in collaboration with

STW7089CEM: Introduction to Statistical Methods for Data Science

Submitted By:

Ritesh Rokaha (14825217)

Contents

1. Introduction

The phenomena of customer purchasing is complex and influences not only the individual preferences and actions of consumers but also the whole dynamics of retail markets. It includes a wide range of tasks and procedures necessary for obtaining products or services, from preliminary product assessment and browsing to the last step of completing transactions. Customers make decisions based on a range of criteria within this continuum, including product features, pricing schemes, marketing initiatives, and general shopping experiences. Businesses hoping to remain competitive in today's market must grasp the nuances of customer shopping behavior. Businesses can improve total customer happiness by customizing their marketing strategies, optimizing product offers, and gathering insights into the motives, preferences, and habits of consumers. The development of successful business strategies targeted at satisfying the various requirements and expectations of consumers is based on this thorough understanding (Bellenger et al., 1978; Dholakia, 1999; Holbrook, 1994; Jones et al., 2006; Sproles & Kendall, 1986).

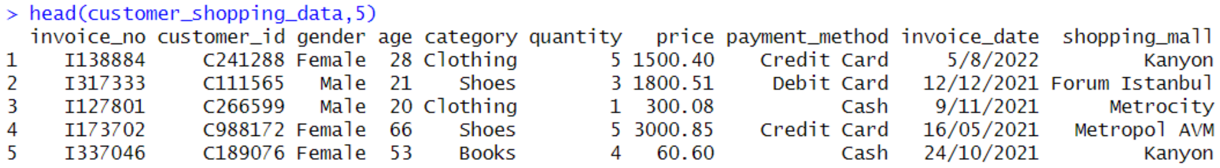
The customer shopping dataset was used for this study to fulfil the project objectives.

2. Methodology

To fulfil the project objective the customer shopping dataset was used for this study. All the analysis such as, data importing, data preprocessing, data converting, regression analysis, plots were conducted under the R-language programming.

2.1 Data Description

The dataset was collected from the 10 different malls in Istanbul from 2021 to 2023. The invoice_no, customer_id, gender, age, category, quantity, price, payment_method, invoice_date, and shopping_mall variables were used which were asked from respondents to collect the data. Where the age, price, quantity was the continues variables and payment method, category, gender, shopping mall were the categorical variables. All the variables have 99457 values inside the variable. The detail view of the dataset given in Figure 1.



```
> head(customer_shopping_data,5)
  invoice_no customer_id gender age category quantity   price payment_method invoice_date  shopping_mall
1    I138884     C241288 Female  28 Clothing        5 1500.40    Credit Card     5/8/2022         Kanyon
2    I317333     C111565   Male  21    Shoes        3 1800.51     Debit Card   12/12/2021 Forum Istanbul
3    I127801     C266599   Male  20 Clothing        1  300.08           Cash    9/11/2021      Metrocity
4    I173702     C988172 Female  66    Shoes        5 3000.85    Credit Card   16/05/2021   Metropol AVM
5    I337046     C189076 Female  53    Books        4   60.60           Cash   24/10/2021         Kanyon
```

Figure 1. Imported customer shopping dataset

2.2 Data cleaning and handling

The customer dataset was import in R-Studio using the “read.csv” function. The missing values of the customer shopping dataset was check and presenting in Figure 2. The Figure 2 explained that there is no problem of missing values exit in our customer shopping dataset.

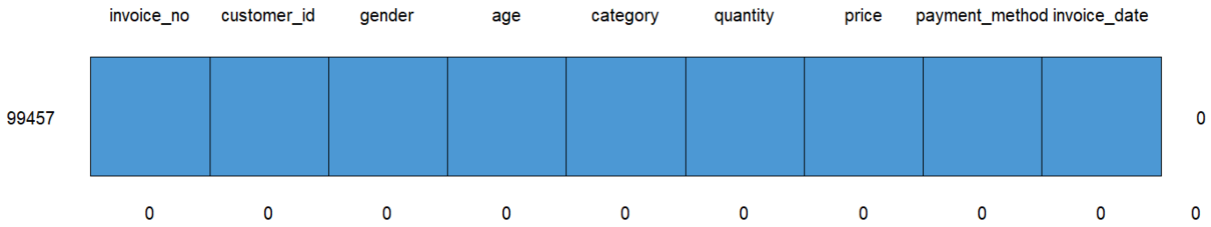
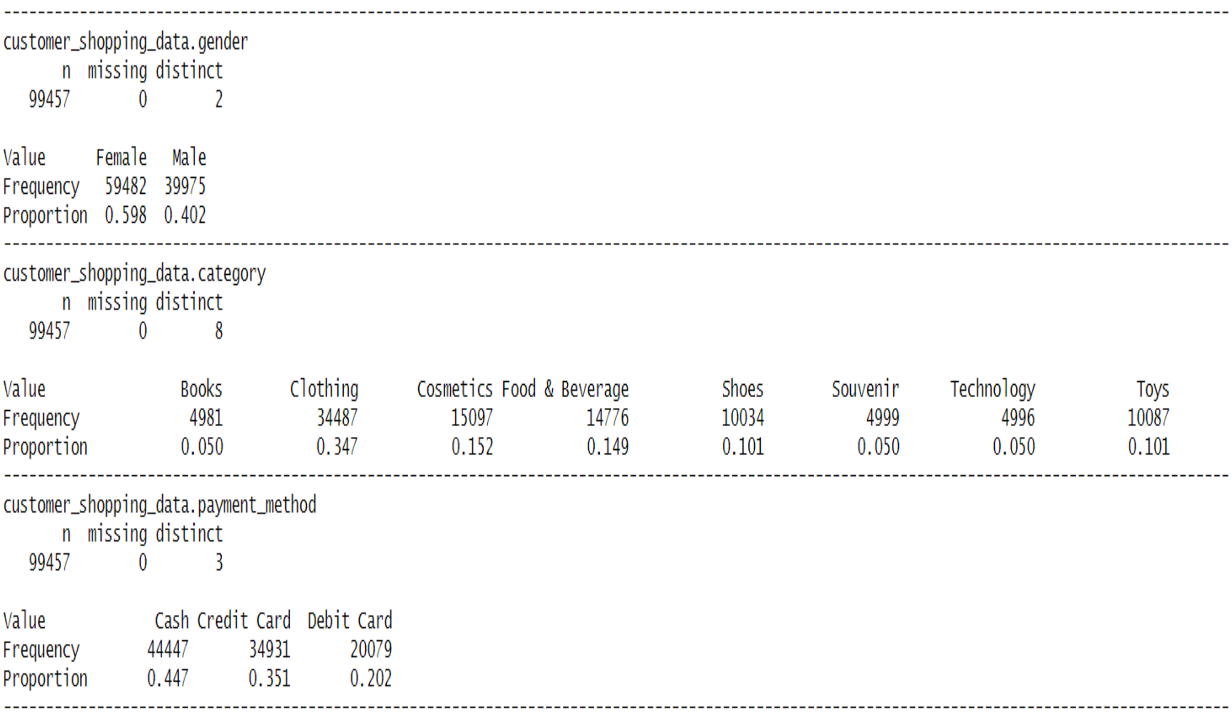


Figure 2: Presenting the missing values in our dataset.

