setwd(choose.dir())

dataset = read.csv('Walmart\_Store\_sales.csv')

str(dataset)

View(dataset)

# Count the number of missing data if any

numberOfNA <- lapply(lapply(dataset, is.na), sum)

numberOfNA

### Finding the store with the maximum sales

max\_sales\_store <- aggregate(Weekly\_Sales ~ Store, data = dataset, sum)

summary(max\_sales\_store)

View(max\_sales\_store)

max\_sales\_store[with(max\_sales\_store, order(-Weekly\_Sales)), ]

# Store 20 has the highest total sales with 301397792$

### Finding the store with the maximum standard deviation

max\_sd <- aggregate(Weekly\_Sales ~ Store, data = dataset, sd)

View(max\_sd)

max\_sd[with(max\_sd, order(-Weekly\_Sales)),]

# Hence, Store 14 is the most unstable store with a standard deviation of 317569.9$

# A new data frame to compile data for answering further questions

df <- data.frame(Store = max\_sales\_store[,1], Weekly\_Sales\_Sum = max\_sales\_store[,2],

Max\_sd = max\_sd[,2])

str(df)

### Finding the Q3'2012 of each store to figure out which has a good quarterly growth rate

library(zoo) # Alternative libraries like lubridate works as well

library(sqldf)

dataset$subdate <- paste0(substr(dataset$Date, 7, length(dataset$Date)),

substr(dataset$Date, 3, 6),

substr(dataset$Date, 1, 2))

dataset$subdate <- as.Date(dataset$subdate, "%Y-%m-%d")

class(dataset$subdate)

dataset$Quarterly <- as.character(as.yearqtr(dataset$subdate,

format = "%Y-%m-%d"))

df <- cbind(df, sqldf('SELECT SUM(Weekly\_Sales) as "Q3\_Sales" FROM dataset

WHERE Quarterly LIKE "2012 Q3"

GROUP By Store'))

df <- cbind(df, sqldf('SELECT SUM(Weekly\_Sales) as "Q2\_Sales" FROM dataset

WHERE Quarterly LIKE "2012 Q2"

GROUP By Store'))

str(df)

sqldf('SELECT Store, ((Q3\_Sales - Q2\_Sales)/Q3\_Sales)\*100 as Q3\_Growth from df order by Q3\_Growth DESC LIMIT 15')

# Store 4 has the highest sales throughout Q3 with up to 27796792$ in revenue.

# However, in terms of growth, is considered a low performing considering its negative growth.

# On a different note, Store 7 grew by 11.8% outgrowing all other stores by a large margin.

# This is followed by stores 16 and 35 respectively. Additionally, only 10 stores experience a positive growth.

### Can some holidays have a negative impact on sales?

library(tidyverse)

dataset$subdate <- as.character(dataset$subdate)

# Filtering Sales by each holiday and storing the average into a new data frame

names(dataset)

holidays\_temp <- dataset %>% filter(subdate == '2012-02-10' | subdate == '2011-02-11'

| subdate == '2010-02-12' | subdate == '2013-02-8')

holidays\_df <- data.frame(Superbowl = mean(holidays\_temp$Weekly\_Sales))

holidays\_temp <- dataset %>% filter(subdate == '2013-09-06' | subdate == '2012-09-07'

| subdate == '2011-09-09' | subdate == '2010-09-10')

holidays\_df$LabourDay <- mean(holidays\_temp$Weekly\_Sales)

holidays\_temp <- dataset %>% filter(subdate == '2013-11-29' | subdate == '2012-11-23'

| subdate == '2011-11-25' | subdate == '2010-11-26')

holidays\_df$Thanksgiving <- mean(holidays\_temp$Weekly\_Sales)

holidays\_temp <- dataset %>% filter(subdate == '2013-12-27' | subdate == '2012-12-28'

| subdate == '2011-12-30' | subdate == '2010-12-31')

holidays\_df$Chirstmas <- mean(holidays\_temp$Weekly\_Sales)

View(holidays\_df)

str(holidays\_df)

# Filtering all holidays with flag == 0 (normal days) and find the mean

holidays\_temp <- dataset %>% filter( Holiday\_Flag == '0')

mean\_non\_holiday = mean(holidays\_temp$Weekly\_Sales)

