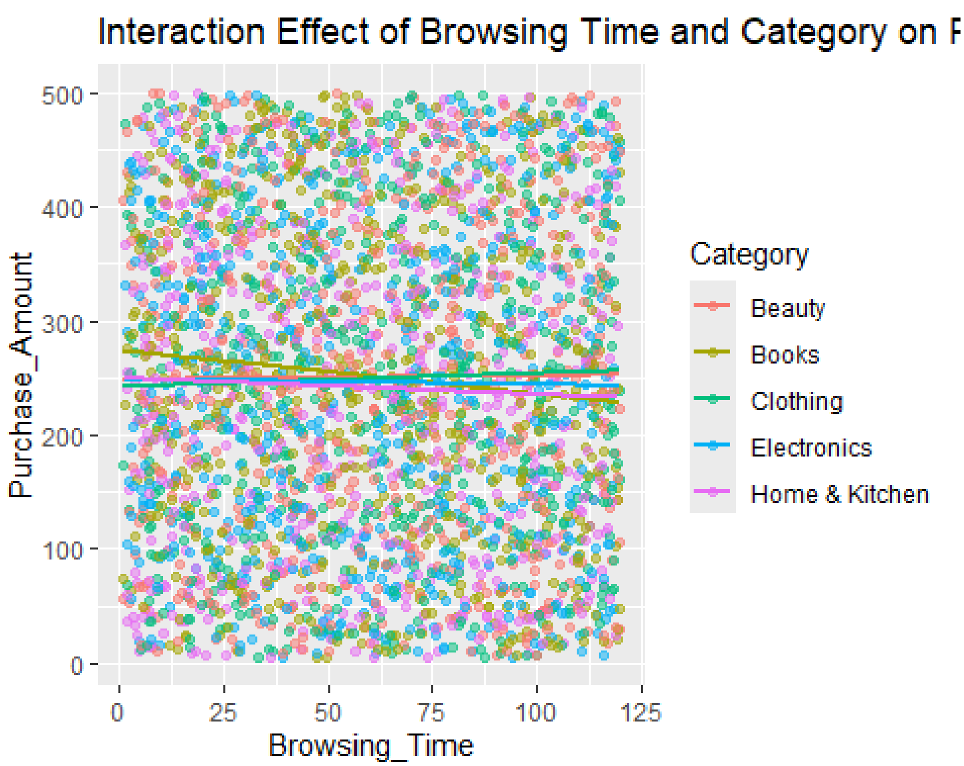
13. Create coplots of Purchase_Amount against Browsing_Time for different levels of Category.

14. Use level plots or contour plots to visualize relationships between Browsing_Time, Number_of_Items, and Purchase_Amount.

