DECLARATION:

I want to acknowledge using Co-Pilot as a tool to complete this project. Also, this is aligned with the university-referenced guidelines that 'use of AI-supported/AI-integrated tools is permitted'. Further, I acknowledge the use of GenAI tools in this assessment for the following:

- For developing ideas.
- To assist with research or gathering information.
- To help me understand key theories and concepts.
- To identify trends and themes as part of my data analysis.
- To suggest a plan or structure for my assessment.
- To give me feedback on a draft.
- To generate images, figures or diagrams.
- To proofread and correct grammar or spelling errors.
- To generate citations or references.

I have not used any GenAI tools to prepare this report. I have referenced using GenAI outputs within my assessment per the University referencing guidelines.

INTRODUCTION:

A supermarket is a place where people of all genders come to shop for various items and use multiple payment methods to purchase them. But the real question is: what factors affect the total sales of the supermarket?

With many consumers and various products coupled with stiff competition, every supermarket wants to stay ahead. Options available to customers across all facets are substantial, and one needs to analyse whether these options are indeed having an impact on sales and performances or not. A thorough analysis of all the data is essential to find out the same. The analysis conducted will help us answer the below-enumerated aspects :

1. To understand the fundamental trends between columns using charts.
2. Missing data handling
3. Hypothesis testing: What categories impact total sales?

OBJECTIVES:

The main objective here is to help a supermarket understand its strengths and trends, make informed decisions to improve its business, enhance the customer experience, and highlight the factors to consider to boost its sales.

Further, this exercise emphasises that data cleaning and error handling are critical aspects of the date and time column. As the supermarket has a wide variety of items, it is only fitting that hypothesis testing be done to spot the crucial factors and relations between them.

DATA:

The dataset that I have chosen is the “Supermarket sales” dataset from Kaggle.com, which is a well-known site for finding datasets to work with. It shows transaction records from a supermarket chain in Myanmar across three cities: Yangon, Naypyitaw, and Mandalay. It has a total of 1000 rows. It provides a comprehensive view of sales activities, customer demographics, and payment methods from January to March 2019. First, I cleaned and filtered this data to prepare it for my analysis. Outside of my taught syllabus, I used the **lubridate package** to correct the date. The format was m-dd and mm-dd in the date column, all in character. Using the **strptime()** function, the date and time are combined to get the exact time. Following this, I extracted the date from this column into a new date and then got the month.

I used pipelining and filter to remove the unneeded columns. The only columns that I kept are:

- branch,
- city,
- customer type
- Gender,
- product line,
- unit price,
- quantity,
- sales,
- data,
- payment,
- new date
- month.

RESULTS:

First, we have to look at the summary of the dataset. The focus is mainly on mean, standard deviation, minimum, and maximum. The columns that are focused on are the quantity, sales and rating columns. I used the **summarise** function instead of the summary, and I used pipelining.

Table: Summary Statistics for Total Sales, Quantity, and Rating

Statistic	Total Sales	Quantity	Rating
Mean	322.9667	5.576704	6.97270
Standard Deviation	245.8853	2.934045	1.71858
Min	10.6785	1.000000	4.00000
Max	1042.6500	10.000000	10.00000

Per transaction, the store earns 323 Kyat. Kyat is the Myanmar currency. The standard deviation is 246 Kyat. Total sales are calculated in this way:

Cost of goods sold = quantity * unit price

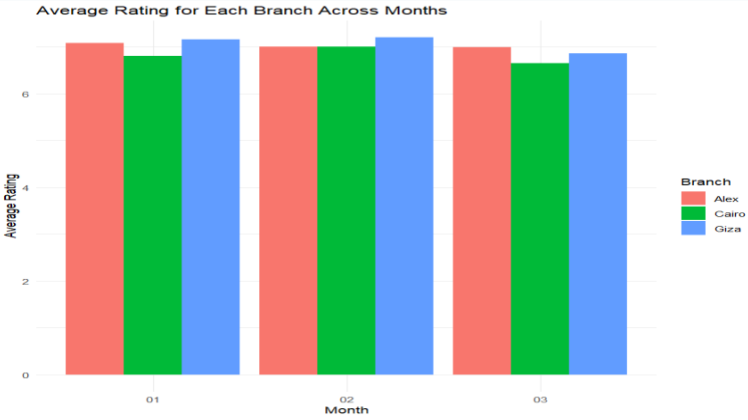
Tax = 0.05 * cost of goods sold

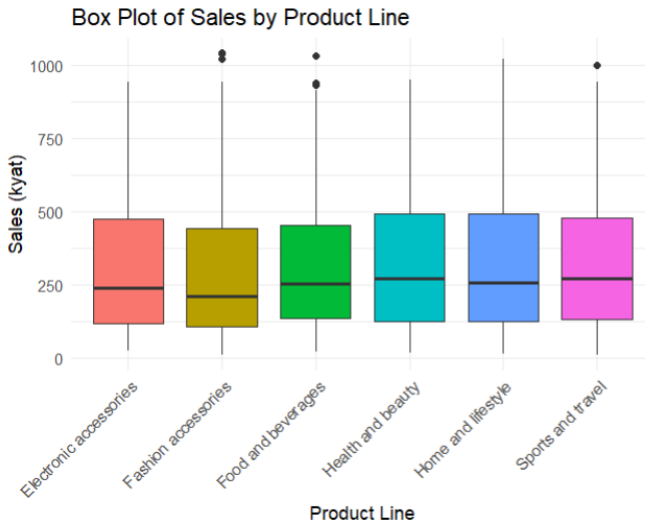
Total sales = cost of goods sold + tax

From this, the store keeps 4.76% of its gross income.

Gross income = 4.76/100 * total sales

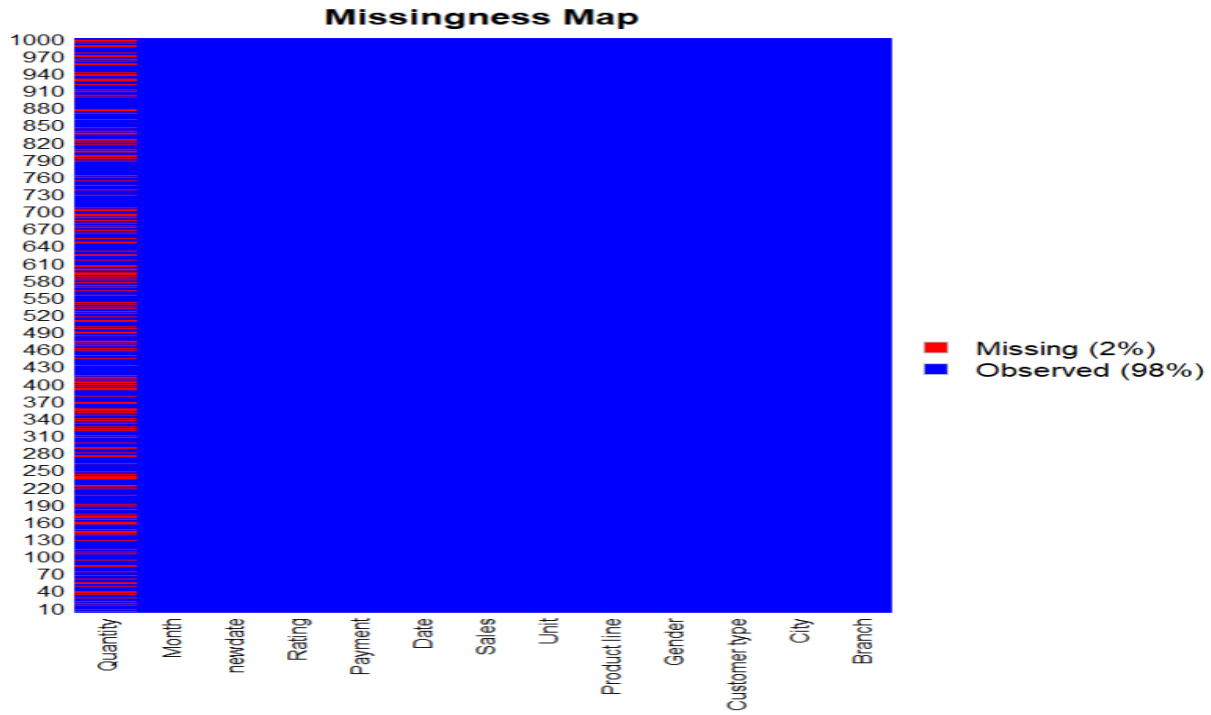
But the average rating for the store is just under 7 (6.97). Ratings range from 10 to 4. Based on this, we can deduce that people give an average rating of 5 to 8 for the three stores. The quantities in which people buy their items vary from 1 to 10, showing that small and big shopping happens. This is a basic understanding of the data that has been cleaned. The graphs for this data given below will help us understand better. The first graph shows us the three branches of three cities in Myanmar and their average ratings across months in a histogram. The second graph shows us the six types of items sold as a boxplot.

