```
#Analysis Walmart Store Sales
#Date Created: 30/5/2021
#Author: Sukanto Mukherjee

getwd()
setwd('/users/sukanto/WD/Walmart_StoreSale_Analysis/
Walmart_StoreSales_Analysis')
install.packages("fpp2")

#Installing and loading packages
my_packages <- c("ggplot2","lubridate","dplyr", "plyr","ggplot2",
                 "lubridate","raster","zoo","sp",
                 "usdm","lmtest","forecast")
lapply(my_packages, require, character.only = TRUE)
stores <- read.csv('Walmart_Store_sales.csv')
head(stores)
summary(stores)
colnames(stores)
str(stores)

#Data preprocessing and Exploratory data analysis

#sum(is.na(stores))
#duplicated(stores)


#Formatting date column


stores$Date <- as.Date(stores$Date, format = c("%m-%d-%Y"))
str(stores)

#Which store had maximum sales?
each_store <- aggregate(Weekly_Sales ~ Store, stores,sum)
each_store <- arrange(each_store, desc(Weekly_Sales))
max(each_store)
options(scipen = 999)
jpeg('max_store_sale.jpg')
ggplot(each_store, aes(Store, Weekly_Sales)) +
  geom_bar(stat = 'identity', color = ' dark blue',
           fill = ' dark blue')
dev.off()

#Which store had maximum standard deviation and finding coeff of mean to sd

each_store_sd <- aggregate(Weekly_Sales~Store,stores, sd)
each_store_sd <- rename(each_store_sd, c(Weekly_Sales = 'SD_Sales'))
max(each_store_sd)
each_store_mean <- aggregate(Weekly_Sales~Store, stores, mean)
each_store_mean <- rename(each_store_mean, c(Weekly_Sales = 'Mean_Sales'))
```

```
each_store_mean_sd <- cbind(each_store_mean, each_store_sd)
each_store_mean_sd_coeff <- transform(each_store_mean_sd, Coeff = SD_Sales/
Mean_Sales)

#Which store had good quarterly growth rate in Q32012?

quarter_store <- transform(stores, Q_Flag= ifelse((Date>='2012-04-01' & Date<=
'2012-06-30'),"Q2_2012",
                                    ifelse((Date>='2012-07-01' & Date<=
'2012-09-30'),"Q3_2012","-")))
# confirming start and end date for each quarter
aggregate(Date ~ Q_Flag, quarter_store, min)
aggregate(Date ~ Q_Flag, quarter_store, max)

# summarizing and then reshaping
quarter_store_sum <- aggregate(Weekly_Sales~Store+Q_Flag,quarter_store,sum)
str(quarter_store_sum)

quarter_store_sum_t <- reshape(quarter_store_sum,idvar="Store",timevar
='Q_Flag',direction="wide")
View(quarter_store_sum_t)
quarter_store_sum_t_GR <- transform(quarter_store_sum_t,
                           GR=((Weekly_Sales.Q3_2012-Weekly_Sales.Q2_2012)/
Weekly_Sales.Q2_2012))
jpeg('store_growth_q3.jpg')
ggplot(quarter_store_sum_t_GR, aes(Store, GR))+geom_bar(stat='identity',
                                        color='dark red',
                                        fill = 'dark red')
dev.off()

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

non_holiday_Sales <- filter(stores,Holiday_Flag==0)
View(non_holiday_Sales)
Avg_non_holiday_Sales <- mean(non_holiday_Sales$Weekly_Sales)
Declining_Holiday_Sales <- filter(stores,Weekly_Sales>Avg_non_holiday_Sales &
Holiday_Flag==1)
unique(Declining_Holiday_Sales$Date)

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

stores_month_year <- transform(stores,Year =as.numeric(format(Date,"%Y"))
                               ,Month =as.numeric(format(Date,"%m")))
Summarized_View <-
aggregate(Weekly_Sales~Month+Year,stores_month_year,sum)
Insight_data <- arrange(Summarized_View,desc(Weekly_Sales))
jpeg('month_sale.jpg')
ggplot(Insight_data, aes(Month, Weekly_Sales)) + geom_bar(stat='identity',
```

```
                                        fill = 'navy')
dev.off()
jpeg('year_sale.jpg')
ggplot(Insight_data, aes(Year, Weekly_Sales)) + geom_bar(stat='identity',
                                        fill = 'dark red')
dev.off()


# we had experienced maximum sales in Dec 2010 and post that it was in June
2012.