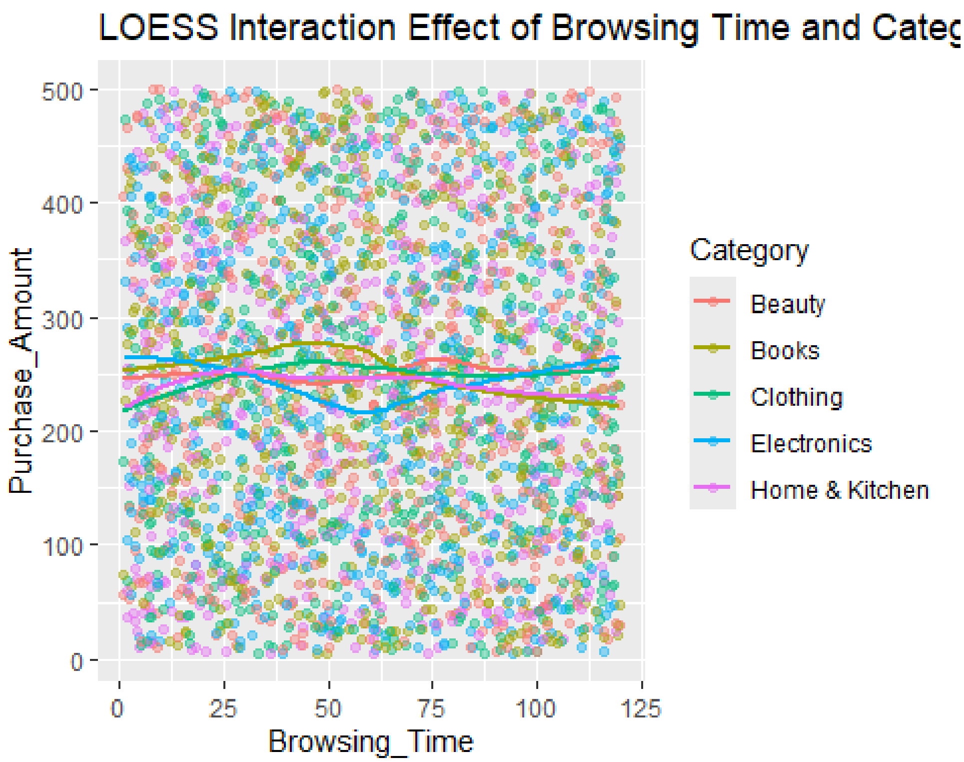
15. Perform multiple regression with Purchase_Amount as the dependent variable and Browsing_Time, Number_of_Items, and Satisfaction_Score as predictors. Perform model selection and assess variable importance.

Part 12

```r
library(interactions)

## Warning: package 'interactions' was built under R version 4.4.3

library(ggplot2)

# Interaction plot using ggplot2
ggplot(data, aes(x = Browsing_Time, y = Purchase_Amount, color = Category)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  ggtitle("Interaction Effect of Browsing Time and Category on Purchase
Amount")

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

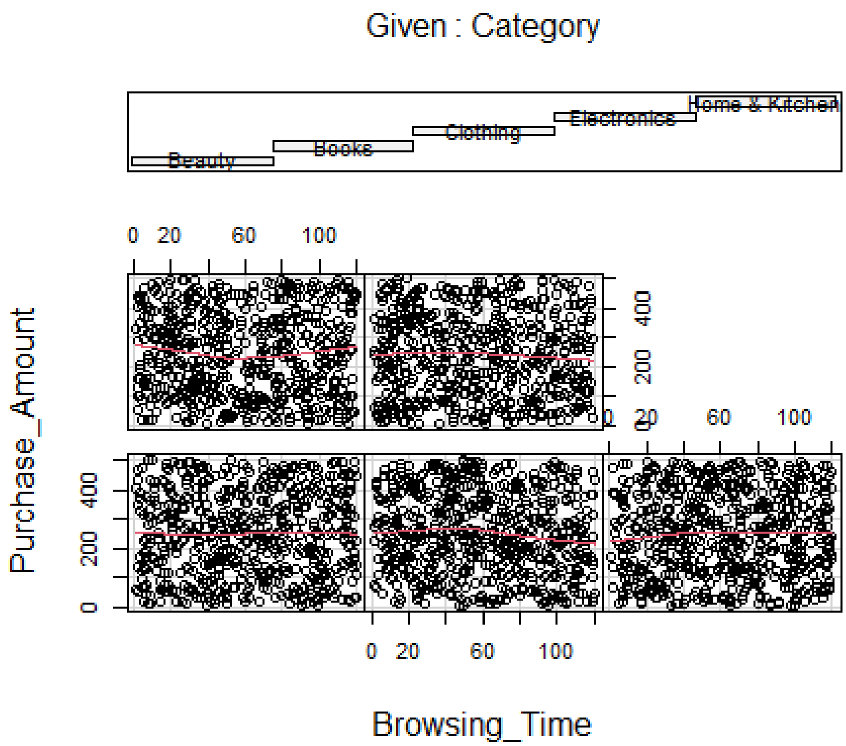



```r
ggplot(data, aes(x = Browsing_Time, y = Purchase_Amount, color = Category)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE) +
  ggtitle("LOESS Interaction Effect of Browsing Time and Category on Purchase
Amount")