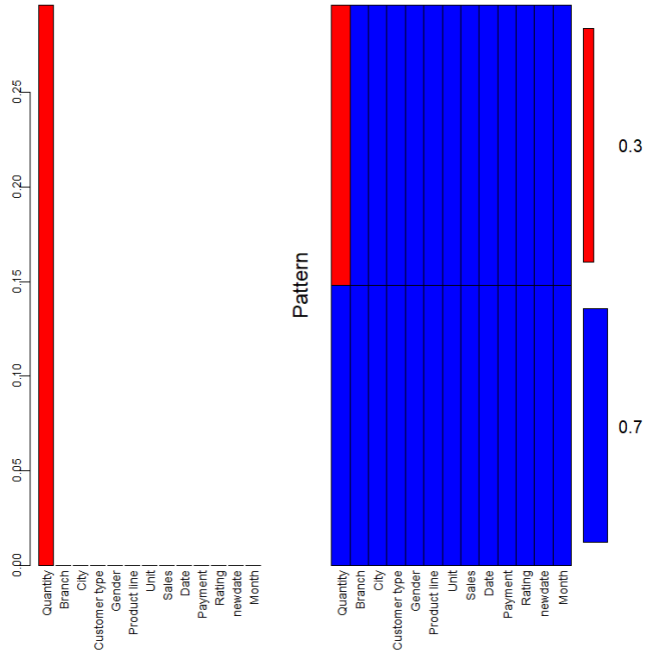




MISSING DATA HANDLING:

To show how missing data is handled, I had no missing data in this dataset, so I had to create missing values using ChatGPT. I used that new dataset with missing values for my analysis. I plotted the missingness map using **the Amelia** package.





I first printed the number of missing values in each column to handle missing data. I used **sapply()** for this. The quantity has 296 missing values. The missing data handling package that we use here is MICE. We impute the missing data using the MICE package with m=10. After doing so, we get the completed dataset.

CORRELATION AND HYPOTHESIS:

Following this, we fit a regression model on each imputed dataset, pool the result using **Rubin's rule** for multiple imputations and combine the results from numerous imputed datasets. It replaces each missing value with a set of plausible values representing the

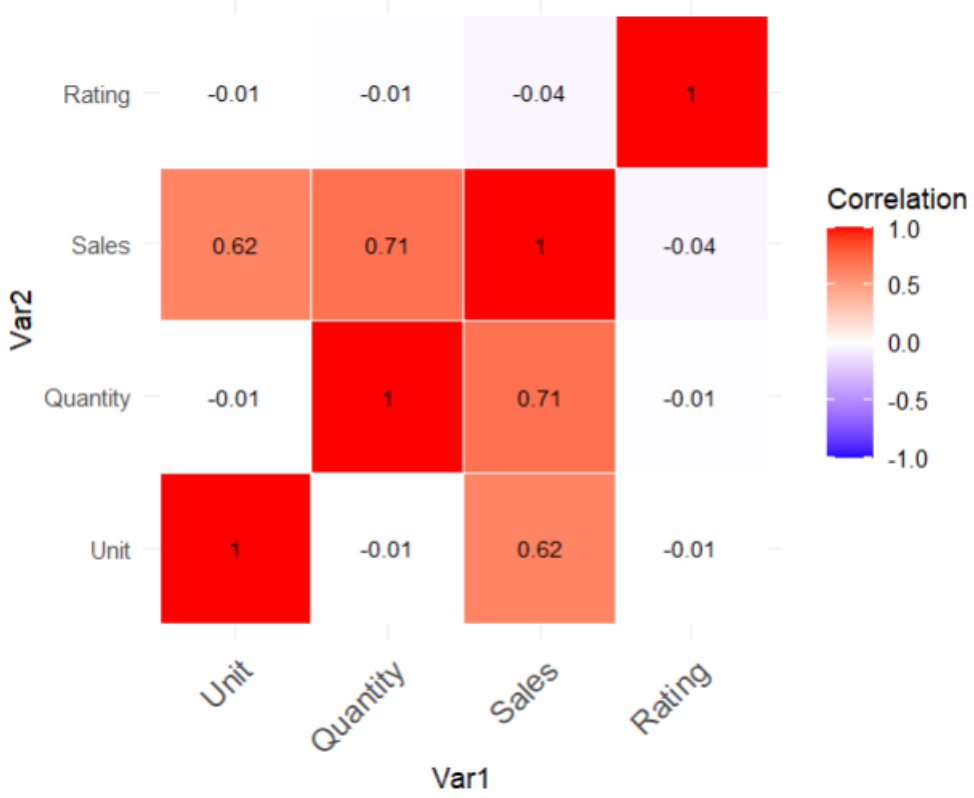
uncertainty about the correct value to impute. Post this, we create a well-defined table showing the results of this.

