WALMART STATS ANALYSIS

Sales across Walmart stores

Synopsis

Walmart is one of the largest global retail chains that operates multiple stores across various locations in the United States. To optimize store performance and enhance customer satisfaction, Walmart aims to analyze sales data from several stores over a three-year period (2010 to 2012). This analysis will help the company understand sales trends, the impact of holidays, and the relationship between sales and other factors like temperature, fuel prices, CPI, and unemployment rates.

Objective:

Analyze Walmart store sales data to practice R programming skills, focusing on data manipulation, statistical analysis, and data visualization.

Dataset File: Walmart - Original.csv

Steps for Data Analysis :

- **Setting up the environment - Install and load the packages :**

```
install.packages("tidyverse")
install.packages("summarytools")
install.packages ("ggplot2")
install.packages("dplyr")
library(tidyverse)
library(summarytools)
library(ggplot2)
library(dplyr)
```

Descriptive statistics of key metrics

Highest Weekly Sales: $3,818,686.45
Lowest Weekly Sales: $209,986.25

Summary of key metrics for first 6 rows of data

- Store 4 has both the highest mean and median weekly sales, indicating it performs significantly better in terms of sales compared to other stores.
- Store 5, with the lowest mean and median sales, is underperforming relative to other stores.

Values	
highest_sal…	3818686.45
highest_val…	3818686.45
lowest_sales	209986.25

A tibble: 6 × 13

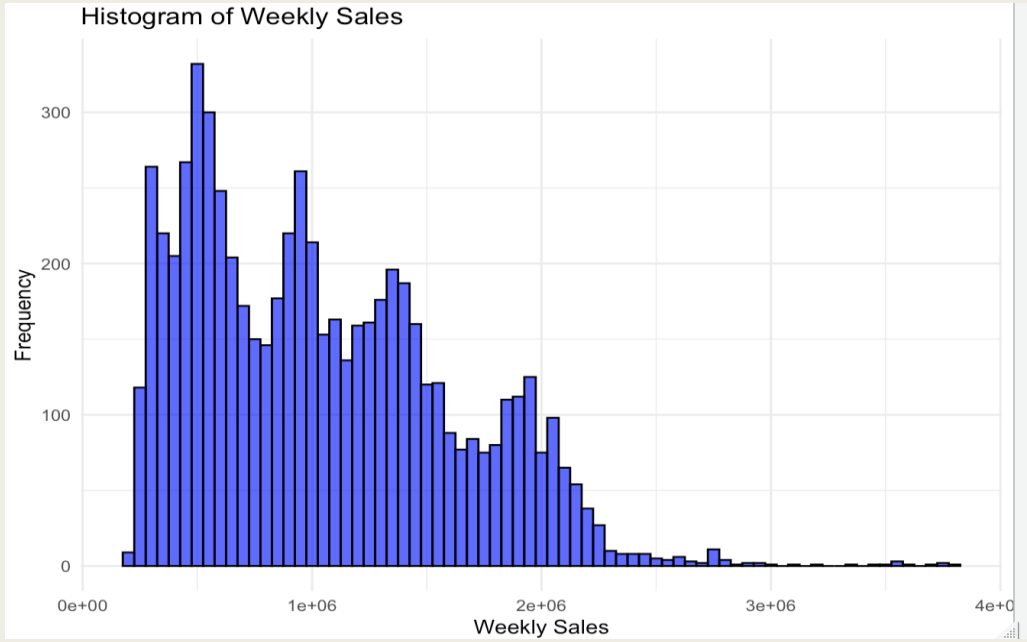
Store	Mean_Weekly_Sales	Median_Weekly_Sales	SD_Weekly_Sales	Mean_Fuel_Price	Median_Fuel_Price
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1555264.	1534850.	155981.	3.22	3.29
2	1925751.	1879107.	237684.	3.22	3.29
3	402704.	395107.	46320.	3.22	3.29
4	2094713.	2073951.	266201.	3.22	3.29
5	318012.	310338.	37738.	3.22	3.29
6	1564728.	1524390.	212526.	3.22	3.29

ℹ 7 more variables: SD_Fuel_Price <dbl>, Mean_CPI <dbl>, Median_CPI <dbl>, SD_CPI <dbl>, Mean_Unemployment <dbl>, Median_Unemployment <dbl>, SD_Unemployment <dbl>