2.2.1 Frequency Distribution of categorical variables

The frequency distribution of categorical variables are given in Figure 3. There are total 59482 females and 39975 males are used as respondent to collect data. Similarly, category of shopping also presenting, and payment method also. The 44447 people used cash method, 34931 used credit card method, 20079 used debit card method.



```
--------------------------------------------------------------------------------------------------------------------------------
customer_shopping_data.gender
       n  missing distinct
   99457        0        2

Value      Female   Male
Frequency   59482  39975
Proportion  0.598  0.402
--------------------------------------------------------------------------------------------------------------------------------
customer_shopping_data.category
       n  missing distinct
   99457        0        8

Value                 Books       Clothing       Cosmetics Food & Beverage          Shoes        Souvenir      Technology           Toys
Frequency              4981          34487           15097           14776          10034            4999            4996          10087
Proportion            0.050          0.347           0.152           0.149          0.101           0.050           0.050          0.101
--------------------------------------------------------------------------------------------------------------------------------
customer_shopping_data.payment_method
       n  missing distinct
   99457        0        3

Value            Cash Credit Card  Debit Card
Frequency       44447       34931       20079
Proportion      0.447       0.351       0.202
--------------------------------------------------------------------------------------------------------------------------------
```

Figure 3: Presenting the Frequency Distribution of categorical variables.

4. Results

Task 1

Task 1.1: Time series plots (of input and output of customer shopping data)

Analyzing time series data entails looking at information gathered over a number of intervals. It assists in identifying dependencies, patterns, and trends in the dataset—all of which are essential for forecasting and making decisions. It makes use of methods like autocorrelation and decomposition and is widely used in disciplines like economics, finance, and weather forecasting. In time series analysis, the ARIMA model is a widely used forecasting method (Hyndman & Athanasopoulos, 2018).

The Figure 4. Presenting the time series plot of the input(x) variables regarding to the invoice_date. The age, price, quantity variables presenting that as much as date increase the product purchasing quantity decreased.

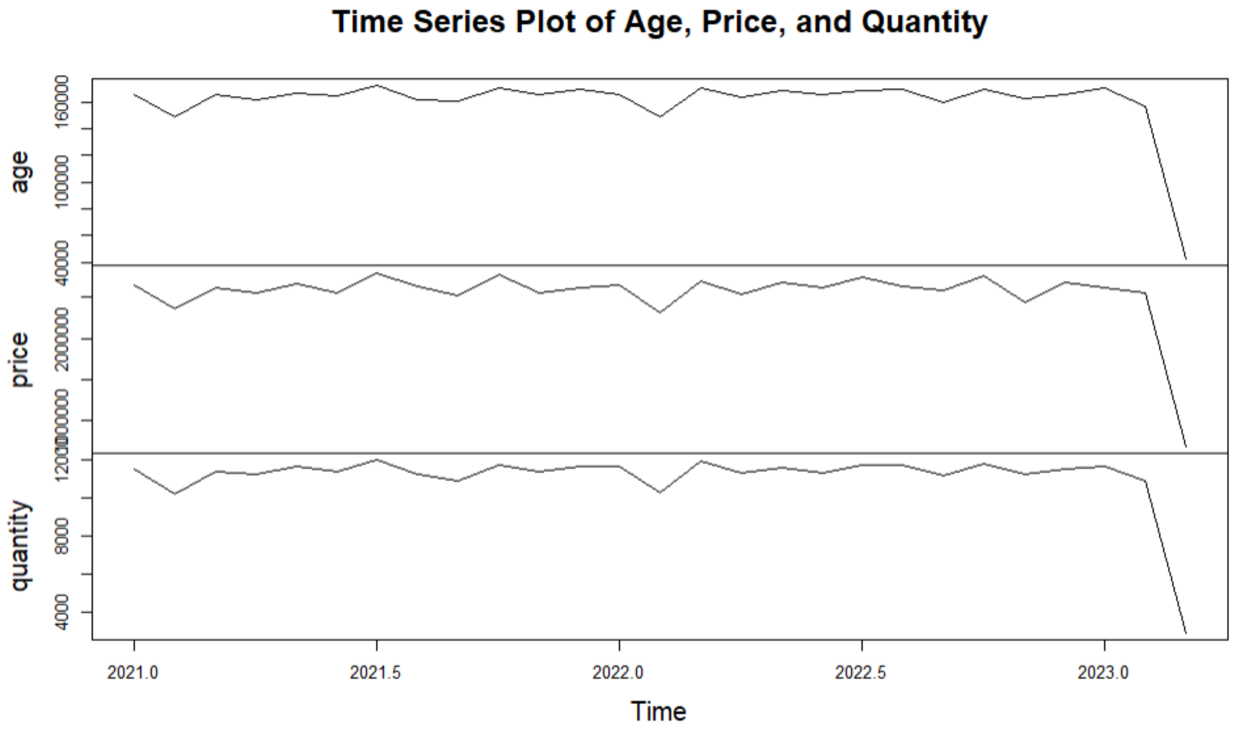


Figure 4. Time series plot of input(x) variables.

Similarly the time series plot was also conduct for the output(y) variable which is presenting in Figure 5. The Total sales quantity variables presenting that as much as date increase the product purchasing quantity decreased.

