Retail Analysis of Walmart

Report and Source Code:

Import the dataset Walmart_Sales_store.csv from the location it saved in your computer

```
read.csv("Walmart_Store_sales.csv")->walmart
View(walmart)
head(walmart)

library(dplyr)
library(corrplot)
library(reshape2)
library(caTools)
```

Analysis Tasks and Basic Statistics tasks

Q.1. Which store has maximum sales?

```
walmart%>%group_by(Store)%>%
  summarise(Sales = sum(Weekly_Sales))%>%
  mutate(Rank = rank(desc(Sales)))%>%filter(Rank==1)
```

Conclusion: Store 20 is found to have maximum sales.

Q.2. Which store has maximum standard deviation i.e., the sales vary a lot. Also, find out the coefficient of mean to standard deviation.

```
walmart%>%group_by(Store)%>%
  summarise(Dev_Sales = sd(Weekly_Sales), Mean_Sales = mean(Weekly_Sales))%>%
  mutate(cv = Dev_Sales*100/Mean_Sales)%>%
  mutate(Rank = rank(desc(Dev_Sales)))%>%filter(Rank==1)
```

Conclusion: Store 14 is found to have maximum standard deviation.

Q.3. Which store/s has good quarterly growth rate in Q3'2012?

```
as.Date(walmart$Date, format = "%d-%m-%Y")->walmart$Date
format.Date(x = walmart$Date, "%m")->walmart$Month
format.Date(x = walmart$Date, "%Y")->walmart$Year
data.frame(lapply(walmart[,c("Month", "Year")], as.numeric))->walmart[,c("Month", "Year")]
str(walmart)
v = c(1,3, 6, 9, 12)
cut(walmart$Month, breaks = v, labels = c(1,2,3,4),
    include.lowest = T, ordered_result = T)->walmart$Quarters
walmart%>%filter(Year ==2012)%>%group_by(Store,Quarters)%>%
  summarize(Sales = sum(Weekly_Sales))%>%filter(Quarters%in% c(2,3))->res
dcast(res, Store ~ Quarters, value.var = "Sales")->d
head(d)
names(d)[c(2,3)] = c("Q2", "Q3")
d%>%mutate(Growth = (Q3-Q2)*100/Q2)%>%mutate(Rank =
rank(desc(Growth)))%>%filter(Rank==1)
```

Conclusion: Store 7 is found to have good quarterly growth rate.

Q.4. Some holidays have a negative impact on sales. Find out holidays which have higher sales than the mean sales in non-holiday season for all stores together.

```
walmart = as_tibble(walmart)
```

Mean sales in non-holiday season for all stores together:

```
walmart%>%filter(Holiday_Flag==0)%>%summarize(mean(Weekly_Sales))->x
x[1,1]
```

Those holidays which have higher sales than the above value:

```
walmart%>%filter(Holiday_Flag==1)%>%filter(Weekly_Sales>x[1,1])
```

Q.5. Provide a monthly and semester view of sales in units and give insights.

```
 walmart%>%group_by(Month, Year)%>%
summarise(AveSales = mean(Weekly_Sales))%>%arrange(Year, Month)%>%
mutate(M_Y = paste(Month, Year, sep = "/"))
```

Building the Linear Regression model

Q. Restructure dates as 1 for 5 Feb 2010 (starting from the earliest date in order)

```
as.numeric(format.Date(walmart$Date, "%d"))->walmart$Day
walmart$Day
ifelse(walmart$Day<=7, 1,
    ifelse(walmart$Day<=14,2,
        ifelse(walmart$Day<=21,3,
            ifelse(walmart$Day<=28,4,5))))->walmart$WeekNum
```