# It is found that the average of weekly sales in normal days is 1041256$

holidays\_df\_normalized <- as.data.frame(lapply(holidays\_df, FUN = function(x) x - mean\_non\_holiday))

str(holidays\_df\_normalized)

rm(holidays\_temp, mean\_non\_holiday)

# Consequently, Christmas sales perform below average compared to the average sales on normal days

# This could be attributed to the overall spending mostly occurring during the first and last

# weeks of Christmas, as well as most people spending time with family and friends on other days.

# On the other hand, Thanksgiving had the highest average of 1471273$

# Overall, the average weekly sales on holidays outperform the average weekly sales on normal days

### Provide a monthly and semester view of sales in units and give insight

glimpse(dataset)

dataset$month <- substr(dataset$Date, 4, 5)

monthly\_sales\_df <- sqldf('SELECT month, SUM(Weekly\_Sales) as Monthly\_sales, AVG(Weekly\_Sales) as Monthly\_AVG,

STDEV(Weekly\_Sales) as Monthly\_sd FROM dataset GROUP By month')

monthly\_sales\_df[order(-monthly\_sales\_df$Monthly\_sales),]

monthly\_sales\_df[order(-monthly\_sales\_df$Monthly\_AVG),]

monthly\_sales\_df[order(-monthly\_sales\_df$Monthly\_sd),]

# Looking at the monthly sales, we can see that the average monthly sales across the stores are highest

# during July. Additionally, the lowest sales are on January.

# The trend aligns with expectations, since July is often considered month of holidays for students.

# On the other hand, January is preceded by celebrations and parties taken in December for New year's eve and

# Christmas.

# However, as noted previously we can see the standard deviation of December is also the greatest due to

# The imbalance spending throughout the month.

# Additional insight looking at the monthly sales of each stores rather than of all combined.

monthly\_sales\_dfstores <- sqldf('SELECT Store,month, SUM(Weekly\_Sales) as Monthly\_sales, AVG(Weekly\_Sales) as Monthly\_AVG,

STDEV(Weekly\_Sales) as Monthly\_sd FROM dataset GROUP By Store, month')

monthly\_sales\_dfstores[order(-monthly\_sales\_dfstores$Monthly\_sales), ][1:15,]

# Interestingly, Store 14 holds the highest monthly sale and as found before is the store with highest deviation.

# However, Store 20 holds more spots at the top, as expected considering its found to be the most lucrative store.

# Additionally, July is seen multiple times as a top performing month.

# Still, April and June are also months performing considerably well too.

# Semester View of sales

dataset$semester <- ifelse(as.numeric(dataset$month) < 7, 1 , 2 )

yearly\_sales\_df <- sqldf('SELECT semester, SUM(Weekly\_Sales) as Semester\_sales, AVG(Weekly\_Sales) as Semester\_AVG,

STDEV(Weekly\_Sales) as Semester\_sd FROM dataset GROUP By semester')

glimpse(yearly\_sales\_df)

barplot(yearly\_sales\_df[, 2] ~ yearly\_sales\_df[, 1], main = 'Semesterly Revenue', col=c("blue","red"),

xlab = "Semester", ylab = "Total Revenue", legend= c("Semester 1", "Semester 2"))

# Semester two has a total of 95.8 million dollars more than semester one in revenue.

# Additionally, It is also higher in terms of average revenue per sale and standard deviation of sales.

# Hence, the trend follows the previous findings considering that July and December are both in semester 2.

# Similarly, a low performing month like January is in the 1st semester.

### Building a linear regression model

library(lubridate)

glimpse(dataset)

# Stores 20 and 4, the stores with the highest sales were chosen (since they are more likely to behave similarly)

dataset\_model <- dataset[dataset$Store == 20 | dataset$Store == 4,]

dataset\_model <- dataset\_model[,-9:-12]

View(dataset\_model)