Table: Summary of Pooled Regression Results

term	estimate	std.error	statistic	df	p.value
(Intercept)	-305.204734	16.5892029	-18.3977938	153.34245	0.0000000
Unit	5.948239	0.1126838	52.7869750	99.91335	0.0000000
Quantity	59.189799	0.9223733	64.1711999	469.11863	0.0000000
Rating	-4.447936	1.5914308	-2.7949292	305.78614	0.0055192
GenderMale	-6.753923	5.5723109	-1.2120506	332.91354	0.2263525
CityNaypyitaw	1.682342	6.7328049	0.2498724	314.77712	0.8028490
CityYangon	1.203440	6.4739501	0.1858895	521.31268	0.8526037
`Customer type`Normal	1.090709	5.4096661	0.2016223	510.34190	0.8402923
`Product line`Fashion accessories	-1.501262	9.3153907	-0.1611593	282.72733	0.8720830
`Product line`Food and beverages	2.012120	9.5414746	0.2108815	208.16209	0.8331860
`Product line`Health and beauty	1.360539	9.9224655	0.1371170	191.81807	0.8910821
`Product line`Home and lifestyle	4.774813	9.8817748	0.4831939	169.96800	0.6295799
`Product line`Sports and travel	-2.748748	9.3025863	-0.2954821	379.80077	0.7677871

```
>
```

The coefficient of the month is 0.03, indicating that the month has a positive relationship with sales. However, it is not significant as it has a p-value of 0.135. The quantity has a very low p-value and a coefficient of 1.23, which indicates that it is very effective on total sales. For each unit increase in quantity, sales more than doubles. Gender, however, has a negative relationship. Its p-value is also 0.073. The p-value proves the significance. If less than 0.05, it is significant. Otherwise, the importance is not valid.