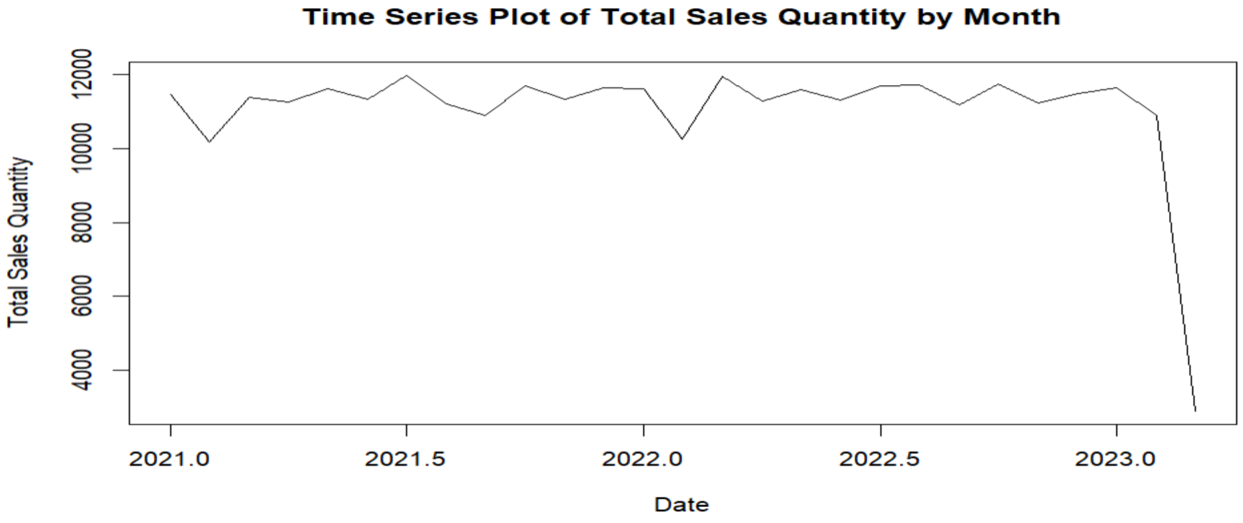


Figure 5. Time series plot of output(y) variables.

Task 1.2: Distribution for each sales data

The distribution of continuous variables were plotted using the "par" function in R. The distribution plot of age, price, quantity presenting in Figure 6. Where distribution of age presenting with blue line, for price presenting with red line, and for quantity with green line.

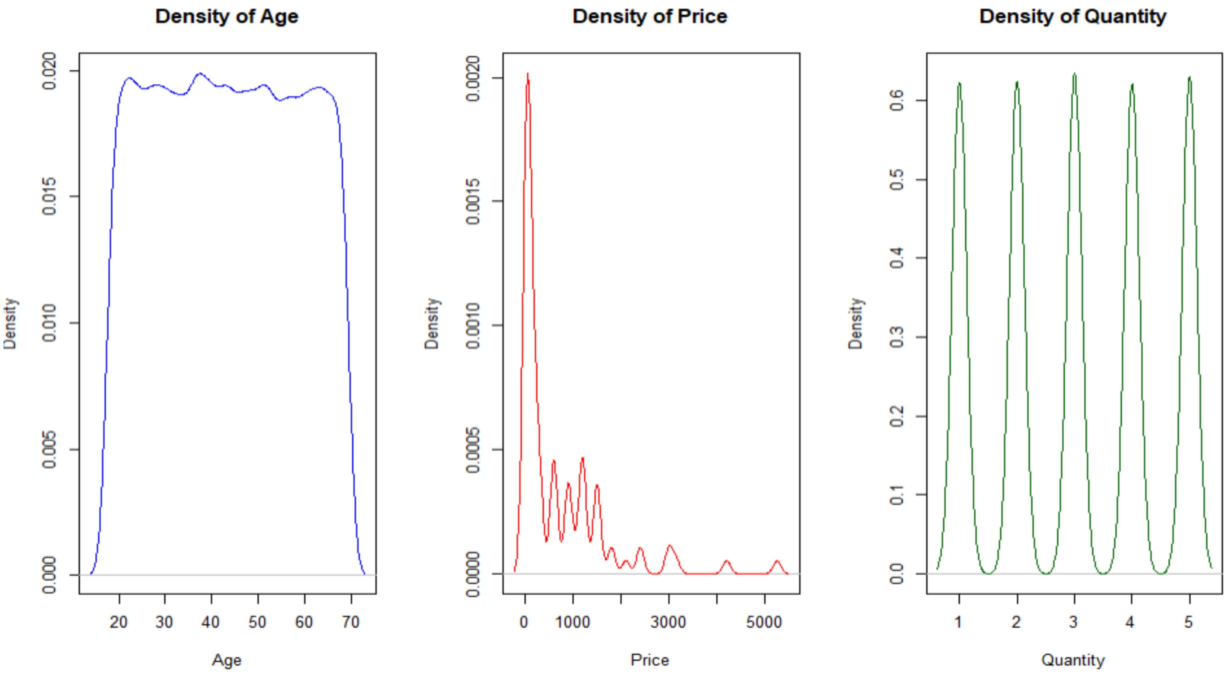


Figure 6. Distribution plot for each sales data

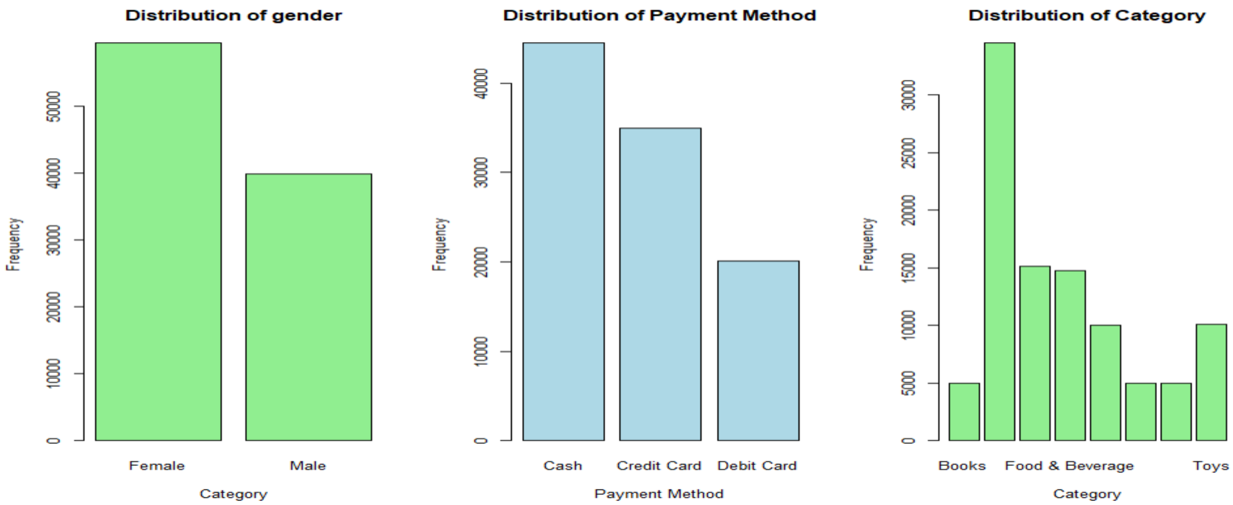


Figure 7. Bar plot for each categorical sales data

Task 1.3: Correlation and scatter plots (between different input customer data and the output predicted sales quantity) to examine their dependencies

The correlation matrix between the different input customer data and the output predicted sales quantity given in Table 1 and graphical view presenting in Figure 8.