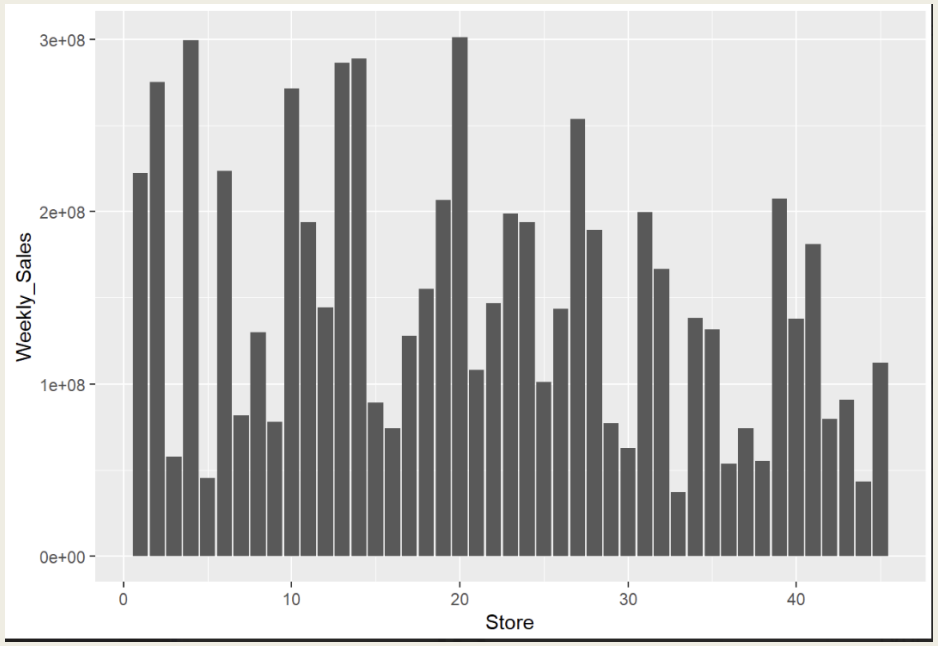
A tibble: 6 × 13

	Store	Mean_Weekly_Sales	Median_Weekly_Sales	SD_Weekly_Sales
	<dbl>	<dbl>	<dbl>	<dbl>
1	40	964128.	954234.	119002.
2	41	1268125.	1243815.	187907.
3	42	556404.	556046.	50263.
4	43	633325.	634815.	40598.
5	44	302749.	298080.	24763.
6	45	785981.	764014.	130169.

