



Retail Analysis with Walmart Data

Business Analytic Foundation with R Tools- Question

Abstract

One of the leading retail stores in the US, Walmart, would like to predict the sales and demand accurately. There are certain events which impact sales on each day.

Using the data of 45 stores provided, build a forecast model which gives best accuracy.

Presented by: Ankita Agarwal

Problem Statement:

One of the leading retail stores in the US, Walmart, would like to predict the sales and demand accurately. There are certain events and holidays which impact sales on each day. There are sales data available for 45 stores of Walmart. The business is facing a challenge due to unforeseen demands and runs out of stock sometimes, due to the inappropriate machine learning algorithm. An ideal ML algorithm will predict demand at different points of time covering seasonality and ingest factors like economic conditions including CPI, Unemployment Index, etc.

Walmart runs several promotional markdown events throughout the year. These markdowns precede prominent holidays, the four largest of all, which are the Super Bowl, Labour Day, Thanksgiving, and Christmas. The weeks including these holidays are weighted five times higher in the evaluation than non-holiday weeks. Part of the challenge presented by this competition is modeling the effects of markdowns on these holiday weeks in the absence of complete/ideal historical data.

Detailed description of the given dataset:

This is the historical data which covers sales from 2010-02-05 to 2012-11-01 with the following fields:

Store: the store number

Date: the week of sales

Weekly_Sales: sales for the given store

Holiday_Flag: whether the week is a special holiday week 1 – Holiday week 0 – Non-holiday week

Temperature: Temperature on the day of sale

Fuel_Price: Cost of fuel in the region

CPI: Prevailing consumer price index

Unemployment: Prevailing unemployment rate

Holiday Events: Super Bowl: 12-Feb-10, 11-Feb-11, 10-Feb-12, 8-Feb-13 Labour Day: 10-Sep-10, 9-Sep-11, 7-Sep-12, 6-Sep-13 Thanksgiving: 26-Nov-10, 25-Nov-11, 23-Nov-12, 29-Nov-13 Christmas: 31-Dec-10, 30-Dec-11, 28-Dec-12, 27-Dec-13

To Analyze:

Basic Statistics tasks: -

1. Which store has maximum 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.
3. Which store/s has a good quarterly growth rate in Q3'2012.
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.
5. Provide a monthly and semester view of sales in units and give insights.

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.
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.

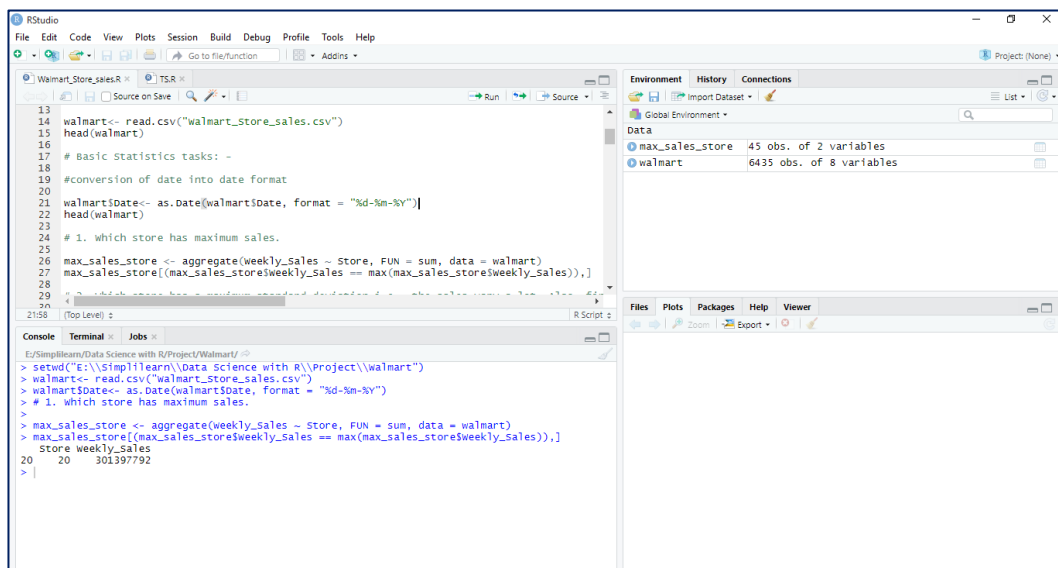
Select the model which gives best accuracy.