Table 1: Correlation matrix

```
                  age     quantity       price
age      1.0000000000 0.0006666459 0.001694134
quantity 0.0006666459 1.0000000000 0.344879843
price    0.0016941339 0.3448798428 1.000000000
```

The age and price variables were used to check the correlation with quantity because these are continuous variables. The price has low positive correlation with R=0.3448 with quantity of sales.

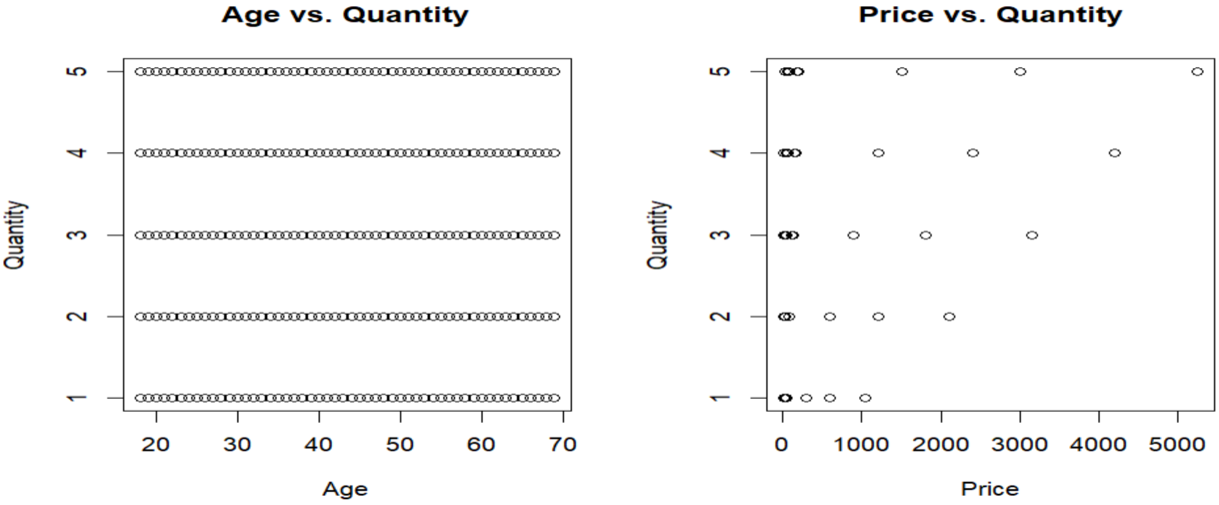


Figure 8. Bar plot for each categorical sales data

Task 2

Task 2.1: Fit ridge regression models

By incorporating polynomial terms such as x^2, X^3, and so on, polynomial regression expands on linear regression by accounting for nonlinear interactions. Beyond linear relationships, it is useful for modeling complicated data patterns (Kleinbaum & Kupper, 1978). Higher-degree polynomials, however, require care to avoid overfitting (Hastie et al., 2009). This problem is addressed by regularization techniques such as ridge regression (Hoerl & Kennard, 1970).

The regression is fit using the "lm" function in R. The coefficients and theats results of 5 seleced models are given in Table 2.

$$\text{Model 1:}\quad y = \theta_1 x_4 + \theta_2 {x_1}^2 + \theta_3 {x_1}^3 + \theta_4 {x_2}^4 + \theta_5 {x_1}^4 + \theta_{bias} + \varepsilon$$

$$\text{Model 2:}\quad y = \theta_1 x_4 + \theta_2 {x_1}^3 + \theta_3 {x_3}^4 + \theta_{bias} + \varepsilon$$

$$\text{Model 3:}\quad y = \theta_1 {x_3}^3 + \theta_2 {x_3}^4 + \theta_{bias} + \varepsilon$$

$$\text{Model 4:}\quad y = \theta_1 x_2 + \theta_2 {x_1}^3 + \theta_4 {x_3}^4 + \theta_{bias} + \varepsilon$$

$$\text{Model 5:}\quad y = \theta_1 x_4 + \theta_2 {x_1}^2 + \theta_3 {x_1}^3 + \theta_4 {x_3}^4 + \theta_{bias} + \varepsilon$$

Table 2: Coefficients and thetas of the 5 models.

> print(coef(ridge_model1)) 7 x 1 sparse Matrix of class "dgCMatrix" s0 (Intercept) 3.000651e+00 . X4 1.146945e-03 2.287433e-07 1.790613e-09 -3.270184e-08 1.713208e-11	> print(coef(ridge_model2)) 5 x 1 sparse Matrix of class "dgCMatrix" s0 (Intercept) 2.971200e+00 . X4 1.320384e-03 1.648575e-09 1.967372e-15
> print(coef(ridge_model3)) 4 x 1 sparse Matrix of class "dgCMatrix" s0 (Intercept) 2.954292e+00 V1 . V2 8.594136e-12 V3 9.581656e-16	> print(coef(ridge_model4)) 5 x 1 sparse Matrix of class "dgCMatrix" s0 (Intercept) 3.016082e+00 . X2 -1.146489e-02 1.276509e-09 2.007756e-15
> print(coef(ridge_model5)) 6 x 1 sparse Matrix of class "dgCMatrix" s0 (Intercept) 2.971119e+00 . X4 1.320438e-03	

6.689937e-08 1.108452e-09 1.967371e-15	

Task 2.2: Calculate Sum of squared error (RSS)

Residual Sum of Squares, or RSS, is a crucial regression analysis statistic that expresses the squared discrepancies between observed and expected values. Regression methods seek to improve predicted accuracy by reducing RSS (Montgomery et al., 2012).

The RSS were calculated and presenting in Table 3 for 5 models. On the based of RSS the model 3 got lower RSS which indicate that model3 is good fitted.

Table 3. Presenting the RSS of 5 models.

	Model1	**Model2**	**Model3**	**Model4**	**Model5**
RSS	198577.6	192239.1	189851.9	192028.1	192239.1

Task 2.3: Compute log-likelihood for each model

In statistical modeling, the function of log likelihood is a crucial tool for assessing how well a model matches observed data. To aid in parameter estimation, it measures the probability of seeing the data provided a set of parameters for the model (Gelman et al., 2013).

The loglikelihood function were calculated and presenting in Table 4 for 5 models. On the based of log-likelihood the model 3 got lower log-likelihood which indicate that model3 is good fitted.

Table 4. Presenting the log-likelihood of 5 models.

	Model1	**Model2**	**Model3**	**Model4**	**Model5**
log-likelihood	-175508.4	-173895.2	-173273.8	-173840.6	-173895.2

Task 2.4: Compute AIC and BIC for each model

Statistical measures called AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion) are used in the model selection process. In order to choose the most frugal model that sufficiently describes the data, they strike a compromise between model fit and complexity (Burnham & Anderson, 2002). AIC and BIC offer distinct trade-offs among goodness-of-fit and model complexity; lower values denote better-fitting models.