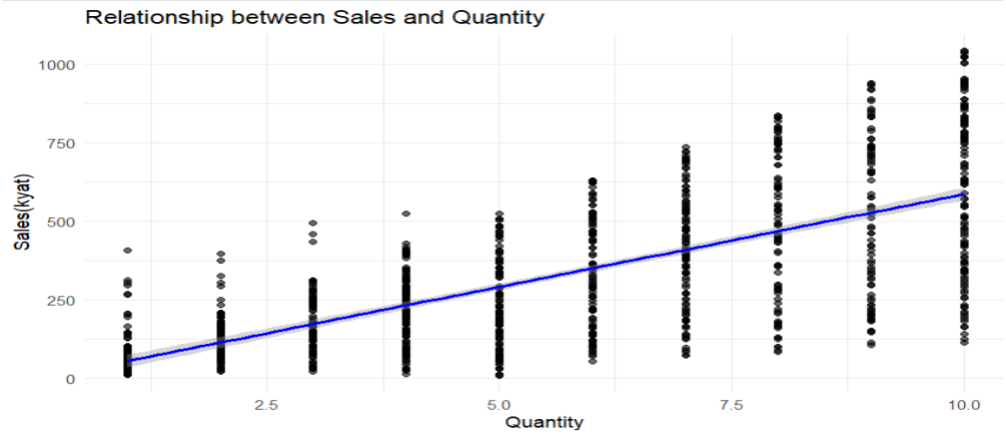


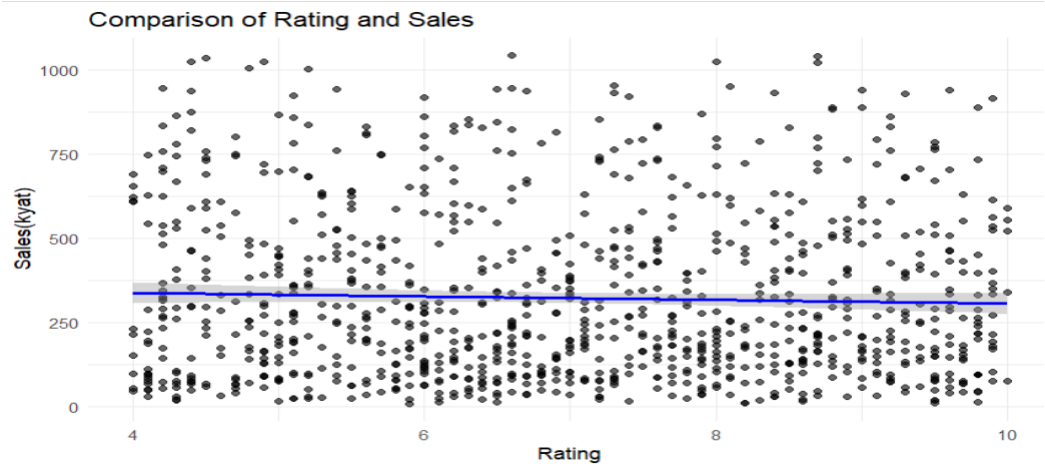
We can deduce three relationships from this correlation heatmap between the numeric values. That is:

1. Strong positive
2. Strong Negative

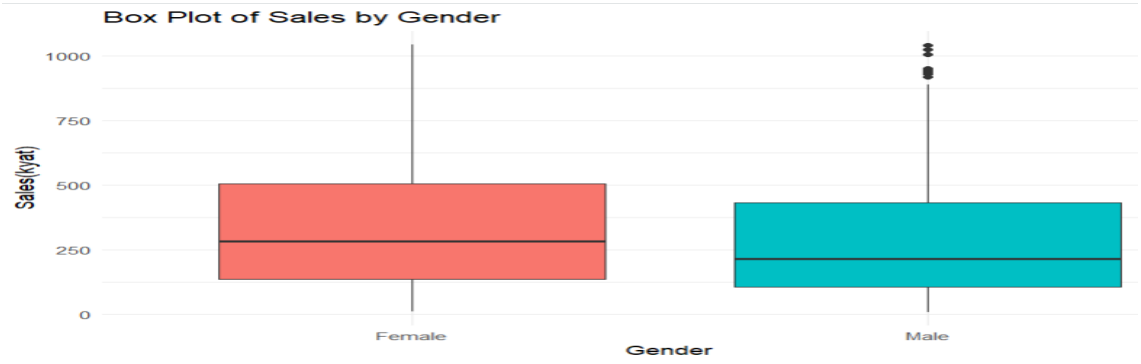
3. No correlation

For the first one, we can see that quantity and sales have a strong positive relationship. The more items purchased per transaction, the more the total sales will be for the supermarket. Therefore, they should encourage customers to shop more with innovative strategies and offers. We can understand this relation through the scatter plot that I created. However, rating and unit price shows a negative relationship, suggesting that higher ratings are associated with lower unit prices. Cheaper items are better rated.





However, gender, on the other hand, has no real effect on total sales, even though 57% of customers are females, as they are a weak correlation. The boxplot below explains that relationship.



For customer kind, this is how we can compare:

So, gender is not a significant parameter when evaluating total sales or strategising. For members, shopping is much more common than for normal customers. Across three months, members have contributed much more than normal on average. 56% of customers are members of supermarkets. But total sales are a slightly different story. This shows that when normal customers come, they are fewer in number but do more shopping in March. The **linear regression equation** for this is written as:

Sales = -305.2047 + 5.948239 * Unit Price + 59.1898 * Quantity -4.447936 * Rating - 6.753923 * Gender + 1.682342 * City + 1.20344 * Customer type + 1.090709 * Product line

LIMITATION:

The dataset provides much information to us, and we have used the data to correlate it between many columns. Yet, every data has its demerits, and this supermarket dataset has the following limitations:

- **Poor choice for categorisation of items**: The categories for the items are very mixed up. Health and beauty, sports and travel are two such categories. Health products like medicine, bandages, and maybe even health drinks and proper meals with nutrition can be available. While beauty is makeup, lotion, soaps, and creams. Sports can involve bats, balls, rackets, and shuttles, and travel is more about clothing, travel bags, suitcases, and other things. This is why the product line is not so effective in total sales. A better categorisation would have led to better results and projected more accuracy on the effect on store business.
- **Imputed Missing Data and Uncertainty**: Missing values were addressed using statistical imputation, in which information is estimated based on current patterns. Although the process shows completeness, there is a degree of uncertainty in it. Therefore, its reliability is questionable.
- **Lack of more rows:** The dataset only contains 1000 rows of data. This is only for sales for three months, that is, one quarter of 2019. This may seem optimum, but more

data is necessary to create better plans. We could have made a better predictive analysis if the data had been for a year.

- **No information on Member discount:** Comparing members and regular customers regarding mean sales, it is clear that members have made much bigger sales. 56% of the customers are members but spend a lot more. An insight into the discount or membership perks would have helped us understand. We don't know what perks or advantages they receive, which is a big miss when analysing this dataset.
- **Only one branch in one city:** Many branches in a town would be preferred for a supermarket.

CONCLUSION:

To conclude, this report showed us the trends and flows that the supermarket observes across branches, products, and months. We observed and analysed many factors to determine whether they affect total sales and, if they do, how much effect they have.

This was the crux of our hypothesis testing. The test showed us that significant predictors such as unit price, quantity, and rating strongly impact Sales. People tend to give higher ratings for products with lower unit prices, suggesting that "all that glistens is not gold." Members contribute to the sales more often than normal customers. But some normal customers do a lot of shopping sometimes, which can suggest that they might not be informed. Especially in March.

However, this analysis has some limitations caused by the limited data—only three months of sales—and the poor choice of categorisation, which doesn't provide us with the insight needed to identify some more key issues that can be found in the data that were made available. However, with the existing dataset, some fundamental factors were identified, which the store can use to improve its performance in sales and give customers a better shopping experience.