Analysis and Interpretations:

Basic Statistics tasks: -

1. Which store has maximum sales.

To start with, converted the date into a more readable format. In order to check which store has the maximum sales in the data provided, used the aggregate-sum function.



```
13 walmart<- read.csv("walmart_store_sales.csv")
14 head(walmart)
15
16 # Basic Statistics tasks: -
17
18 #conversion of date into date format
19
20 walmart$Date<- as.Date(walmart$Date, format = "%d-%m-%Y")
21 head(walmart)
22
23 # 1. Which store has maximum sales.
24
25 max_sales_store <- aggregate(weekly_sales ~ Store, FUN = sum, data = walmart)
26 max_sales_store[(max_sales_store$weekly_sales == max(max_sales_store$weekly_sales)),]
27
28 # 2. Which store has a maximum standard deviation of weekly sales.
29
30
```

Environment

| Object | Class | Attributes |
|-----------------|------------|--------------------------|
| Data | data.frame | |
| max_sales_store | data.frame | 45 obs. of 2 variables |
| walmart | data.frame | 6435 obs. of 8 variables |

Console

```
> setwd("E:\\Simplilearn\\Data Science with R\\Project\\walmart")
> walmart<- read.csv("walmart_store_sales.csv")
> walmart$Date<- as.Date(walmart$Date, format = "%d-%m-%Y")
> # 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)),]
  Store weekly_sales
20      301397792
> |
```

Interpretation:

It can clearly be said that Store 20 has the maximum sales in the given time period.

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.

In order to check which store has the maximum standard deviation in sales in the data provided, used the aggregate-sd function. We also find the coefficient of mean to standard deviation i.e., coefficient of variation.

```

# 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$weekly_sales)),]
      Store weekly_sales
14      14      317569.9
>
> max_zscore_store <- aggregate(zscore ~ Store, FUN = max, data = walmart)
> max_zscore_store[(max_zscore_store$zscore == max(max_zscore_store$zscore)),]
      Store    zscore
14      14  4.911207
>
> Coefficient_variation<-sd_sales/mean_sales*100
> Coefficient_variation
[1] 53.90502

```

Environment pane values:

| Object | Class | Attributes |
|------------------|------------|--------------------------|
| max_sales_store | data.frame | 45 obs. of 2 variables |
| max_sd_store | data.frame | 45 obs. of 2 variables |
| max_zscore_store | data.frame | 45 obs. of 2 variables |
| walmart | data.frame | 6435 obs. of 8 variables |

Interpretation:

It can clearly be seen that Store 14 has the maximum deviation from mean of sales for the given time period. A z-score almost equal to +5, indicated that it is 5 standard deviation above the mean. Also, the coefficient of variation is 53.9% i.e., the standard deviation is 53.9% of the mean.

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

In order to check which store has a good quarterly growth rate in Q3'2012, firstly we convert the date into quarter using the `as.yearqtr` function. We then group the sales by stores and arrange them by quarters in order to get quarter over quarter growth rate. To specifically check the growth rate for Q3'2012, we then subset the data for the said period and then use `aggregate-max` function to find the store with good quarterly growth rate.

```

44 # 3. Which store/s has a good quarterly growth rate in Q3'2012.
45
46 yq<-as.yearqtr(walmart$Date)
47 head(format(yq, format = "%y/%q"))
48
49 max_sales_qtr <- aggregate(weekly_Sales ~ Store+yq, FUN = sum, data = walmart)
50 max_sales_qtr[(max_sales_qtr$weekly_Sales == max(max_sales_qtr$weekly_Sales)),]
51
52 qtrdata = max_sales_qtr %>%
53   group_by(Store) %>%
54   e.c.
55
56 4438 (Top Level) >