Weekly Sales Analysis for Walmart Stores

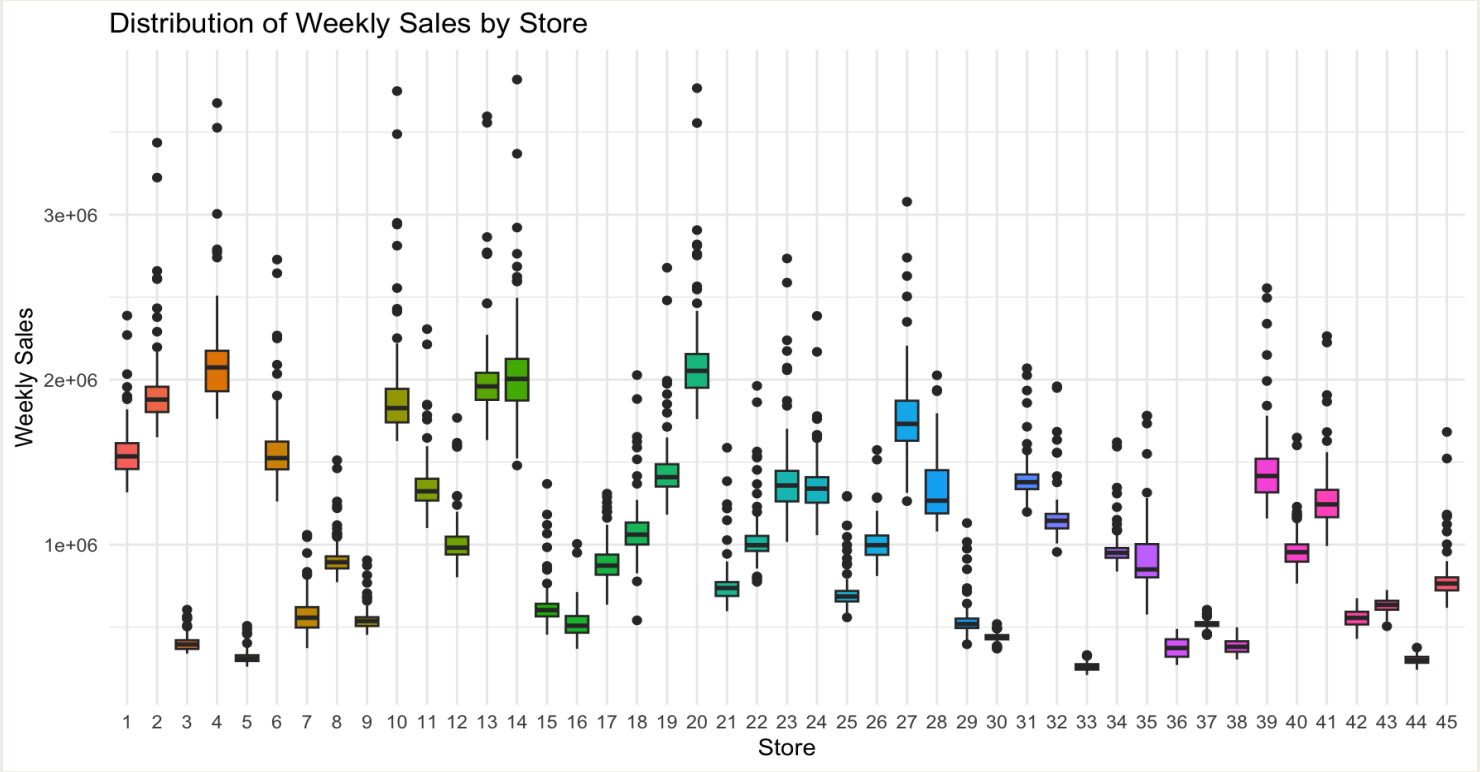


Distribution of weekly sales across all stores



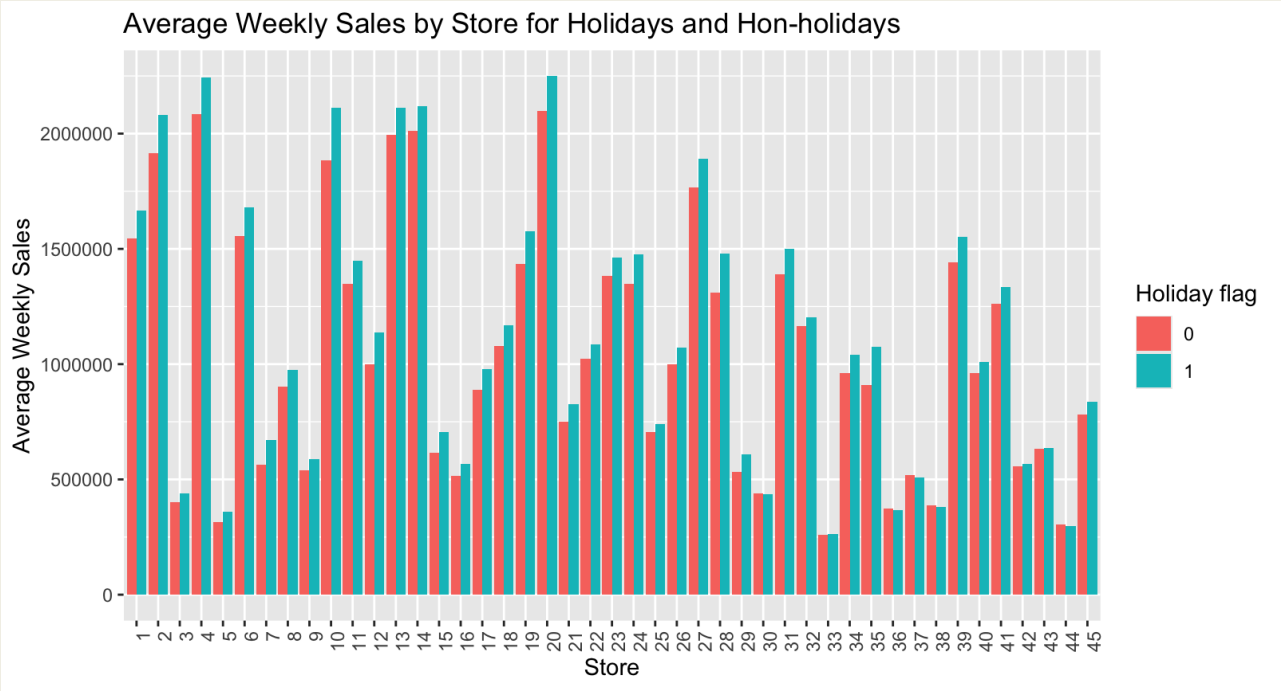