APPENDIX:

```
# Load necessary libraries

install.packages("lubridate")

install.packages("mice")

install.packages("ggplot2")

install.packages("Amelia")

library(lubridate)

library(reshape2)

library(mice)

library(dplyr)
```

```
library(ggplot2)
library(patchwork)
library(VIM)
library(broom)
library(tidyverse)
library(knitr)
library(Amelia)
rdata <- read_csv("C:/Users/homehp/Downloads/SuperMarket_Analysis_Missing_Only.csv")
str(rdata)

#correcting the date and time into one set
rdata$fulltime <- paste(rdata$Date, rdata$Time)
rdata$fulltime
rdata$fulltime <- strptime(rdata$fulltime, format = "%m/%d/%Y %H:%M:%S")
sort(rdata$fulltime)
rdata$newdate <- as.Date(rdata$fulltime)
str(rdata$newdate)
rdata$Month <- format(rdata$newdate, "%m")
str(rdata$Month)
#Remove Columns using pipelining
rdata <- rdata %>%
  select(-Tax, -`Invoice ID`, -cogs, -`gross margin percentage`, -`gross income`, -fulltime, -
Time, -Date)
#Renaming "Unit price" as "Unit_Price" using pipelining
rdata <- rdata %>%
  rename(Unit_Price = `Unit price`)
view(rdata)

#Summary table
summ_tab <- rdata %>%
```

```
  summarise(
    Statistic = c("Mean", "SD", "Min", "Max"),
    `Total Sales` = c(mean(Sales, na.rm = TRUE), sd(Sales, na.rm = TRUE), min(Sales, na.rm = TRUE), max(Sales, na.rm = TRUE)),
    Quantity = c(mean(Quantity, na.rm = TRUE), sd(Quantity, na.rm = TRUE), min(Quantity, na.rm = TRUE), max(Quantity, na.rm = TRUE)),
    Rating = c(mean(Rating, na.rm = TRUE), sd(Rating, na.rm = TRUE), min(Rating, na.rm = TRUE), max(Rating, na.rm = TRUE))
  )
summ_tab
kable(summary_specific, format = "markdown", caption = "Summary Statistics for Total Sales, Quantity, and Rating")


#Line chart of mean sales of type of customers across months
mean_sales <- completed_data %>%
  group_by(Month, `Customer type`) %>%
  summarise(Mean_Sales = mean(Sales, na.rm = TRUE))
mean_sales_plot <- ggplot(mean_sales, aes(x = Month, y = Mean_Sales, color = `Customer type`, group = `Customer type`)) +
  geom_line() +
  labs(title = "Mean Sales Across Months for Normal and Member Customers", x = "Month", y = "Mean Sales(kyat)") +
  theme_minimal()
mean_sales_plot


#Scatter plot of Sales and Quantity
sq_plot <- ggplot(completed_data, aes(x = Quantity, y = Sales)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Relationship between Sales and Quantity", x = "Quantity", y = "Sales(kyat)") +
  theme_minimal()
```

```
sq_plot


#Histogram showing ratings by branch
branch_rating <- completed_data %>%
  group_by(Month, Branch) %>%
  summarise(Average_Rating = mean(Rating, na.rm = TRUE))
rating_hist <- ggplot(branch_rating, aes(x = Month, y = Average_Rating, fill = Branch)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Rating for Each Branch Across Months", x = "Month", y = "Average
Rating") +
  theme_minimal()
rating_hist


# Scatter plot of rating and sales
rating_sales_plot <- ggplot(completed_data, aes(x = Rating, y = Sales)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Comparison of Rating and Sales", x = "Rating", y = "Sales(kyat)") +
  theme_minimal()
print(rating_sales_plot)


# Box plot of Sales and Gender
sg_plot <- ggplot(completed_data, aes(x = Gender, y = Sales, fill = Gender)) +
  geom_boxplot() +
  labs(title = "Box Plot of Sales by Gender", x = "Gender", y = "Sales(kyat)") +
  theme_minimal() +
  theme(legend.position = "none")
sg_plot


#box plot for Sales by Product Line
```

```
sales_plot <- ggplot(rdata, aes(x = `Product line`, y = Sales, fill = `Product line`)) +
  geom_boxplot() +
  labs(title = "Box Plot of Sales by Product Line", x = "Product Line", y = "Sales (kyat)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
sales_plot

#Line chart showing average sales per month
average_sales_plot <- rdata %>%
  group_by(Month, `Product line`) %>%
  summarise(Average_Sales = mean(Sales, na.rm = TRUE)) %>%
  ggplot(aes(x = Month, y = Average_Sales, color = `Product line`, group = `Product line`)) +
  geom_line() +
  labs(title = "Average Sales for Items by Month", x = "Month", y = "Average Sales(kyat)") +
  theme_minimal()
average_sales_plot

#bar graph showing sales for normal and member customers for January, February, and
March
filtered_data <- completed_data %>%
  filter(Month %in% c("01", "02", "03"))
sales_bar_plot <- ggplot(filtered_data, aes(x = Month, y = Sales, fill = `Customer type`)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Sales for Normal and Member Customers (Jan, Feb, Mar)", x = "Month", y =
"Sales(kyat)") +
  theme_minimal()
sales_bar_plot

#Print missing values
miss_val <- sapply(rdata, function(x) sum(is.na(x)))
```

```
print(miss_val)

# Missingness map

missmap(rdata, main = "Missingness Map", col = c("red", "blue"), legend = TRUE)


#Visualize missing values using the aggr plot from the VIM package

aggr_plot <- aggr(rdata, col = c('blue', 'red'), numbers = TRUE, sortVars = TRUE, labels =
names(rdata), cex.axis = .7, gap = 3, ylab = c("Missing data", "Pattern"))

aggr_plot


#Perform missing data analysis using MICE

md.pattern(rdata)


#Impute missing data

imputed_data <- mice(rdata, m = 10, maxit = 50, method = 'pmm', seed = 500)

summary(imputed_data)


#Complete the dataset with imputed values

completed_data <- complete(imputed_data, 1)


#View the completed dataset

view(completed_data)


#Fit a regression model on each imputed dataset

fit <- with(imputed_data, lm(Sales ~ Unit_Price + Quantity + Rating + Gender + City +
`Customer type` + `Product line`))


#Pool results using Rubin's rule

pooled_results <- pool(fit)


#Extract the coefficients from the pooled results
```

```
coefficients <- summary(pooled_results)$estimate

coefficients


#linear regression equation

intercept <- coefficients[1]

unit_price_coef <- coefficients[2]

quantity_coef <- coefficients[3]

rating_coef <- coefficients[4]

gender_coef <- coefficients[5]

city_coef <- coefficients[6]

customer_type_coef <- coefficients[7]

product_line_coef <- coefficients[8]

cat("Linear Regression Equation is:\n")

cat("Sales =", intercept, "+", unit_price_coef, "* Unit_Price +", quantity_coef, "* Quantity +",
rating_coef, "* Rating +", gender_coef, "* Gender +", city_coef, "* City +",
customer_type_coef, "* Customer type +", product_line_coef, "* Product line\n")


#Print the summary table

print(summary_table)

kable(summary_table, format = "markdown", caption = "Summary of Pooled Regression
Results")

#Save the summary table to a CSV file

write.csv(summary_table,
"C:/Users/homehp/Desktop/Project/pooled_results_summary.csv", row.names = FALSE)


#Hypothesis testing

htest <- summary(pooled_results, conf.int = TRUE)

htest

kable(htest, format = "markdown", caption = "Hypothesis Testing Results for Each Predictor")
```