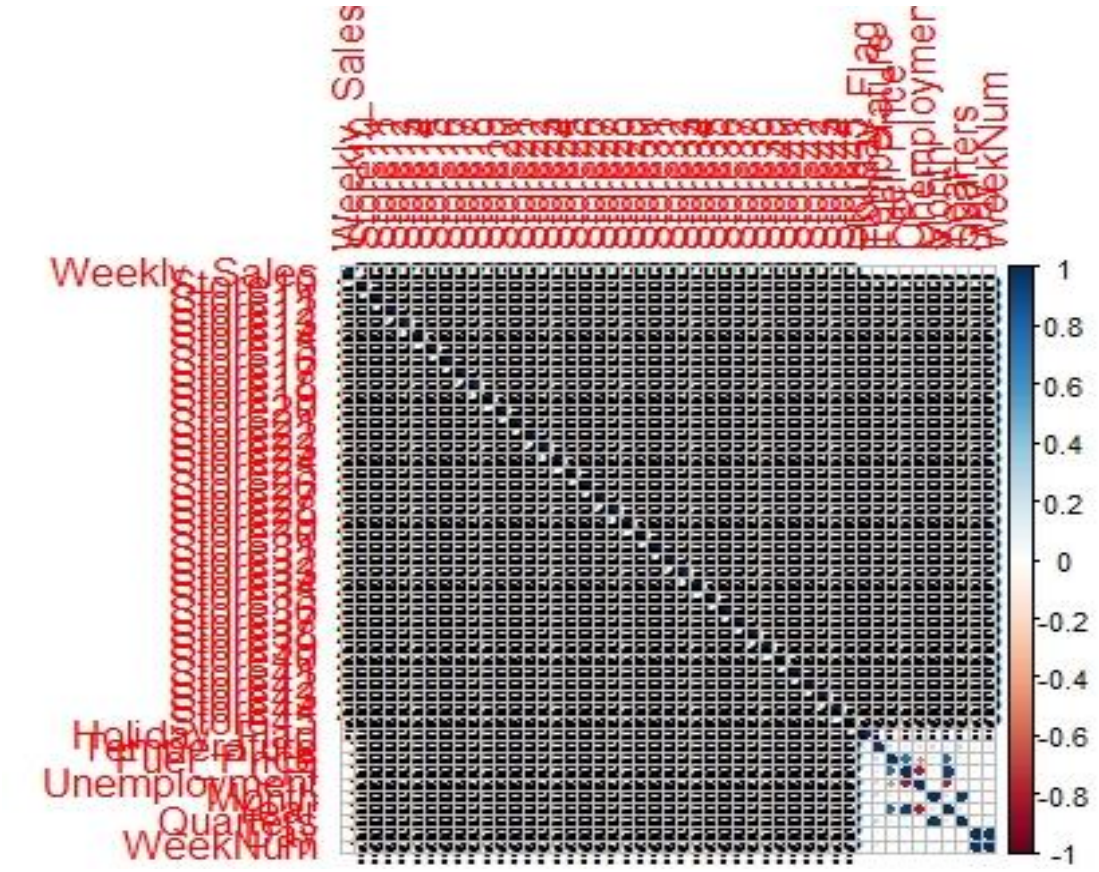
Q. Change dates into days by creating new variable.

```
as.numeric(format.Date(walmart$Date, "%d"))->walmart$Day
format.Date(walmart$Date, "%A")->walmart$WeekDay
walmart$WeekDay
```

Here, the requirement is to build a Linear Regression Model for the continuous response variable 'Weekly_Sales' with respect to the other remaining variables in the given dataset. Now proceeding towards the first step; Data Pre-processing which involves data cleaning. Thus, will look for rows with data-entry error i.e. in this case negative data.

```
which(is.na(walmart$Weekly_Sales))
```

Since there is no negative data, no need to delete rows. The following plot shows the relationship between each variable in the dataset which helps in analysing variables with relevant correlation with the response variable i.e. 'Weekly_Sales'



After analysing the dataset, the required pre-processing tasks have been performed. In this case, not much was required as the data is already minimal. Some Feature Engineering tasks have been performed by creating new columns namely 'Day', 'Month', 'Year', 'WeekNum', 'Quarters', 'WeekDay' for obtaining correlation between the response variable 'Weekly_Sales' as observed above in the plot (Data Visualisation).Now, after performing Inferential statistics for feature selection and proceeding to Dummy encoding of the nominal categorical variables (numerical dataset). The following steps have been performed to obtain the best fit model with the highest accuracy possible.