```

Console Output:

```

> # 3. Which store/s has a good quarterly growth rate in Q3'2012.
>
> yq<-as.yearqtr(walmart$Date)
> head(format(yq, format = "%y/%q"))
[1] "10/01" "10/01" "10/01" "10/01" "10/01" "10/01"
>
> max_sales_qtr <- aggregate(weekly_Sales ~ Store+yq, FUN = sum, data = walmart)
> max_sales_qtr[(max_sales_qtr$weekly_Sales == max(max_sales_qtr$weekly_Sales)),]
  Store      yq weekly_Sales
155   20 2010 Q4    32573123
>
> qtrdata = max_sales_qtr %>%
+   group_by(Store) %>%
+   arrange(yq) %>%
+   mutate(qoverq=weekly_Sales/lag(weekly_Sales,1))
>
> subset_qtr3 <- subset(qtrdata, yq == "2012 Q3")
>
> max_sales_Q3_2012 <- aggregate(qoverq ~ Store+yq, FUN = max, data = subset_qtr3)
> max_sales_Q3_2012[(max_sales_Q3_2012$qoverq == max(max_sales_Q3_2012$qoverq)),]
  Store      yq      qoverq
7    2012 Q3    1.133308

```

Interpretation:

Store 20 has the max sales of 32573123 in Q4'2010. However, as per the problem statement, that the quarterly growth rate in Q3'2012 is the highest in Store 7 with a quarterly growth rate of 1.33.

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.

