**Programming Codes:**

#Reading Walmart Data and loading libraries

rm(list=ls())

library(dplyr)

library(ggplot2)

library(zoo)

library(lubridate)

library(forecast)

library(tseries)

setwd("E:\\Simplilearn\\Data Science with R\\Project\\Walmart")

walmart<- read.csv("Walmart\_Store\_sales.csv")

head(walmart)

**# Basic Statistics tasks: -**

#conversion of date into date format

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

head(walmart)

# 1. Which store has maximum sales.

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

max\_sales\_store[(max\_sales\_store$Weekly\_Sales == max(max\_sales\_store$Weekly\_Sales)),]

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

mean\_sales<-mean(walmart$Weekly\_Sales)

sd\_sales<-sd(walmart$Weekly\_Sales)

zscore<- (walmart$Weekly\_Sales-mean\_sales)/sd\_sales

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

max\_sd\_store[(max\_sd\_store$Weekly\_Sales == max(max\_sd\_store$Weekly\_Sales)),]

max\_zscore\_store <- aggregate(zscore ~ Store, FUN = max, data = walmart)

max\_zscore\_store[(max\_zscore\_store$zscore == max(max\_zscore\_store$zscore)),]

Coefficient\_variation<-sd\_sales/mean\_sales\*100

Coefficient\_variation

# 3. Which store/s has a good quarterly growth rate in Q3’2012.

yq<-as.yearqtr(walmart$Date)

format(yq, format = "%y/0%q")

max\_sales\_qtr <- aggregate(Weekly\_Sales ~ Store+yq, FUN = sum, data = walmart)

max\_sales\_qtr[(max\_sales\_qtr$Weekly\_Sales == max(max\_sales\_qtr$Weekly\_Sales)),]

qtrdata = max\_sales\_qtr %>%

group\_by(Store) %>%

arrange(yq) %>%

mutate(qOverq=Weekly\_Sales/lag(Weekly\_Sales,1))

subset\_Qtr3 <- subset(qtrdata, yq == "2012 Q3")

max\_sales\_Q3\_2012 <- aggregate(qOverq ~ Store+yq, FUN = max, data = subset\_Qtr3)

max\_sales\_Q3\_2012[(max\_sales\_Q3\_2012$qOverq == max(max\_sales\_Q3\_2012$qOverq)),]

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

Nonholiday<- subset(walmart, Holiday\_Flag == FALSE)

mean\_nonholiday<-mean(Nonholiday$Weekly\_Sales)

mean\_nonholiday

Holiday<- subset(walmart, Holiday\_Flag == TRUE)

diff\_meansales<-Holiday$Weekly\_Sales-mean\_nonholiday

Holiday<-cbind(Holiday,diff\_meansales)

Holiday\_meansales <- aggregate(diff\_meansales ~ Date, FUN = sum, data = Holiday)

max\_meansales\_date <- Holiday\_meansales [with(Holiday\_meansales ,order(-diff\_meansales)),]

max\_meansales\_date <- max\_meansales\_date[1:10,]

max\_meansales\_date

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

monthly<- as.yearmon(walmart$Date, format = "%d-%m-%Y")

monthly<-cbind(walmart,monthly)

monthmax<- aggregate(Weekly\_Sales ~ Store+monthly, FUN = sum, data = monthly)

monthmin<- aggregate(Weekly\_Sales ~ Store+monthly, FUN = sum, data = monthly)

monthmax1<- aggregate(Weekly\_Sales ~ monthly, FUN = sum, data = monthly)

monthmin1<- aggregate(Weekly\_Sales ~ monthly, FUN = sum, data = monthly)

monthmax[(monthmax$Weekly\_Sales == max(monthmax$Weekly\_Sales)),]

monthmin[(monthmin$Weekly\_Sales == min(monthmin$Weekly\_Sales)),]

monthmax1[(monthmax1$Weekly\_Sales == max(monthmax1$Weekly\_Sales)),]

monthmin1[(monthmin1$Weekly\_Sales == min(monthmin1$Weekly\_Sales)),]