The loglikelihood function were calculated and presenting in Table 4 for 5 models. On the based of AIC and BIC the model 3 got lower AIC and BIC which indicate that model3 is good fitted.

Table 5. Presenting the AIC and BIC of 5 models.

	Model1	Model2	Model3	Model4	Model5
AIC	351028.7	347798.3	346553.6	347689.1	347800.3
BIC	351085.8	347836.4	346582.1	347727.2	347847.9

Task 2.5: QQ-plot

A graphical tool called a Q-Q plot (Quantile-Quantile plot) is used to determine whether two datasets have comparable distributions or whether a piece of data follows a given probability distribution. In most cases, the normal distribution is used as a theoretical distribution against which the quantiles of the data that is observed are compared. A roughly straight line indicates that the data is distributed according to the given distribution (Wilks, 2011).

The Q-Q plot also suggest that model 3 is much better than other models which is presenting in Figure 9.

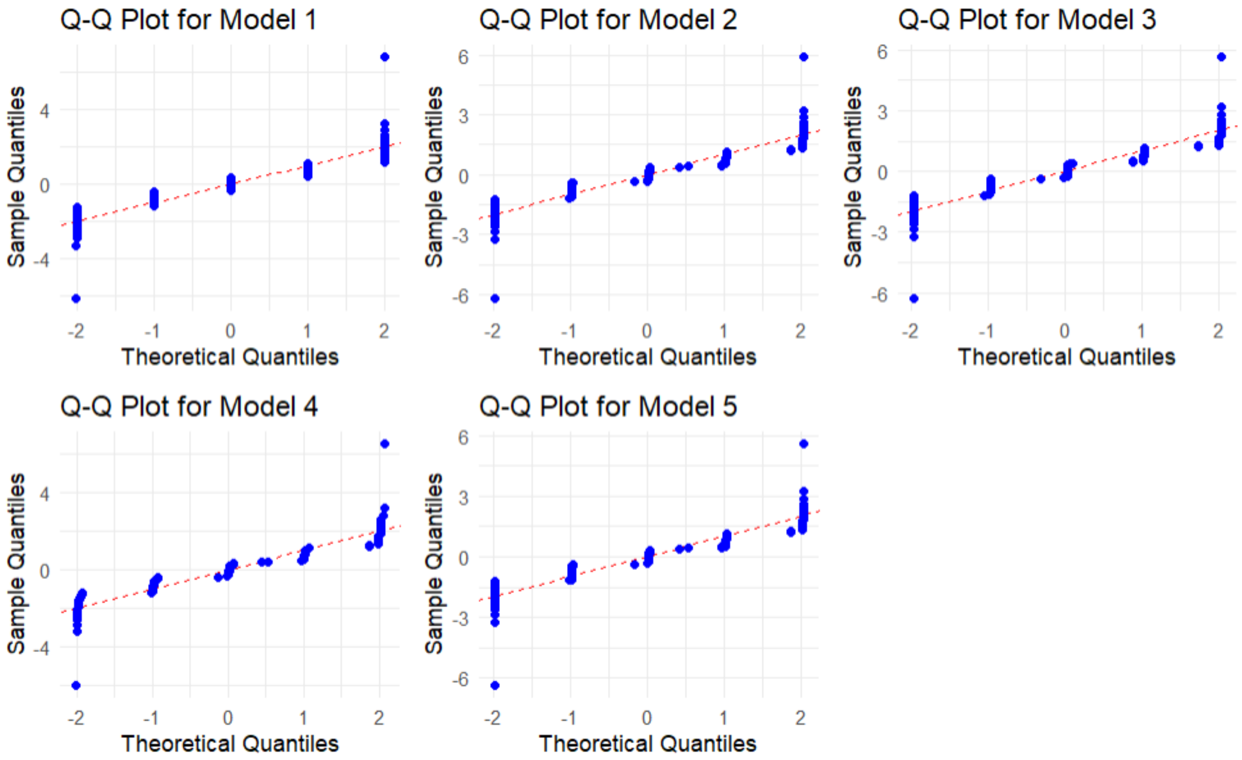


Figure 9: Q-Q plot for 5 models

Task 2.6: Select 'best' regression model according to the AIC, BIC and distribution of model residuals from the 5 candidate models, and explain why you would like to choose this specific model.

All the criteria such as AIC, BIC, RSS, and qq plot suggest that the model 3 is the best model which can be use for further prediction process.

Task 2.7:

The 70% of the dataset was used as training part and 30% was used as testing part. The training samples were used to trained the best model on training samples. The prediction od best model and their confidence interval were given in Figure 10. Where the blue line present actual data and red present predicted data and green present the confidence intervals.