Firstly, we create two subsets where in 1 is the data for Holiday and 1 for Non-Holiday. We then calculate the mean sales for Non-Holiday season. Then calculate the difference from mean for Holiday subset. We then aggregate the difference of mean by Date as we need to find the holidays that have higher sales than the mean sales in a non-holiday season.

```

60 max_sales_q3_2012[(max_sales_q3_2012$quarter == max(max_sales_q3_2012$quarter))].
61
62 # 4. Some holidays have a negative impact on sales. Find out holidays which have higher
63 sales than the mean sales in a non-holiday season for all stores together.
64 nonholiday <- subset(walmart, holiday_flag == FALSE)
65
> # 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$weekly_sales)
> mean_nonholiday
[1] 1041256
> holiday <- subset(walmart, holiday_flag == TRUE)
> diff_meansales <- holiday$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
  date diff_meansales
7 2011-11-25 19737068.15
3 2010-11-26 18964466.13
9 2012-02-10 3152870.81
1 2010-02-12 1480140.52
10 2012-09-07 1473522.20
5 2011-02-11 479655.68
6 2011-09-09 -93309.58
8 2011-12-30 -814076.07
2 2010-09-10 -1222139.27
4 2010-12-31 -6424018.11

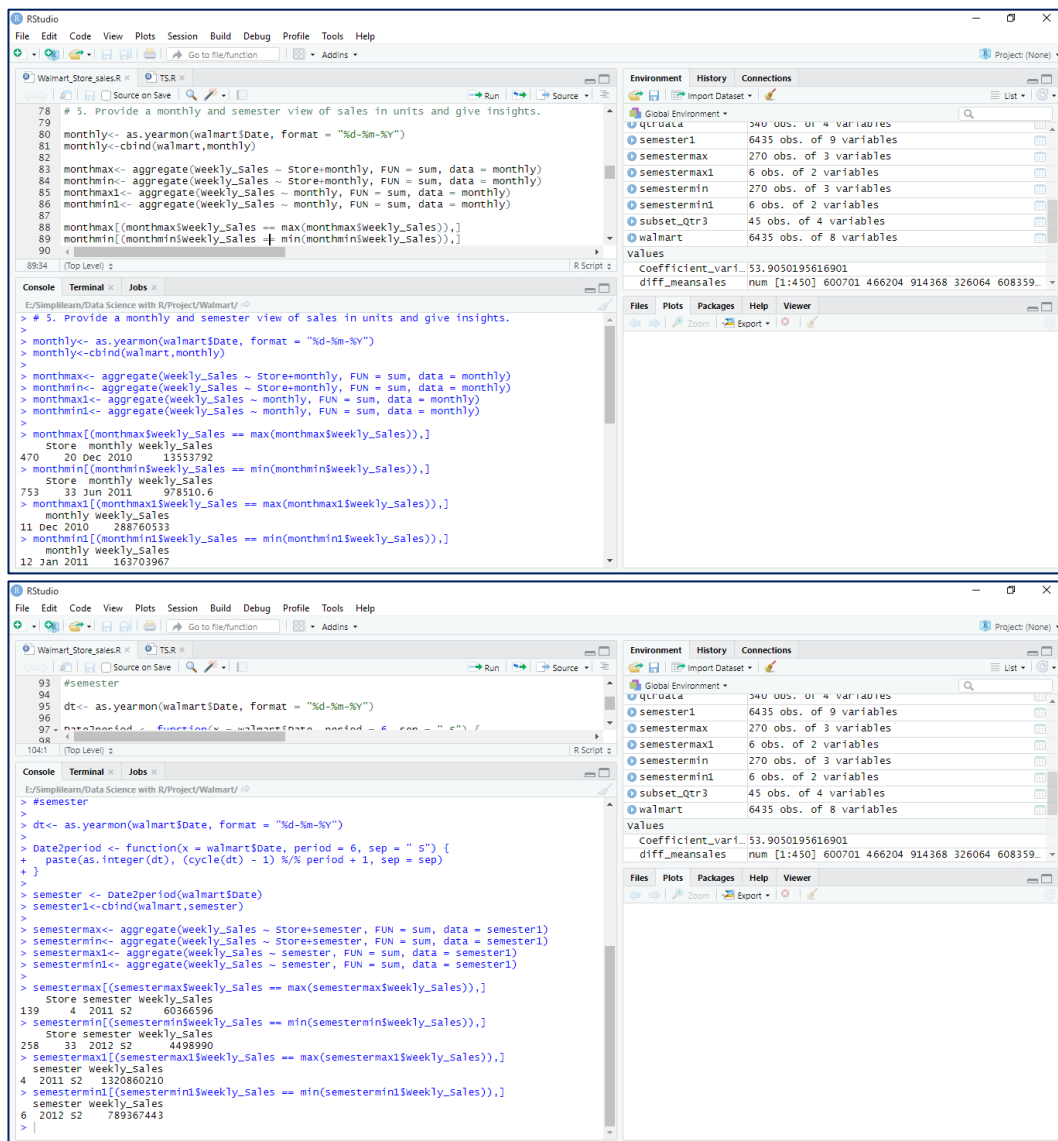
```

Interpretation:

Fairly easy to conclude here. The Dates or Holidays associated with it that have higher sales as compared to the mean sales on a Non-Holiday season are 25-11-2011 and 26-11-2010 (both Thanksgiving) with the difference being 19737068 and 18964466 respectively.

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

Convert the Date in the data into months and semesters using the libraries “zoo” and “lubridate”. Function as.yearmon converts the Date to Months and then we subsequently divide the data into 2 semesters. To understand the view of sales i.e., the minimum or maximum sales, we simply use the aggregate functions.



Interpretation:

Monthly view - The maximum sales of 288760533 is in the month of Dec 2010 whereas the minimum sales recorded at 163703967 is in the month of Jan 2011. Store 20 has the maximum sales in the month on Dec 2010 when compared to all store sales monthly data. Conversely, Store 33 recorded the least sales in the month of Jan 2011.

Semester view - The maximum sales of 1320860210 is in semester 2 of 2011 whereas the minimum sales recorded at 789367443 is in semester 2 of 2012. Store 4 has the maximum sales in S2 of 2011 when compared to all store sales semester data. Conversely, Store 33 recorded the least sales in S2 of 2012.

Statistical Model: -

For Store 1 – Build prediction models to forecast demand

Create a subset for Store 1 from the data provided.