#semester

dt<- as.yearmon(walmart$Date, format = "%d-%m-%Y")

Date2period <- function(x = walmart$Date, period = 6, sep = " S") {

paste(as.integer(dt), (cycle(dt) - 1) %/% period + 1, sep = sep)

}

semester <- Date2period(walmart$Date)

semester1<-cbind(walmart,semester)

semestermax<- aggregate(Weekly\_Sales ~ Store+semester, FUN = sum, data = semester1)

semestermin<- aggregate(Weekly\_Sales ~ Store+semester, FUN = sum, data = semester1)

semestermax1<- aggregate(Weekly\_Sales ~ semester, FUN = sum, data = semester1)

semestermin1<- aggregate(Weekly\_Sales ~ semester, FUN = sum, data = semester1)

semestermax[(semestermax$Weekly\_Sales == max(semestermax$Weekly\_Sales)),]

semestermin[(semestermin$Weekly\_Sales == min(semestermin$Weekly\_Sales)),]

semestermax1[(semestermax1$Weekly\_Sales == max(semestermax1$Weekly\_Sales)),]

semestermin1[(semestermin1$Weekly\_Sales == min(semestermin1$Weekly\_Sales)),]

**# Statistical Model: -**

# For Store 1 – Build prediction models to forecast demand

# 6. Linear Regression – Utilize variables like date and restructure dates as 1 for 5 Feb 2010(starting from the earliest date in order). Hypothesize if CPI, unemployment, and fuel price have any impact on sales.

Store1<- subset(walmart, Store == 1)

Store1<- Store1 %>% mutate(WeekNum = 1:n())

Store1%>%select(WeekNum,Date,Weekly\_Sales)

head(Store1)

model<-lm(Weekly\_Sales ~ CPI+Unemployment+Fuel\_Price, data = Store1)

summary (model)

# 7. Time series forecasting model – Hypothesize if the data is fit for time series analysis – check for white noise probability test

# 8. Make adjustments in historical data for events like holidays, if applicable

# 9. Build ARIMA model to forecast 6 months i.e., input utilize only till April 2012.

# 10. Predict next 6 months i.e., June to Oct 2010. Check for MAPE.

# 11. Select the model which gives best accuracy.

# Time series forecasting model:-

ts\_data <- ts(Store1$Weekly\_Sales,start=c(2010,6), end=c(2012,43), frequency = 52)

ts\_data

plot.ts(ts\_data)

abline(reg = lm(ts\_data~time(ts\_data)))

plot(decompose(ts\_data))

kpss.test(ts\_data)

adf.test(ts\_data,alternative="stationary")

nsdiffs(ts\_data)

ndiffs(ts\_data)

kpss.test(diff(log(ts\_data)))

adf.test(diff(log(ts\_data)),alternative="stationary")

Acf(diff(log(ts\_data)), lag.max=52,plot=TRUE, main = "ACF Plot")

Pacf(diff(log(ts\_data)),lag.max=52,plot=TRUE, main = "PACF Plot")

### Preparing Train and Test data.

train\_sales<-window(diff(log(ts\_data)),start=c(2010,6),end=c(2012,17),frequency=52)

test\_sales<-window(diff(log(ts\_data)),start=c(2012,18),end=c(2012,43), frequency=52)

# Fit the ARMA model

fit = auto.arima(train\_sales)

summary(fit)

box<-Box.test(residuals(fit),lag=52,type="Ljung-Box")

box

# Predict using the model

pred <- forecast(fit,h=25)

plot(pred,main="Predicition from Auto ARIMA for Weekly Sales")

lines(test\_sales,col="Red",lwd=3)

accuracy(pred,test\_sales)

-------------------------------------------------------------------The End-------------------------------------------------------------------