Weekly sales by Individual Store

Distribution of Weekly Sales across stores



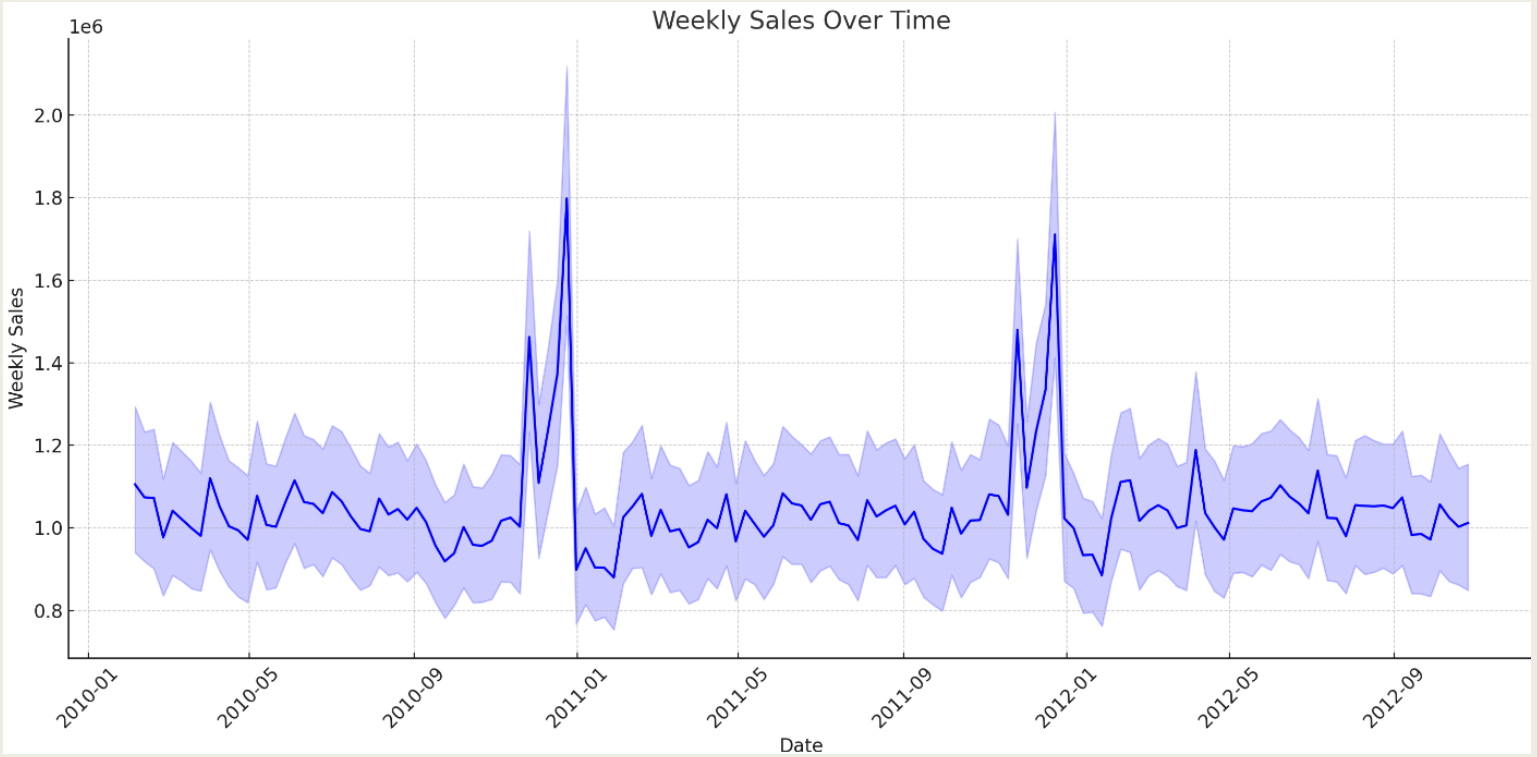
Boxplot shows sales performance across different stores