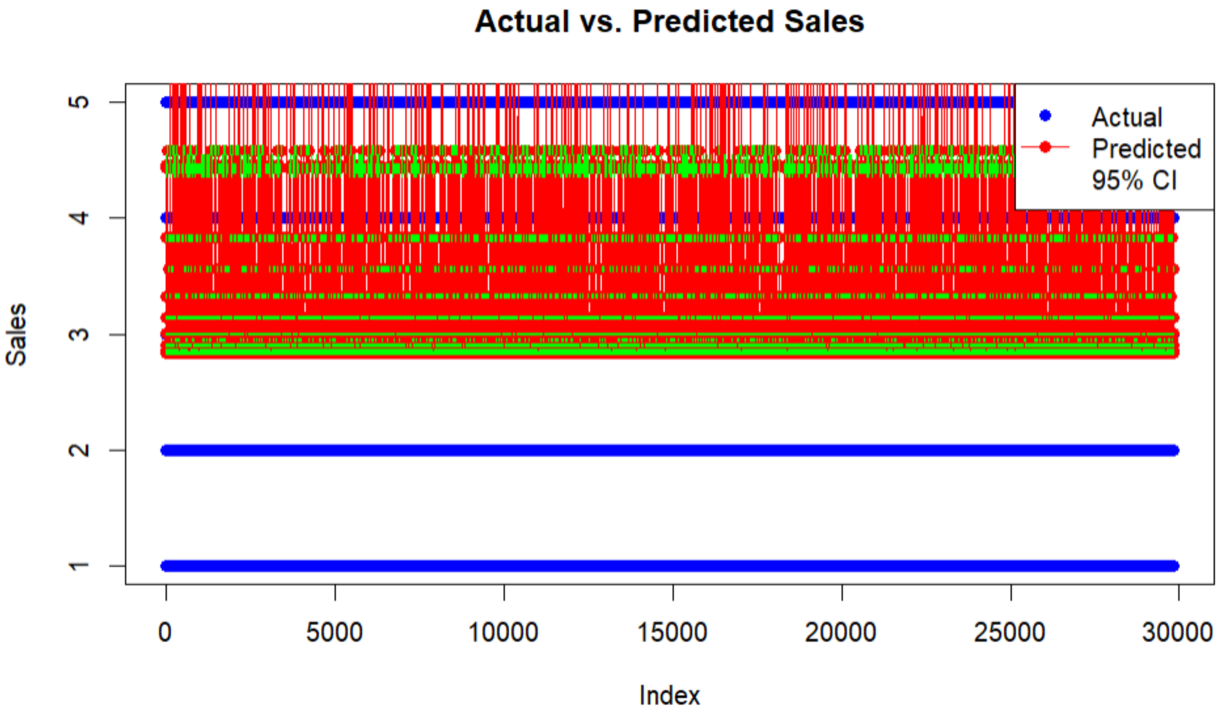


Figure 10: Actual vs predicted sales using best model.

Task 3

The model 3 was selected as best model for Approximate Bayesian Computation (ABC) the two values were specified for the model 3 such as largest and lowest value. The uniform distribution was sued as prior for two selected parameters and 100 range was specified. So if the RSS was below the specified threshold value was used as for plot.

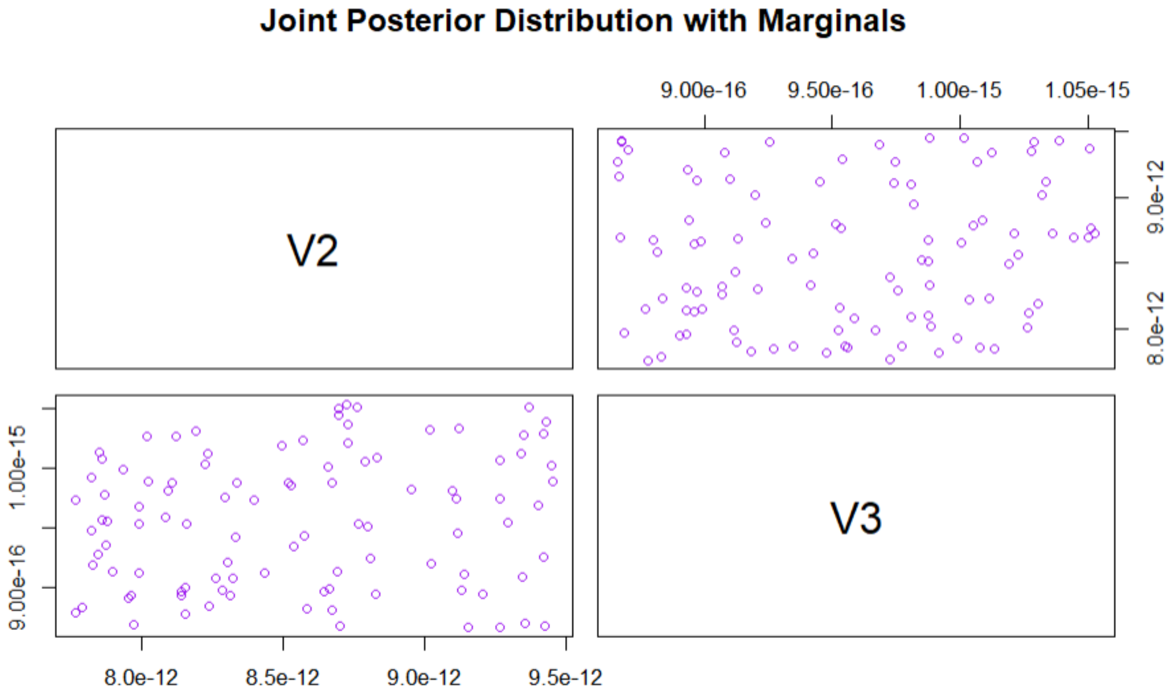


Figure 11: Marginal and joint distributions for the best selected model of output variable

Conclusion

The study objective was to find out the best regression model for the customer shopping dataset for sales. The Model 3 was selected as a best model on the base of RSS, AIC, BIC, qq plot. The model have facility if RSS is below the specified threshold value was used as for plot.

References

Bellenger, D. N., Robertson, D. H., & Hirschman, E. C. (1978). Impulse Buying Varies by Product. Journal of Advertising Research, 18(6), 15–18.

Burnham, K. P., & Anderson, D. R. (2002). Model selection and multimodel inference: a practical information-theoretic approach. Springer Science & Business Media.

Dholakia, U. M. (1999). Going shopping: Key determinants of shopping behaviors and motivations. International Journal of Retail & Distribution Management, 27(4), 154–165.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67.

Holbrook, M. B. (1994). The Nature of Customer Value: An Axiology of Services in the Consumption Experience. Service Quality: New Directions in Theory and Practice, 21(2), 21–71.

Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice. OTexts.

Jones, M. A., Reynolds, K. E., & Arnold, M. J. (2006). Hedonic and utilitarian shopping value: Investigating differential effects on retail outcomes. Journal of Business Research, 59(9), 974–981.

Kleinbaum, D. G., & Kupper, L. L. (1978). Applied regression analysis and other multivariable methods. PWS-Kent Publishing Company.

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). Introduction to linear regression analysis (Vol. 821). John Wiley & Sons.

Sproles, G. B., & Kendall, E. L. (1986). A Methodology for Profiling Consumers' Decision-Making Styles. Journal of Consumer Affairs, 20(2), 267–279.

Wilks, D. S. (2011). Statistical methods in the atmospheric sciences (Vol. 100). Academic Press.