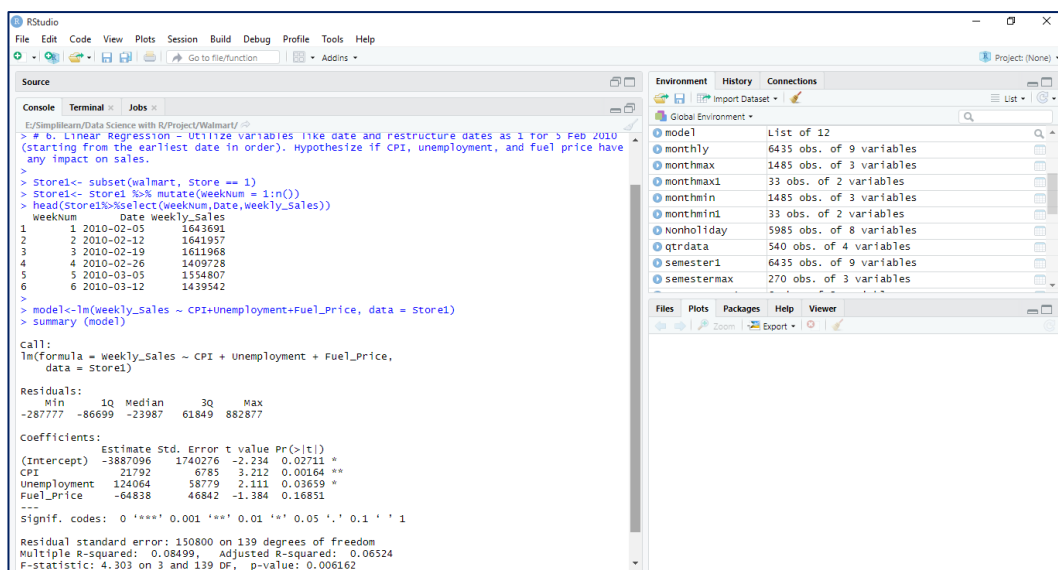
6. Linear Regression – Utilize variables like date and restructure dates as 1 for 5 Feb 2010(starting from the earliest date in order). Hypothesize if CPI, unemployment, and fuel price have any impact on sales.

To analyze if there is a relation between CPI, unemployment, fuel price and Weekly Sales, we use multiple linear regression model to analyze as we have more than one independent variable.

Defining Hypothesis:

Ho: CPI, unemployment, fuel price has no an impact on the Weekly Sales.

Ha: CPI, unemployment, fuel price has an impact on the Weekly Sales.



```
# 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())
> head(store1)%>%select(weekNum,date,weekly_sales))
  weekNum      date weekly_sales
1        1 2010-02-05      1643691
2        2 2010-02-12      1641957
3        3 2010-02-19      1611968
4        4 2010-02-26      1409728
5        5 2010-03-05      1554807
6        6 2010-03-12      1439542

> model<-lm(weekly_sales ~ CPI+unemployment+fuel_price, data = Store1)
> summary(model)

call:
lm(formula = weekly_sales ~ CPI + unemployment + fuel_price,
    data = Store1)

Residuals:
    Min       1Q   Median       3Q      Max
-287777    -86699    -23987     61849    882877

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -3887096   1740276  -2.234  0.02711 *
CPI           21792     6785    3.212  0.00164 **
unemployment 124064    58779   2.111  0.03659 *
fuel_price   -64838    46842  -1.384  0.16851
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 150800 on 139 degrees of freedom
Multiple R-squared:  0.08499,    Adjusted R-squared:  0.06524
F-statistic: 4.303 on 3 and 139 DF,  p-value: 0.006162
```