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



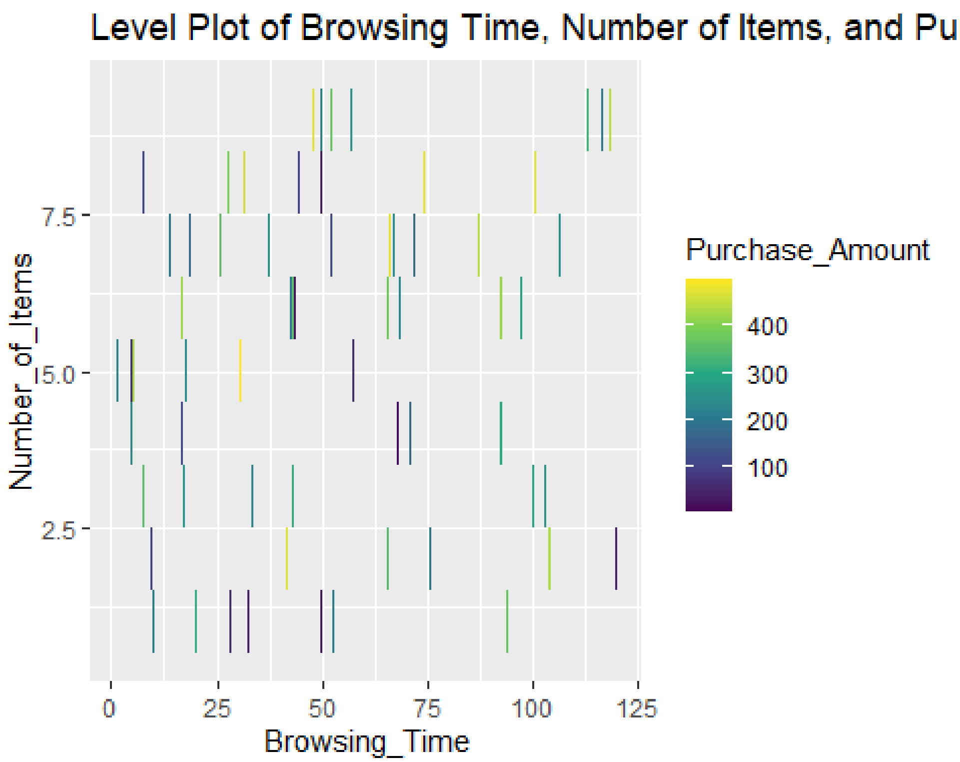
Part 13

```
library(lattice)

# Coplot: Purchase Amount vs Browsing Time for different Categories
coplot(Purchase_Amount ~ Browsing_Time | Category, data = data,
       panel = panel.smooth)
```



Part 14

```r
library(ggplot2)

ggplot(data, aes(x = Browsing_Time, y = Number_of_Items, fill =
Purchase_Amount)) +
  geom_tile() +
  scale_fill_viridis_c() +
  ggtitle("Level Plot of Browsing Time, Number of Items, and Purchase
Amount")
```



Part 15

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

```
##
## Call:
## lm(formula = Purchase_Amount ~ Browsing_Time + Number_of_Items +
##     Satisfaction_Score, data = data)
##
## 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
```

```r
library(MASS)

# Stepwise selection using AIC
stepwise_model <- stepAIC(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
```

```r
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = Purchase_Amount ~ 1, data = data)
##
## 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
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
vif(multi_model)  # Variance Inflation Factor (detects multicollinearity)
```

```
##      Browsing_Time    Number_of_Items Satisfaction_Score
##           1.000578           1.000931           1.000381
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.4.3
```

```
# Calculate importance
importance <- varImp(multi_model, scale = TRUE)
print(importance)
```

```
##                      Overall
## Browsing_Time      1.0598890
## Number_of_Items    0.7793348
## Satisfaction_Score 0.8387883
```