REFERENCES:

1. **lubridate**:
    - Grolemund, G., & Wickham, H. (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. doi:10.18637/jss.v040.i03
2. **mice**:
    - van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software, 45(3), 1-67. doi:10.18637/jss.v045.i03
3. **ggplot2**:
    - Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. doi:10.1007/978-3-319-24277-4
4. **Amelia**:
    - Honaker, J., King, G., & Blackwell, M. (2011). Amelia II: A Program for Missing Data. Journal of Statistical Software, 45(7), 1-47. doi:10.18637/jss.v045.i07
5. **reshape2**:
    - Wickham, H. (2007). Reshaping Data with the reshape Package. Journal of Statistical Software, 21(12), 1-20. doi:10.18637/jss.v021.i12
6. **dplyr**:
    - Wickham, H., François, R., Henry, L., & Müller, K. (2023). dplyr: A Grammar of Data Manipulation. R package version 1.0.10. https://CRAN.R-project.org/package=dplyr
7. **patchwork**:
    - Pedersen, T. L. (2020). patchwork: The Composer of Plots. R package version 1.1.1. https://CRAN.R-project.org/package=patchwork
8. **VIM**:
    - Kowarik, A., & Templ, M. (2016). Imputation with the R Package VIM. Journal of Statistical Software, 74(7), 1-16. doi:10.18637/jss.v074.i07
9. **broom**:
    - Robinson, D., Hayes, A., & Couch, S. (2023). broom: Convert Statistical Objects into Tidy Tibbles. R package version 0.7.12. https://CRAN.R-project.org/package=broom
10. **tidyverse**:

- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., ... & Yutani, H. (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686. doi:10.21105/joss.01686

11. **knitr**:

    - Xie, Y. (2015). Dynamic Documents with R and knitr. Chapman and Hall/CRC. doi:10.1201/b17408

Citation in APA Style:

Grolemund, G., & Wickham, H. (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. doi:10.18637/jss.v040.i03

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate Imputation by Chained Equations in R. Journal of Statistical Software, 45(3), 1-67. doi:10.18637/jss.v045.i03

Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. doi:10.1007/978-3-319-24277-4

Honaker, J., King, G., & Blackwell, M. (2011). Amelia II: A Program for Missing Data. Journal of Statistical Software, 45(7), 1-47. doi:10.18637/jss.v045.i07

Wickham, H. (2007). Reshaping Data with the reshape Package. Journal of Statistical Software, 21(12), 1-20. doi:10.18637/jss.v021.i12

Wickham, H., François, R., Henry, L., & Müller, K. (2023). dplyr: A Grammar of Data Manipulation. R package version 1.0.10. https://CRAN.R-project.org/package=dplyr

Pedersen, T. L. (2020). patchwork: The Composer of Plots. R package version 1.1.1. https://CRAN.R-project.org/package=patchwork

Kowarik, A., & Templ, M. (2016). Imputation with the R Package VIM. Journal of Statistical Software, 74(7), 1-16. doi:10.18637/jss.v074.i07

Robinson, D., Hayes, A., & Couch, S. (2023). broom: Convert Statistical Objects into Tidy Tibbles. R package version 0.7.12. https://CRAN.R-project.org/package=broom

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., ... & Yutani, H. (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686. doi:10.21105/joss.01686

Xie, Y. (2015). Dynamic Documents with R and knitr. Chapman and Hall/CRC. doi:10.1201/b17408