```
##### Clear Workspace #####
rm(list=ls())


##### Options with digits is used to control the decimal places in R Console View #####
options(digits=10)


##### Options with Sigfig is used to control the decimal places in Tibble #####
options(pillar.sigfig = 10)


##### Set Working Directory #####
setwd("C:/Users/SANDERAT/OneDrive - Capgemini/Learn/PGP DS/1. R Programming/Workspace/
Project")


##### Load Libraries #####
library(plyr)
library(dplyr)
library(tibble)
library(lubridate)
library(corrplot)


##### Data Import & Cleaning #####
WallMart_Data <- read.csv("Walmart_Store_sales.csv")
head(WallMart_Data)
str(WallMart_Data)
View(WallMart_Data)
colnames(WallMart_Data)


# Code to convert "Date" column from String #####
# WallMart_Data$Date <- as.Date(WallMart_Data$Date, format = c("%d-%m-%Y"))
d <- as.Date(WallMart_Data$Date, format="%d-%m-%Y")
d[is.na(d)] <- as.Date(WallMart_Data$Date[is.na(d)], format="%d/%m/%Y")
WallMart_Data$Date <- d
str(WallMart_Data)
View(WallMart_Data)


# Code to convert "Store" column from String to Factor #####
WallMart_Data$Store <- as.factor(WallMart_Data$Store)
levels(WallMart_Data$Store)
class(WallMart_Data$Store)




##### Question 1: Which store has maximum sales? #####
# Answer 1: Store 20 has maximum Weekly Sales as 301397792.46

# Code
Storewise_Sales <- arrange(aggregate(Weekly_Sales~Store, WallMart_Data,
sum),desc(Weekly_Sales))
Storewise_Sales[1,]

# Output
#   Store   Weekly_Sales
#1  20      301397792.5
```

```
##### Question 2: Which store has maximum standard deviation i.e., the sales vary a lot. Also,
find out the coefficient of mean to standard deviation. #####
# Answer 2: Store 14 has maximum standard deviation as 317569.95 and Store 35 has maximum
coefficient of mean to standard deviation.

# Code
Stores_Mean <- aggregate(Weekly_Sales~Store,WallMart_Data,mean)
Stores_Mean <- plyr::rename(Stores_Mean, c("Weekly_Sales"="Mean_Sales"))

Stores_SD <- aggregate(Weekly_Sales~Store, WallMart_Data, sd)
Stores_SD <- plyr::rename(Stores_SD, c("Weekly_Sales"="SD_Sales"))
Stores_SD <- arrange(Stores_SD, desc(SD_Sales))[1,]
Stores_SD

Stores_Mean_SD <- cbind(Stores_Mean,SD_Sales=Stores_SD$SD_Sales)
Stores_Mean_SD_Coeff <- mutate(Stores_Mean_SD, Mean_Coeff=(SD_Sales/Mean_Sales))
Stores_Mean_SD_Coeff <- arrange(Stores_Mean_SD_Coeff, desc(Mean_Coeff))
Stores_Mean_SD_Coeff

# Output - Mean
#   Store   SD_Sales
#1  14      317569.9495

# Output - Coefficient
#     Store   Mean_Sales    SD_Sales      Mean_Coeff
#1    33      259861.6920   317569.9495   1.2220729689




##### Question 3: Which store/s has good quarterly growth rate in Q3'2012? #####
# Answer 3: Store 7 has top quarterly growth rate as 13.30%.

# Code
WallMartData_Quarter <- WallMart_Data %>%
  mutate(Quarter = ifelse(Date >= "2012-04-01" & Date <= "2012-06-30", "Q2_2012", ifelse(Date
>= "2012-06-01" & Date <= "2012-09-30", "Q3_2012", "-")))

WallMartData_Quarterly_Sales <- aggregate(Weekly_Sales~Store+Quarter, WallMartData_Quarter,
sum)

WallMartData_Quarterly_Sales <- reshape(WallMartData_Quarterly_Sales, idvar = "Store", timevar
= "Quarter", direction = "wide")
WallMartData_Quarterly_Sales_GR <- mutate(WallMartData_Quarterly_Sales,
GrowthRate=(Weekly_Sales.Q3_2012 - Weekly_Sales.Q2_2012)/Weekly_Sales.Q2_2012)
arrange(WallMartData_Quarterly_Sales_GR, desc(GrowthRate))[1,]

# Output
#   Store   Weekly_Sales.-  Weekly_Sales.Q2_2012  Weekly_Sales.Q3_2012    GrowthRate
# 1 7       66044628.48     7290859.27            8262787.39              0.1333077603