# Days are converted to numbers

dataset\_model$Date <- dmy(dataset\_model$Date)

dataset\_model$Date <- yday(dataset\_model$Date)

glimpse(dataset\_model)

library(corrplot)

corr <- cor(dataset\_model[, c(2,3,5,6,7,8)])

corrplot(corr, method = "color", outline = T, cl.pos = 'n', rect.col = "black", tl.col = "indianred4", addCoef.col = "black", number.digits = 2, number.cex = 0.60, tl.cex = 0.7, cl.cex = 1, col = colorRampPalette(c("green4","white","red"))(100))

# Selecting the variables that have some level of correlation

corr <- cor(dataset\_model[, c(3,6,7,8)])

corrplot(corr, method = "color", outline = T, cl.pos = 'n', rect.col = "black", tl.col = "indianred4", addCoef.col = "black", number.digits = 2, number.cex = 0.60, tl.cex = 0.7, cl.cex = 1, col = colorRampPalette(c("green4","white","red"))(100))

# Splitting the data (no need to check for na since it was checked initially in the main dataset)

library(caTools)

set.seed(145)

modelling <- dataset\_model[, c(3,6,7,8)]

sample <- sample.split(modelling, SplitRatio = 0.7)

trainingSet <- subset(modelling, sample == T)

testSet <- subset(modelling, sample == F)

# Create model

model <- lm(formula = Weekly\_Sales ~ . , data = trainingSet)

summary(model)

# Drop Fuel\_Price and check the model again

trainingSet$Fuel\_Price <- NULL

model = lm(formula = Weekly\_Sales ~ . , data = trainingSet)

summary(model)

# It is clear the CPI and Unemployment affect the sales of Walmart

library(ggplot2)

# Visualizing the training set results

y\_pred\_train = predict(model, newdata = trainingSet)

summary(y\_pred\_train)

ggplot() +

geom\_point(aes(x=trainingSet$Weekly\_Sales,y=y\_pred\_train)) +

xlab('actual\_price') +

ylab('predicted\_price')+

ggtitle('comparison of train data')

# Visualizing the test set results

y\_pred\_test = predict(model, newdata = testSet)

summary(y\_pred\_test)

ggplot() +

geom\_point(aes(x=testSet$Weekly\_Sales,y=y\_pred\_test)) +

xlab('actual\_price') +

ylab('predicted\_price')+

ggtitle('comparison of test data')

# Parameters to validate the accuracy of the model and improvise.

library(MLmetrics)

MAPE(y\_pred\_test,testSet$Weekly\_Sales)

RMSE(y\_pred\_test,testSet$Weekly\_Sales)

# A MAPE value of 7% is obtained, which is acceptable by industry standards as it is close to 5%

# Removing extreme outliers (found after the next approach) improves the MAPE to 5.8%

################################################

# One more prediction model will be performed. In this case, rather than grouping stores together

# The outliers shall be removed to reduce the number of errors.

boxplot(dataset$Weekly\_Sales)

summary(dataset$Weekly\_Sales)

max\_sale <- max(dataset$Weekly\_Sales)

boxplot(dataset$Weekly\_Sales[dataset$Weekly\_Sales < max\_sale\*0.8])

# Usually 1.5x values away from the mean are removed. However, this dataset does not contain

# outliers below (considering its nature it will become negative and non-intuitive)

# Meanwhile, selecting 0.8\* of the mean kept many clustered outliers, while removing extreme valued ones.

dataset\_model <- dataset[dataset$Weekly\_Sales < max\_sale\*0.8,]

dataset\_model <- dataset\_model[,-9:-12]

View(dataset\_model)

# Days are converted to numbers

dataset\_model$Date <- dmy(dataset\_model$Date)

dataset\_model$Date <- yday(dataset\_model$Date)

glimpse(dataset\_model)

corr <- cor(dataset\_model[, c(2,3,5,6,7,8)])

corrplot(corr, method = "color", outline = T, cl.pos = 'n', rect.col = "black", tl.col = "indianred4", addCoef.col = "black", number.digits = 2, number.cex = 0.60, tl.cex = 0.7, cl.cex = 1, col = colorRampPalette(c("green4","white","red"))(100))

# Selecting the variables that have some level of correlation

corr <- cor(dataset\_model[, c(3,6,7,8)])

corrplot(corr, method = "color", outline = T, cl.pos = 'n', rect.col = "black", tl.col = "indianred4", addCoef.col = "black", number.digits = 2, number.cex = 0.60, tl.cex = 0.7, cl.cex = 1, col = colorRampPalette(c("green4","white","red"))(100))