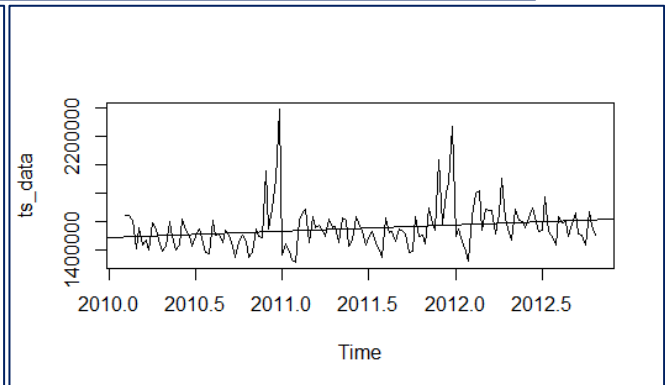
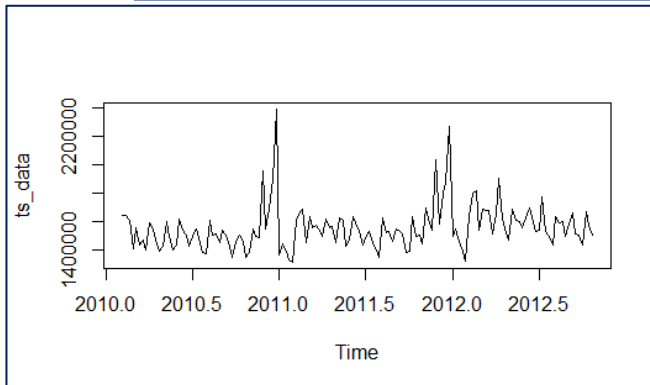
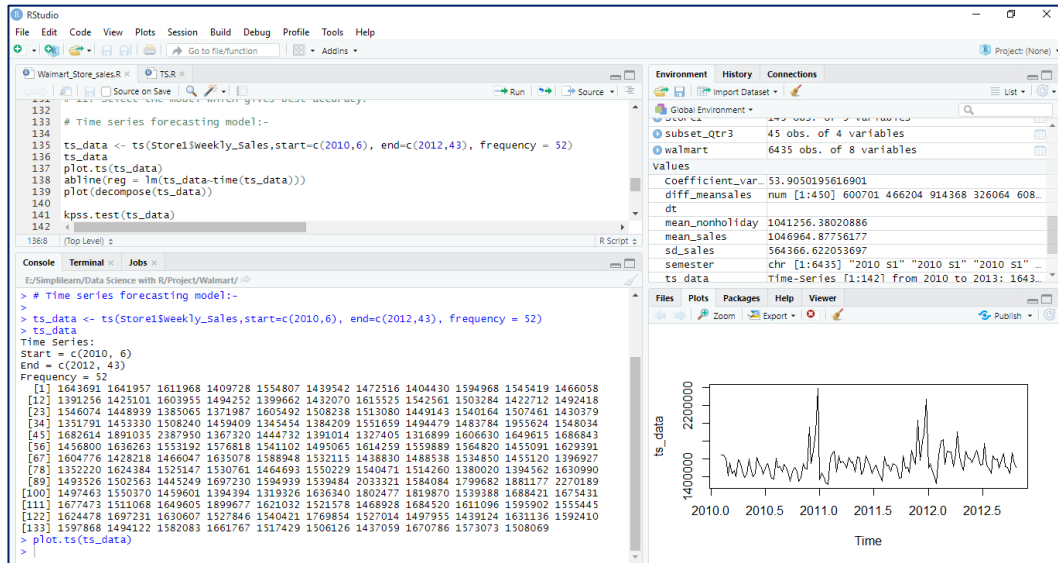
Interpretation:

As we know, $p\text{-value} < \alpha$, $p\text{-value}$ is less than α ; we reject the null hypothesis. We take α value as 0.05 at 95% confidence level.

We can clearly see that the significance codes with $p\text{-value} < 0.05$ for CPI and Unemployment signify that these variables are somewhat impacting the value of weekly sales and hence we cannot reject the null hypothesis.

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

To check if the data is fit for time series analysis, we use “ts” function from the tseries library.