APPENDIX

Codes

```
#import important libraries
library("OpenMx")
library("lavaan")
library("semPlot")
library("tidyverse")
library("mice")
library("MoEClust")
library("performance")
library("mediation")
library("Hmisc")
# Load the data from the CSV file
customer_shopping_data <-
read.csv("D:/work/So/customer_shopping_data_1695379411426.csv")
str(customer_shopping_data)
head(customer_shopping_data,5)

#Data cleaning and check missing values
data_var <-
  customer_shopping_data %>%
  dplyr::select(1:9)
md.pattern(data_var)

#Frequency Distribution of categorical variables
dataframe_categorical_variables=data.frame(customer_shopping_data$gender,customer_shopping_data$category
                                   ,customer_shopping_data$payment_method)
describe(dataframe_categorical_variables)


#Task 1.1

library(ggplot2)
library(corrplot)
#For input
```

```
customer_shopping_data$invoice_date <- as.Date(customer_shopping_data$invoice_date,
format = "%d/%m/%Y")

# Extract month from invoice date
customer_shopping_data$invoice_month <- format(customer_shopping_data$invoice_date,
"%Y-%m")
agg_data <- aggregate(cbind(age, price, quantity) ~ invoice_month, data =
customer_shopping_data, sum)
# Convert aggregated data into a time series object
ts_data <- ts(agg_data [, c("age", "price", "quantity")],
           start = c(2021, 1), frequency = 12)
# Plot the time series
plot.ts(ts_data,
       main = "Time Series Plot of Age, Price, and Quantity",
       ylab = "Values",
       col = 1:3) # Different colors for each variable
dev.off()
#For output
total_sales <- aggregate(quantity ~ invoice_month, data = customer_shopping_data, sum)
sales_ts <- ts(total_sales$quantity, start = c(2021, 1), frequency = 12)
plot(sales_ts, main = "Time Series Plot of Total Sales Quantity by Month", xlab = "Date", ylab =
"Total Sales Quantity")

#Task 1.2
age_hist <- hist(customer_shopping_data$age, plot = FALSE)
age_density <- density(customer_shopping_data$age)

# Histogram density for price
price_hist <- hist(customer_shopping_data$price, plot = FALSE)
price_density <- density(customer_shopping_data$price)

# Histogram density for quantity
quantity_hist <- hist(customer_shopping_data$quantity, plot = FALSE)
quantity_density <- density(customer_shopping_data$quantity)

# Setting up the graphical parameters for multiple plots
par(mfrow = c(1, 3))  # 1 row, 3 columns

# Plotting histogram densities
plot(age_density, col = "skyblue", main = "Density of Age", xlab = "Age", ylab = "Density")
lines(age_density, col = "blue")

plot(price_density, col = "salmon", main = "Density of Price", xlab = "Price", ylab = "Density")
lines(price_density, col = "red")
```

```
plot(quantity_density, col = "lightgreen", main = "Density of Quantity", xlab = "Quantity", ylab
= "Density")
lines(quantity_density, col = "darkgreen")


# Bar plot for category
category_freq <- table(customer_shopping_data$gender)
barplot(category_freq, main = "Distribution of gender", xlab = "Category", ylab = "Frequency",
col = "lightgreen")


# Bar plot for payment method
payment_method_freq <- table(customer_shopping_data$payment_method)
barplot(payment_method_freq, main = "Distribution of Payment Method", xlab = "Payment
Method", ylab = "Frequency", col = "lightblue")

# Bar plot for category
category_freq <- table(customer_shopping_data$category)
barplot(category_freq, main = "Distribution of Category", xlab = "Category", ylab =
"Frequency", col = "lightgreen")


#Task 1.3
cor(customer_shopping_data[, c("age", "quantity", "price")])

par(mfrow = c(1, 2))  # 1 row, 3 columns

# Scatter plots
with(customer_shopping_data, {
  plot(age, quantity, main = "Age vs. Quantity", xlab = "Age", ylab = "Quantity")
  plot(price, quantity, main = "Price vs. Quantity", xlab = "Price", ylab = "Quantity")
 })


#Task 2.1
# Extracting predictor variables
X2 <- as.numeric(factor(customer_shopping_data$category))
X4 <- as.numeric(factor(customer_shopping_data$payment_method))
X1 <- customer_shopping_data$age
X3 <- customer_shopping_data$price

# Creating polynomial features
X_poly <- cbind(X1^2, X1^3, X1^4, X2^4, X3^4, X3^3, X4)

# Adding intercept column
X_poly <- cbind(1, X_poly)
```

```
# Convert target variable to matrix
y <- as.matrix(customer_shopping_data$quantity)
X <- cbind(X1,X2,X3,X4)
# Fit ridge regression models
library(glmnet)

alpha <- 0  # Ridge regression
lambda <- 1  # Regularization parameter

# Define design matrices for each model
Y1 <- cbind(1, X4, X1^2, X1^3, X2^4, X1^4)
Y2 <- cbind(1, X4, X1^3, X3^4)
Y3 <- cbind(1, X3^3, X3^4)
Y4 <- cbind(1, X2, X1^3, X3^4)
Y5 <- cbind(1, X4, X1^2, X1^3, X3^4)

# Fit ridge regression models
ridge_model1 <- glmnet(Y1, y, alpha = alpha, lambda = lambda)
ridge_model2 <- glmnet(Y2, y, alpha = alpha, lambda = lambda)
ridge_model3 <- glmnet(Y3, y, alpha = alpha, lambda = lambda)
ridge_model4 <- glmnet(Y4, y, alpha = alpha, lambda = lambda)
ridge_model5 <- glmnet(Y5, y, alpha = alpha, lambda = lambda)

# Print coefficients
print(coef(ridge_model1))
print(coef(ridge_model2))
print(coef(ridge_model3))
print(coef(ridge_model4))
print(coef(ridge_model5))


#2.2
# Predictions for each model
y_pred_1 <- predict(ridge_model1, newx = Y1)
y_pred_2 <- predict(ridge_model2, newx = Y2)
y_pred_3 <- predict(ridge_model3, newx = Y3)
y_pred_4 <- predict(ridge_model4, newx = Y4)
y_pred_5 <- predict(ridge_model5, newx = Y5)

# Compute residuals for each model
residuals_1 <- y - y_pred_1
residuals_2 <- y - y_pred_2
residuals_3 <- y - y_pred_3
residuals_4 <- y - y_pred_4
residuals_5 <- y - y_pred_5
```

```
# Compute RSS for each model
RSS_1 <- sum(residuals_1^2)
RSS_2 <- sum(residuals_2^2)
RSS_3 <- sum(residuals_3^2)
RSS_4 <- sum(residuals_4^2)
RSS_5 <- sum(residuals_5^2)

# Print RSS for each model
cat("RSS for Model 1:", RSS_1, "\n")
cat("RSS for Model 2:", RSS_2, "\n")
cat("RSS for Model 3:", RSS_3, "\n")
cat("RSS for Model 4:", RSS_4, "\n")