# Splitting the data

modelling <- dataset\_model[, c(3,6,7,8)]

sample <- sample.split(modelling, SplitRatio = 0.7)

trainingSet = subset(modelling, sample == T)

testSet = subset(modelling, sample == F)

# Create model

model = lm(formula = Weekly\_Sales ~ . , data = trainingSet)

summary(model)

# CPI and unemployment both show more significance in the model this time.

# Drop Fuel\_Price and check the model again

trainingSet$Fuel\_Price <- NULL

model = lm(formula = Weekly\_Sales ~ . , data = trainingSet)

summary(model)

# Visualizing the training set results

y\_pred\_train = predict(model, newdata = trainingSet)

summary(y\_pred\_train)

ggplot() +

geom\_point(aes(x=trainingSet$Weekly\_Sales,y=y\_pred\_train)) +

xlab('actual\_price') +

ylab('predicted\_price')+

ggtitle('comparison of train data')

# Visualizing the test set results

y\_pred\_test = predict(model, newdata = testSet)

summary(y\_pred\_test)

ggplot() +

geom\_point(aes(x=testSet$Weekly\_Sales,y=y\_pred\_test)) +

xlab('actual\_price') +

ylab('predicted\_price')+

ggtitle('comparison of test data')

# Parameters to validate the accuracy of the model and improvise.

library(MLmetrics)

MAPE(y\_pred\_test,testSet$Weekly\_Sales)

RMSE(y\_pred\_test,testSet$Weekly\_Sales)

# A MAPE value of 65.8% is found. This indicates that the grouping of stores according to behavior is necessary

################################################

# Same as first model, but with extreme outliers removed

### Building a linear regression model

library(lubridate)

glimpse(dataset)

# Stores 20 and 4, the stores with the highest sales were chosen (since they are more likely to behave similarly)

dataset\_model <- dataset[dataset$Store == 20 | dataset$Store == 4,]

dataset\_model <- dataset\_model[,-9:-12]

View(dataset\_model)

# Days are converted to numbers

dataset\_model$Date <- dmy(dataset\_model$Date)

dataset\_model$Date <- yday(dataset\_model$Date)

boxplot(dataset\_model$Weekly\_Sales)

dataset\_model<- dataset\_model[dataset\_model$Weekly\_Sales < max(dataset\_model$Weekly\_Sales)\*0.8,]

glimpse(dataset\_model)

# Splitting the data (no need to check for na since it was checked initially in the main dataset)

modelling <- dataset\_model[, c(3,6,7,8)]

sample <- sample.split(modelling, SplitRatio = 0.7)

trainingSet <- subset(modelling, sample == T)

testSet <- subset(modelling, sample == F)

# Create model

model <- lm(formula = Weekly\_Sales ~ . , data = trainingSet)

summary(model)

# Drop Fuel\_Price and check the model again

trainingSet$Fuel\_Price <- NULL

model = lm(formula = Weekly\_Sales ~ . , data = trainingSet)

summary(model)

# It is clear the CPI and Unemployment affect the sales of Walmart

library(ggplot2)

# Visualizing the training set results

y\_pred\_train = predict(model, newdata = trainingSet)

summary(y\_pred\_train)

ggplot() +

geom\_point(aes(x=trainingSet$Weekly\_Sales,y=y\_pred\_train)) +

xlab('actual\_price') +

ylab('predicted\_price')+

ggtitle('comparison of train data')

# Visualizing the test set results

y\_pred\_test = predict(model, newdata = testSet)

summary(y\_pred\_test)

ggplot() +

geom\_point(aes(x=testSet$Weekly\_Sales,y=y\_pred\_test)) +

xlab('actual\_price') +

ylab('predicted\_price')+

ggtitle('comparison of test data')

# Parameters to validate the accuracy of the model and improvise.

library(MLmetrics)

MAPE(y\_pred\_test,testSet$Weekly\_Sales)

RMSE(y\_pred\_test,testSet$Weekly\_Sales)

# A MAPE value of 5.8% is obtained, which is acceptable