#Sales Forecast
#Approach 1: Linear Model

fit <- lm(Weekly_Sales ~ Holiday_Flag + Temperature + Fuel_Price
          + CPI + Unemployment, stores)
summary(fit)
#Dropping the insignificant vars ie Temperature and Fuel Price
fit1 <- lm(Weekly_Sales ~ Holiday_Flag + CPI + Unemployment, stores)
summary(fit1)

#The p-values are too low to support any hypothesis with a linear model

#Approach 2 : Time series model
# visually identifying if data is fit for time series
store1 <- aggregate(Weekly_Sales~Date,stores,sum)
plot(store1, type = 'l')
class(store1)
str(store1)

#preparing data for ARIMA model
library(dplyr)
store_month_year <- transform(stores,Year =as.numeric(format(Date,"%Y"))
                              ,Month =as.numeric(format(Date,"%m")))
store_month_year_filtered <- store_month_year %>% dplyr::
select(Weekly_Sales,Year,Month)

# rolling up sales at month level
Walmart_Rolledup <- aggregate(Weekly_Sales~Year+Month,
                        store_month_year_filtered,sum)
# sorting in year and month order
Walmart_sorted <- arrange(Walmart_Rolledup,Year,Month)
# creating a Column with month and year of sale
Walmart_TS <- transform(Walmart_sorted,
                    Time_Of_Sale = as.Date(paste(Year,"-",Month,"-",1,sep=""),
                                           format="%Y-%m-%d"))[,c(4,3)]

# Build up ARIMA model to forecast last 6 months i.e as in input utilize only till
# Building ARIMA model
Walmart_ARIMA <- auto.arima(Walmart_TS[1:30,2])
```

```
Walmart_ARIMA = arima(Walmart_TS[1:30,2],order=c(2,1,2))
Forecasted_Sale <- forecast(Walmart_ARIMA,h=6)
Forecasted_Sale
jpeg('sales_forecast_6m.jpg')
plot(Forecasted_Sale)
dev.off()
# 6 months forecast
Forecasted_Sales <- as.data.frame(Forecasted_Sale)
Forecasted_Sales_6m <- Forecasted_Sales[,1]
View(Forecasted_Sales_6m)
# 6 m actual
Actual_Sales_6m <- Walmart_TS[31:36,]
# concatenating 6 m forecast and actual
Actual_vs_Forecast_last_6_m <- cbind(Forecasted_Sales_6m,Actual_Sales_6m)
View(Actual_vs_Forecast_last_6_m)
p1<-ggplot(Actual_vs_Forecast_last_6_m, aes(Time_Of_Sale,
Forecasted_Sales_6m))+
  geom_line()+ggtitle("Six-Month Sales Forecast")
p1
p2<-ggplot(Actual_vs_Forecast_last_6_m, aes(Time_Of_Sale, Weekly_Sales))+
  geom_line()+ggtitle("Actual Sales")
p2
install.packages('patchwork')
library(patchwork)
p1+p2
Actual_vs_Forecast_last_6_m_deviation <-
transform(Actual_vs_Forecast_last_6_m,
                                      Errors = abs(Forecasted_Sales_6m-Weekly_Sales)/
Weekly_Sales)
MAPE <- mean(Actual_vs_Forecast_last_6_m_deviation$Errors)
MAPE
```