##### Question 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? #####
# Answer 4: Below Holiday Dates have higher sales than the mean sales in non-holiday season.
# "2010-02-12" "2010-09-10" "2010-11-26" "2010-12-31" "2011-02-11" "2011-09-09" "2011-11-25"
"2011-12-30" "2012-02-10" "2012-09-07"

# Code
Non_Holiday_Sales <- filter(WallMart_Data, Holiday_Flag==0)
Mean_Non_Holiday_Sales <- mean(Non_Holiday_Sales$Weekly_Sales)
Mean_Non_Holiday_Sales
#Non Holiday Mean Sales = 1041256.38
```

```
Holiday_Sales <- filter(WallMart_Data, Weekly_Sales>Mean_Non_Holiday_Sales & Holiday_Flag==1)
Holiday_Sales
unique(Holiday_Sales$Date)

# Output
# "2010-02-12" "2010-09-10" "2010-11-26" "2010-12-31" "2011-02-11" "2011-09-09" "2011-11-25"
"2011-12-30" "2012-02-10" "2012-09-07"




##### Question 5: Provide a monthly and semester view of sales in units and give insights?
#####
# Answer 5: Yearly and Monthly view of sales provided below with December-2010 being highest
sales month with 288760532.7 as sales figure.
# January-2011 was worst performing sales month. Wallmart needs to adopt the similar marketing
strategy used in December-2010 and avoid the strategy used in January-2011 to enhance the
sales.

# Code
WallMart_Data_Monthly_Yearly_sales <- mutate(WallMart_Data,
Sales_Year=as.numeric(format(Date,"%Y")),Sales_Month=as.numeric(format(Date,"%m")))

WallMart_Data_Monthly_Yearly_sales <- aggregate(Weekly_Sales~Sales_Year+Sales_Month,
WallMart_Data_Monthly_Yearly_sales, sum)
WallMart_Data_Monthly_Yearly_sales
arrange(WallMart_Data_Monthly_Yearly_sales, desc(Weekly_Sales))

WallMart_Data_Semester <- mutate(WallMart_Data, Semester = semester(WallMart_Data$Date,2010))
WallMart_Data_Semester_Sales <- aggregate(Weekly_Sales~Semester, WallMart_Data_Semester, sum)
WallMart_Data_Semester_Sales

# Output - Monthly View
#   Sales_Year    Sales_Month   Weekly_Sales
#1  2010          12            288760532.7

# Output - Semester View
#   Semester  Weekly_Sales
#1  2010.1    982622260.3



##### Linear Model #####

corrplot(cor(WallMart_Data[-c(1,2)]))
cordata <- cor(WallMart_Data[-c(1,2)])
corrplot(cordata, method = c("number"), type = "lower")

WallMart_Data_LM <- lm(Weekly_Sales ~ Holiday_Flag + Temperature + Fuel_Price + CPI +
Unemployment, WallMart_Data)
summary(WallMart_Data_LM)

#Removing insignificant variables from linear model such as Temperature & Fuel_Price
WallMart_Data_LM <- lm(Weekly_Sales ~ Holiday_Flag + CPI + Unemployment, WallMart_Data)
summary(WallMart_Data_LM)


yday(WallMart_Data$Date - yday(WallMart_Data$Date-1)[1])
WallMart_Data <- add_column(WallMart_Data, Days = yday(WallMart_Data$Date -
yday(WallMart_Data$Date-1)[1]))

#Rebuilding model with Days
WallMart_Data_LM <- lm(Weekly_Sales ~ Holiday_Flag + CPI + Unemployment + Days, WallMart_Data)
summary(WallMart_Data_LM)
```