Sales Trends on Holidays



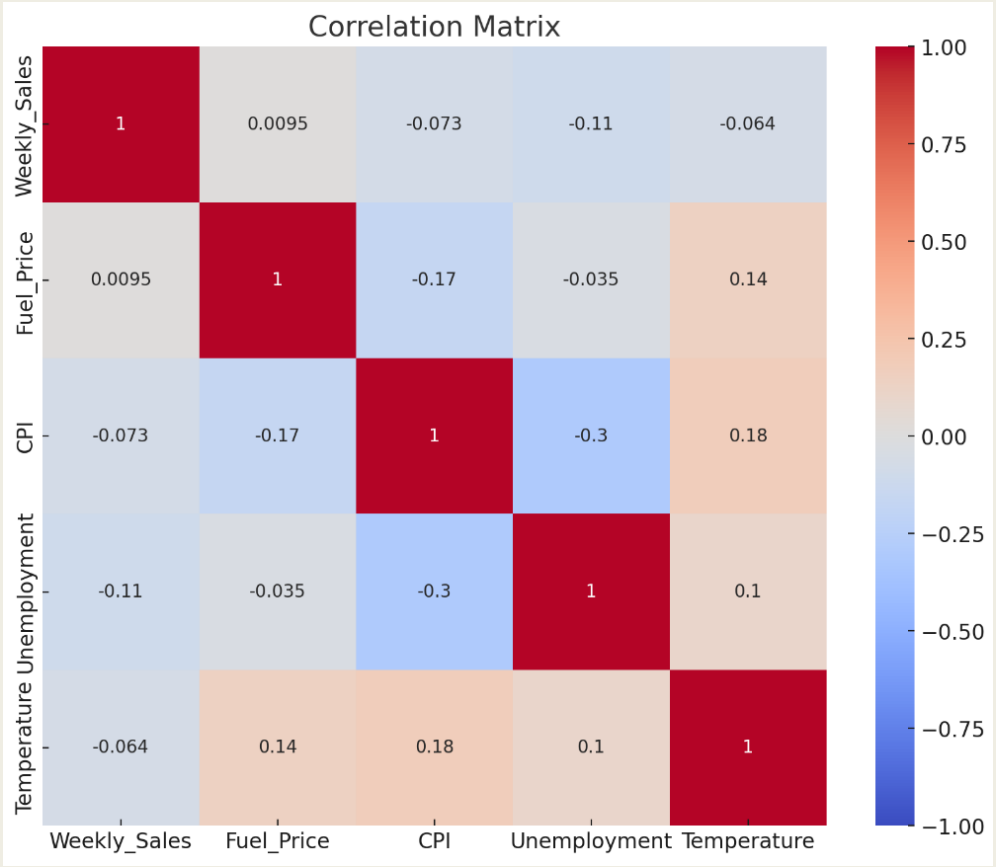
Average weekly sales on holidays are high across many stores

Time series of Weekly Sales



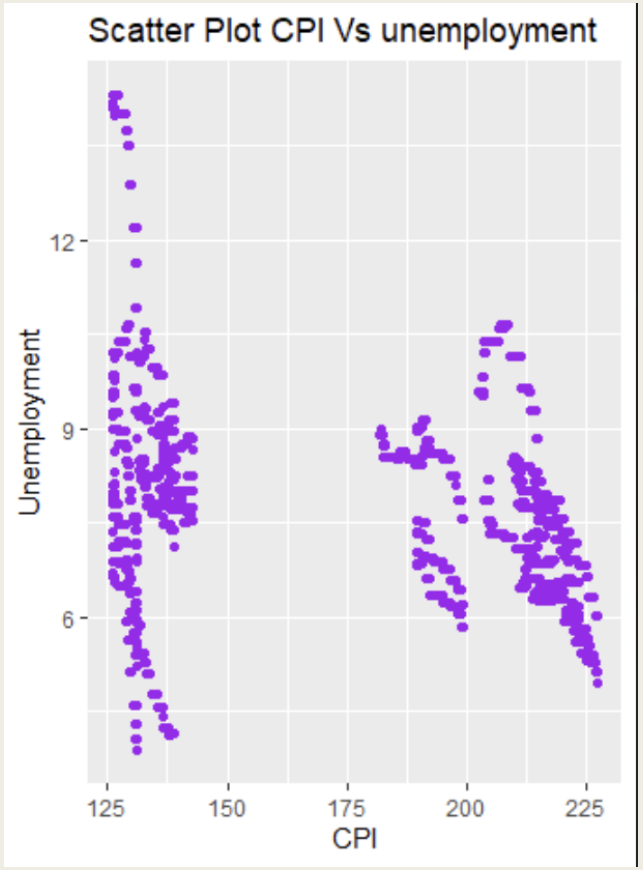
General Trend: The sales fluctuate within a certain range, with some periods showing higher variability than others. Seasonal Peaks: There are noticeable peaks around the end of each year (particularly around 2010, 2011, and 2012). This suggests a seasonal pattern, likely corresponding to holiday shopping periods.