Converting the dataset columns to numerical data to perform required operations:

```
str(walmart)

walmart$Date = NULL

walmart$Holiday_Flag = as.numeric(walmart$Holiday_Flag)

walmart$Quarters = as.numeric(walmart$Quarters)

walmart$Store = as.factor(walmart$Store)

w = walmart

model.matrix(w$Weekly_Sales~., data = w)->mat

mat
```

```
mat = mat[,-1]

df = as.data.frame(mat)

head(df)

cbind(w$Weekly_Sales, df)->w

names(w)[1] = "Weekly_Sales"

head(w)
```

Splitting into Train-Test:

Trainning the model using the training data to learn the parameters of the model:

y = w0 + w1X1 + w2X2 + w3X3 + w4X4 + w5X5 where parameters of the model are the weights or w_i: coefficients and intercept terms.

```
set.seed(100)

split = sample.split(Y = w$Weekly_Sales, SplitRatio = .7)

training = w[split,]

test = w[!split,]
```

```
lm(formula = Weekly_Sales~., data = training)->model

summary(model)

step(model, direction = "both")->mod

summary(mod)
```

Here Adjusted R squared value of more than 0.9 is obtained which reflects more than 90% accuracy of the model as known higher the Adjusted R Squared value better is the accuracy of the model. Model has learned the parameters from the training data, now will test the performance of the model on test data. Check the errors and the accuracy of the model on the test data which is an unseen data for the model. This will help in understanding how good will the model be if deployed the model on completely unseen data. Now calculating the error ‘e’, sum of squared error ‘sse’, ‘sst’ and rsqaured value for determing the accuracy of the model.

```
predict(mod, newdata = test)->p

e = p - test$Weekly_Sales

sse = sum(e^2)
```

```
sst = sum((test$Weekly_Sales-mean(training$Weekly_Sales))^2)

rsq = 1-sse/sst

rsq
```

Here the 'rsq' value is obtained less than 0.09 which implies less variation and thus less error and hence the accuracy of the model increases. Once satisfied with the performance of the model, will save the model for the future deployment and prediction on the unseen data.

Linear Regression Model for Store 1

Similar steps are performed as done for all the stores.

```
walmart%>%filter(Store == 1)->w

model.matrix(w$Weekly_Sales~.,data=w)->mat

mat = mat[,-1]

df = as.data.frame(mat)

head(df)

cbind(w$Weekly_Sales, df)->w

names(w)[1] = "Weekly_Sales"

head(w)


set.seed(100)

split = sample.split(Y = w$Weekly_Sales, SplitRatio = .7)

training = w[split,]

test = w[!split,]


lm(formula = Weekly_Sales~., data = training)->model

summary(model)

step(model, direction = "both")->mod

summary(mod)
```

Adjusted R Square value of more than 0.36 is obtained reflecting the accuracy. Here we observe a lower value as compared to one in previous model due to filtering of stores which may have possibly been significant in positively affecting the sales.

```
predict(mod, newdata = test)->p

e = p - test$Weekly_Sales

sse = sum(e^2)

sst = sum((test$Weekly_Sales-mean(training$Weekly_Sales))^2)

rsq = 1-sse/sst

rsq
```

Here the 'rsq' value of less than 0.2 is obtained showing minimum error. Thus, model can be concluded to be with good accuracy.

Demand Forecasting for Store 1

```
head(walmart)

walmart%>%filter(Store==1)%>%select(Weekly_Sales)->store1

store1

length(store1$Weekly_Sales)

plot(1:length(store1$Weekly_Sales), store1$Weekly_Sales)
```