cat("RSS for Model 5:", RSS_5, "\n")

#2.3 Compute log-likelihood for each model
log_likelihood_1 <- sum(dnorm(y, mean = y_pred_1, sd = sqrt(var(residuals_1)), log = TRUE))
log_likelihood_2 <- sum(dnorm(y, mean = y_pred_2, sd = sqrt(var(residuals_2)), log = TRUE))
log_likelihood_3 <- sum(dnorm(y, mean = y_pred_3, sd = sqrt(var(residuals_3)), log = TRUE))
log_likelihood_4 <- sum(dnorm(y, mean = y_pred_4, sd = sqrt(var(residuals_4)), log = TRUE))
log_likelihood_5 <- sum(dnorm(y, mean = y_pred_5, sd = sqrt(var(residuals_5)), log = TRUE))

# Print log-likelihood for each model
cat("Log-Likelihood for Model 1:", log_likelihood_1, "\n")
cat("Log-Likelihood for Model 2:", log_likelihood_2, "\n")
cat("Log-Likelihood for Model 3:", log_likelihood_3, "\n")
cat("Log-Likelihood for Model 4:", log_likelihood_4, "\n")
cat("Log-Likelihood for Model 5:", log_likelihood_5, "\n")

#2.4
# Number of observations
n <- length(y)

# Count number of parameters for each model
num_params_1 <- sum(coef(ridge_model1) != 0)
num_params_2 <- sum(coef(ridge_model2) != 0)
num_params_3 <- sum(coef(ridge_model3) != 0)
num_params_4 <- sum(coef(ridge_model4) != 0)
num_params_5 <- sum(coef(ridge_model5) != 0)

# Compute AIC for each model
AIC_1 <- -2 * log_likelihood_1 + 2 * num_params_1
AIC_2 <- -2 * log_likelihood_2 + 2 * num_params_2
AIC_3 <- -2 * log_likelihood_3 + 2 * num_params_3
AIC_4 <- -2 * log_likelihood_4 + 2 * num_params_4
AIC_5 <- -2 * log_likelihood_5 + 2 * num_params_5
```

```
# Compute BIC for each model
BIC_1 <- -2 * log_likelihood_1 + log(n) * num_params_1
BIC_2 <- -2 * log_likelihood_2 + log(n) * num_params_2
BIC_3 <- -2 * log_likelihood_3 + log(n) * num_params_3
BIC_4 <- -2 * log_likelihood_4 + log(n) * num_params_4
BIC_5 <- -2 * log_likelihood_5 + log(n) * num_params_5

# Print AIC and BIC for each model
cat("AIC for Model 1:", AIC_1, "\n")
cat("AIC for Model 2:", AIC_2, "\n")
cat("AIC for Model 3:", AIC_3, "\n")
cat("AIC for Model 4:", AIC_4, "\n")
cat("AIC for Model 5:", AIC_5, "\n")

cat("\n")

cat("BIC for Model 1:", BIC_1, "\n")
cat("BIC for Model 2:", BIC_2, "\n")
cat("BIC for Model 3:", BIC_3, "\n")
cat("BIC for Model 4:", BIC_4, "\n")
cat("BIC for Model 5:", BIC_5, "\n")


#2.5
# Load necessary libraries
library(ggplot2)

# Function to create Q-Q plot
create_qq_plot <- function(residuals, model_name) {
  qq_data <- data.frame(Theoretical = quantile(residuals, probs = seq(0, 1, 0.01)),
                        Sample = quantile(rnorm(length(residuals), mean = mean(residuals), sd =
sd(residuals)), probs = seq(0, 1, 0.01)))

  ggplot(qq_data, aes(x = Theoretical, y = Sample)) +
    geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "red") +
    geom_point(color = "blue") +
    labs(title = paste("Q-Q Plot for Model", model_name),
         x = "Theoretical Quantiles",
         y = "Sample Quantiles") +
    theme_minimal()
}

# Create Q-Q plots for each model
qq_plot_1 <- create_qq_plot(residuals_1, "1")
qq_plot_2 <- create_qq_plot(residuals_2, "2")
```

```
qq_plot_3 <- create_qq_plot(residuals_3, "3")
qq_plot_4 <- create_qq_plot(residuals_4, "4")
qq_plot_5 <- create_qq_plot(residuals_5, "5")

# Display Q-Q plots
grid.arrange(qq_plot_1, qq_plot_2, qq_plot_3, qq_plot_4, qq_plot_5, ncol = 3)
dev.off()
#2.7
# Load necessary libraries
library(glmnet)
library(boot)
library(ggplot2)
# Set seed for reproducibility
set.seed(123)

# Split the dataset into training and testing sets (70% training, 30% testing)
train_indices <- sample(nrow(X), 0.7 * nrow(X))
X_train <- Y3[train_indices, ]
y_train <- y[train_indices]
X_test <- Y3[-train_indices, ]
y_test <- y[-train_indices]

# Convert X_train matrix to data frame
X_train_df <- as.data.frame(X_train)

# Train your selected "best" model using the training dataset
best_model <- ridge_model3  # Replace with your best model
best_model_fit <- lm(y_train ~ ., data = X_train_df)  # Fit linear regression model

# Make predictions on the testing data
predictions <- predict(best_model_fit, newdata = as.data.frame(X_test), interval = "confidence")

# Plot the actual testing data and the model predictions
plot(y_test, col = "blue", pch = 16, xlab = "Index", ylab = "Sales", main = "Actual vs. Predicted
Sales")
points(predictions[, 1], col = "red", pch = 16)  # Model predictions
lines(predictions[, 1], col = "red")  # Connect predictions with a line
segments(1:length(y_test), predictions[, 2], 1:length(y_test), predictions[, 3], col = "green")  #
Confidence intervals
legend("topright", legend = c("Actual", "Predicted", "95% CI"), col = c("blue", "red", "green"),
pch = c(16, 16, NA), lty = c(NA, 1, NA))


#3.1
# Assuming the estimated values are:
V1_estimated <- 2.954292e+00  # Intercept
```

```
V4_estimated <- 0  # No coefficient for V4 in Model 3

# Step 3: Define Prior Distributions
V2_estimated <- 8.594136e-12
V3_estimated <- 9.581656e-16

prior_range_V2 <- c(V2_estimated - 0.1 * abs(V2_estimated), V2_estimated + 0.1 *
abs(V2_estimated))
prior_range_V3 <- c(V3_estimated - 0.1 * abs(V3_estimated), V3_estimated + 0.1 *
abs(V3_estimated))

# Step 4: Draw Samples
n_samples <- 100  # Number of samples
V2_samples <- runif(n_samples, min = prior_range_V2[1], max = prior_range_V2[2])
V3_samples <- runif(n_samples, min = prior_range_V3[1], max = prior_range_V3[2])

# Step 3: Draw Samples
n_samples <- 100  # Increasing the number of samples for better accuracy

V2_samples <- runif(n_samples, min = prior_range_V2[1], max = prior_range_V2[2])
V3_samples <- runif(n_samples, min = prior_range_V3[1], max = prior_range_V3[2])

# Joint posterior distribution for V2 and V3
plot(V2_samples, V3_samples, main = "Joint Posterior Distribution for V2 and V3", xlab =
"V2", ylab = "V3", col = "orange")

# Scatterplot with marginal histograms for V2 and V3
pairs(data.frame(V2 = V2_samples, V3 = V3_samples), main = "Joint Posterior Distribution with
Marginals", col = "purple")
```