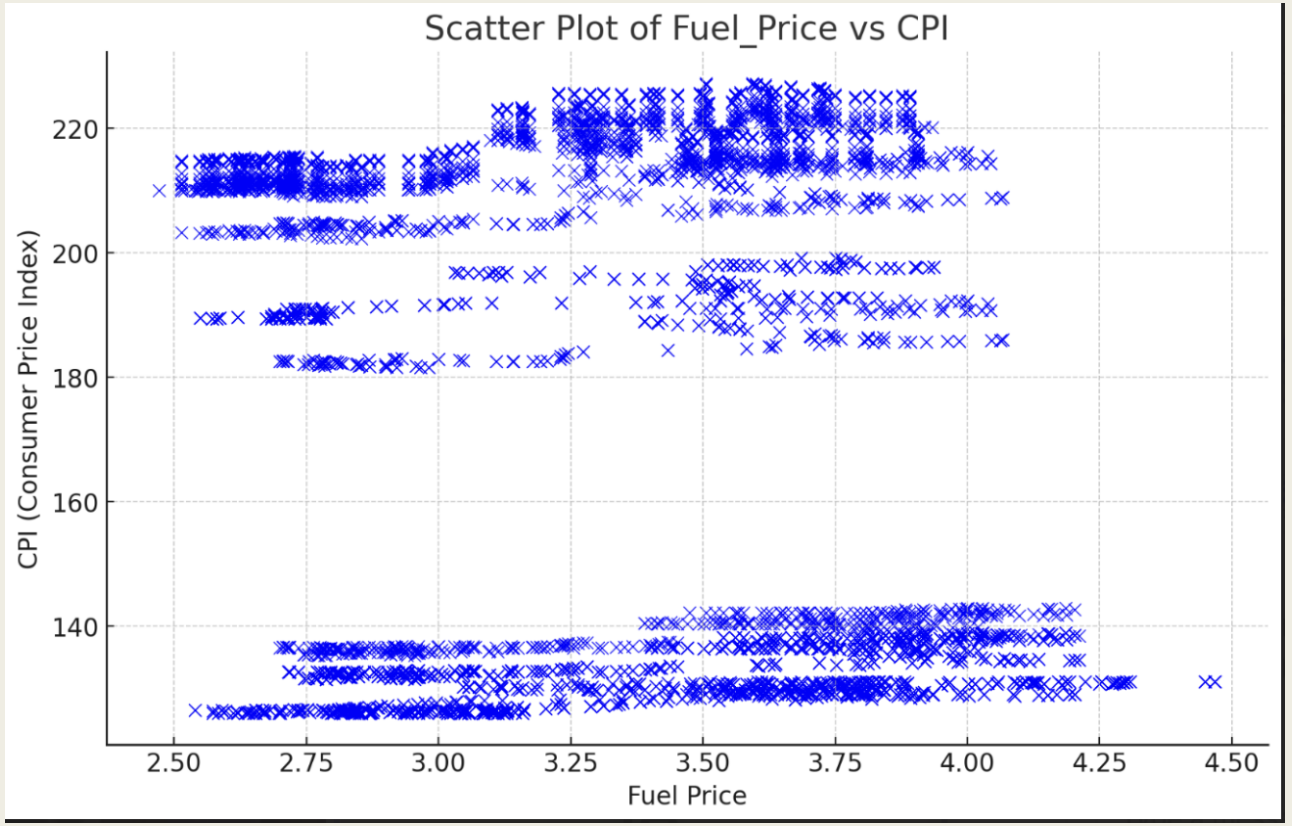
Relationship among key metrics



This heatmap shows that the correlations among the financial metrics are weak.

Impact of Unemployment rate and fuel prices on Consumer Price Index





The matrix shows that most variables in the Walmart dataset have weak or very weak correlations with each other. The only moderate correlation is between CPI and Unemployment (-0.3), which might indicate a significant economic relationship.

These findings suggest that the variables don't have a strong linear relationships.

Key insights

- The **variations in sales** shows that some stores performing well possibly due to **effective strategies** that could be **replicated across other locations**.
- **Holiday periods** present a valuable opportunity to **boost sales**. This suggests that **targeted promotions** during these times should be a priority.
- The analysis highlights the need for Walmart to adapt its **strategies** based on **sales trends** and prepare for both **peak** periods and subsequent **declines**.
- Further investigation is needed into **periods of low sales**, particularly the **sharp declines** to better **plan** and **manage** these **fluctuations**.

Recommendations

- Machine Learning for Customer Segmentation
- Interactive Displays and Experiences

List of Codes

```
# Create the heatmap using ggplot2

# Convert the necessary columns
data$Date <- as.Date(data$Date, format = "%d/%m/%Y")
data$Weekly_Sales <- as.numeric(gsub("[^0-9.-]", "", data$Weekly_Sales))
# Calculate the correlation matrix for relevant numerical columns
correlation_matrix <- cor(data[, c("Weekly_Sales", "Fuel_Price", "CPI", "Unemployment",
"Temperature")], use = "complete.obs")
ggplot(data = melted_cor_matrix, aes(x = Var1, y = Var2, fill = value)) +
 geom_tile(color = "white") +
scale_fill_gradient2(low = "red", high = "blue", mid = "white",
                midpoint = 0, limit = c(-1, 1), space = "Lab",
                name="Correlation") +
 results() +
 results(axis.text.x = element_text(angle = 45, vjust = 1,
                           size = 12, hjust = 1)) +
 coord_fixed() +
 labs(title = "Correlation Matrix Heatmap", x = "", y = "")

# Convert the Date column to Date format
data$Date <- dmy(data$Date)

# Convert Weekly_Sales to numeric (assuming it's in a currency format)
data$Weekly_Sales <- as.numeric(gsub("[^0-9.-]", "", data$Weekly_Sales))

# Create the time series plot
```

```
# Setting Up the Environment -  Install and load the  packages :        install.packages("tidyverse")
   install.packages("summarytools")
   install.packages("ggplot2")
   install.packages("dplyr")

   library(tidyverse)
   library(summarytools)
   library(ggplot2)
   library(dplyr)

# Create a scatter plot for CPI vs Fuel Price
ggplot(data, aes(x = Fuel Price, y = CPI)) + geom_point(color = "blue") + labs(title = "Scatter Plot of CPI vs Fuel Price ",
x = "Fuel Price ",
y = "CPI ") +
Results()

ggplot(data, aes(x = Date, y = Weekly_Sales)) +
 geom_line(color = "blue") +
 labs(title = "Weekly Sales Over Time",
     x = "Date",
     y = "Weekly Sales") +
 theme_minimal() +
 theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
 scale_x_date(date_labels = "%Y-%m")

# Calculate rolling mean and standard deviation (or confidence interval if needed)
data$Rolling_Mean <- zoo::rollmean(data$Weekly_Sales, k = 4, fill = NA)
data$Rolling_SD <- zoo::rollapply(data$Weekly_Sales, width = 4, FUN = sd, fill = NA)

# Create the time series plot with a shaded area for the confidence interval
ggplot(data, aes(x = Date, y = Weekly_Sales)) +
 geom_line(color = "blue") +
 geom_ribbon(aes(ymin = Weekly_Sales - Rolling_SD, ymax = Weekly_Sales + Rolling_SD),
          fill = "blue", alpha = 0.2) +
 labs(title = "Weekly Sales Over Time",
     x = "Date",
     y = "Weekly Sales") +
 results() +
 results(axis.text.x = element_text(angle = 45, hjust = 1)) +
 scale_x_date(date_labels = "%Y-%m")

#Create a bargraph
ggplot(data, aes(x=Store, y=Weekly_Sales)) +
 geom_bar(stat = "identity")
```

Appendix 1

- Histogram of Weekly sales: # Create a histogram for Weekly Sales
  ggplot(data, aes(x = Weekly_Sales)) + geom_histogram(binwidth = 50000, fill = "blue", color = "black", alpha = 0.7) + labs(title = "Histogram of Weekly Sales", x = "Weekly Sales", y = "Frequency") + theme_minimal()

highest_sales <- max(Walmart$Weekly_Sales)
print(highest_sales)

lowest_sales <- min(Walmart$Weekly_Sales)
print(lowest_sales)

Tibble
descriptive_stats_by_store <- data %>%
group_by(Store) %>%
summarise(
Mean_Weekly_Sales = mean(Weekly_Sales),
Median_Weekly_Sales = median(Weekly_Sales),
SD_Weekly_Sales = sd(Weekly_Sales),
Mean_Fuel_Price = mean(Fuel_Price),
Median_Fuel_Price = median(Fuel_Price),
SD_Fuel_Price = sd(Fuel_Price),
Mean_CPI = mean(CPI),
Median_CPI = median(CPI),
SD_CPI = sd(CPI),
Mean_Unemployment = mean(Unemployment),
Median_Unemployment = median(Unemployment),
SD_Unemployment = sd(Unemployment)
)
View(descriptive_stats_by_store)
head(descriptive_stats_by_store)
tail(descriptive_stats_by_store)

Appendix 2

```
# Create the heatmap using ggplot2
# Convert the necessary columnsdata$Date <- as.Date(data$Date, format =
"%d/%m/%Y")data$Weekly_Sales <- as.numeric(gsub("[^0-9.-]", "",
data$Weekly_Sales))# Calculate the correlation matrix for relevant numerical
columnscorrelation_matrix <- cor(data[, c("Weekly_Sales", "Fuel_Price", "CPI",
"Unemployment","Temperature")], use = "complete.obs")ggplot(data =
melted_cor_matrix, aes(x = Var1, y = Var2, fill = value)) + geom_tile(color = "white")
+scale_fill_gradient2(low = "red", high = "blue", mid = "white",                midpoint =
0, limit = c(-1, 1), space = "Lab",                name="Correlation") + results()
+  results(axis.text.x = element_text(angle = 45, vjust = 1,                        size = 12,
hjust = 1)) + coord_fixed() + labs(title = "Correlation Matrix Heatmap", x = "", y = "")

# Create a histogram for Weekly Sales
ggplot(data, aes(x = Weekly_Sales)) +
  geom_histogram(binwidth = 50000, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Histogram of Weekly Sales", x = "Weekly Sales", y = "Frequency") +
  theme_minimal()
```

Appendix 3

```
#Time serries

# Convert the Date column to Date format

data$Date <- dmy(data$Date)


# Convert Weekly_Sales to numeric (assuming it's in a currency format)

data$Weekly_Sales <- as.numeric(gsub("[^0-9.-]", "", data$Weekly_Sales))


# Create the time series plot

ggplot(data, aes(x = Date, y = Weekly_Sales)) +

 geom_line(color = "blue") +

labs(title = "Weekly Sales Over Time",

  x = "Date",

 y = "Weekly Sales") +

theme_minimal() +

 theme(axis.text.x = element_text(angle = 45, hjust = 1)) +

scale_x_date(date_labels = "%Y-%m")
```

```
#Boxplot
ggplot(data, aes(x =
factor(Store), y =
Weekly_Sales, fill =
factor(Store))) +
  geom_boxplot() +
  labs(title = "Distribution
of Weekly Sales by Store",
       x = "Store",
       y = "Weekly Sales") +
  theme_minimal() +
  theme(legend.position =
"none")
```

Appendix 4

```
# Holiday Trends

# Average sales by store during holiday and non-holiday weeks

store_holiday_sales <- data %>%

  group_by(Store, Holiday_Flag) %>%

  summarise(Avg_Weekly_Sales = mean(Weekly_Sales))

# Plot the sales

ggplot(store_holiday_sales, aes(x=factor(Store), y=Avg_Weekly_Sales,
         fill=factor(Holiday_Flag))) +

  geom_bar(stat="identity", position="dodge") +

  labs(x="Store", y="Average Weekly Sales", fill="Holiday flag",
         title="Average Weekly Sales by Store for Holidays and Hon-
         holidays") +

  theme(axis.text.x = element_text(angle=90, hjust=1))
```