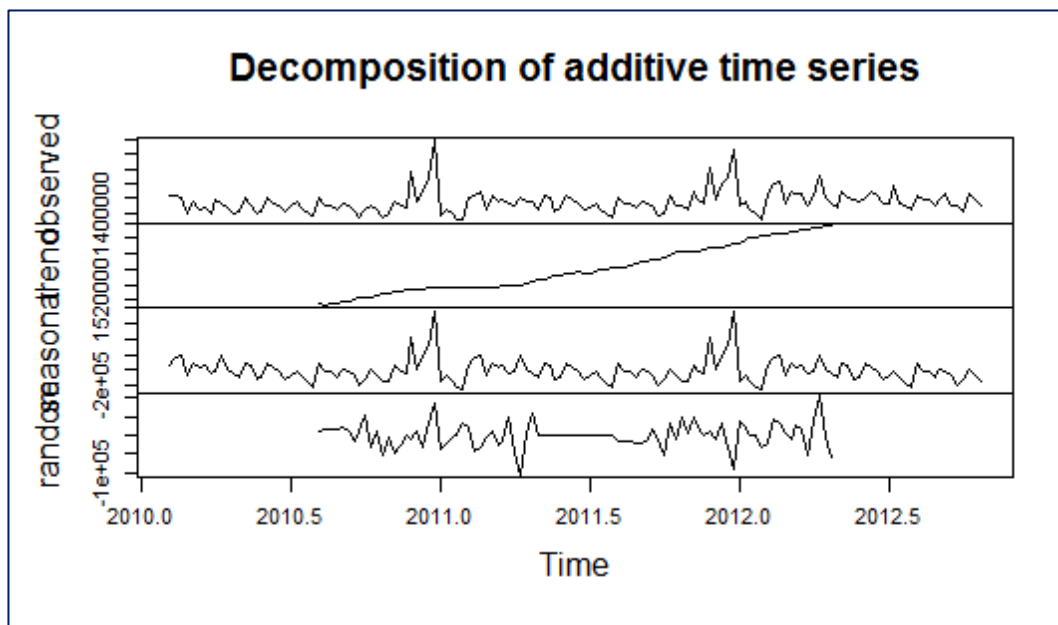
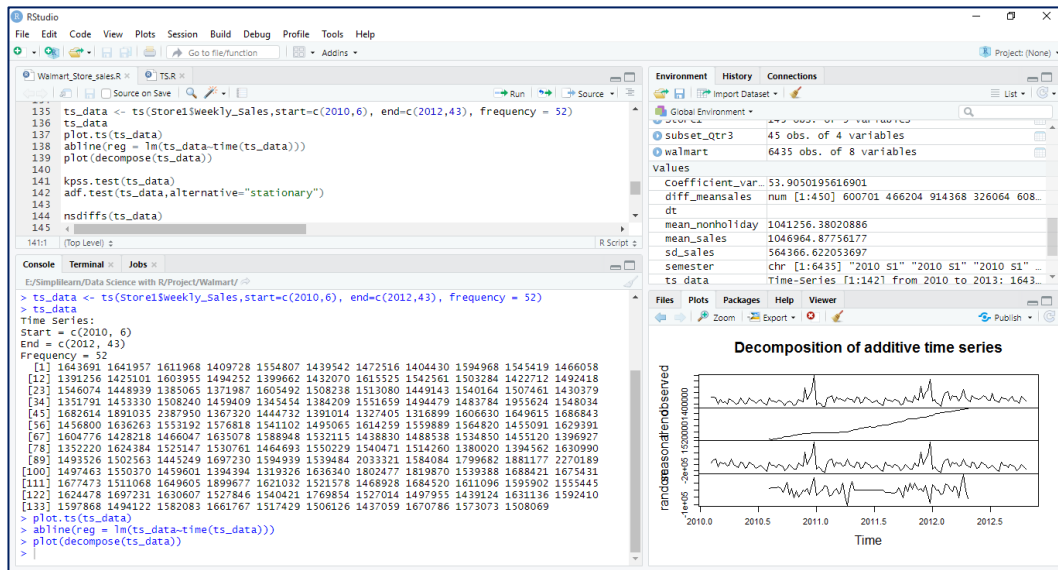
Interpretation:

Using the ts function and plotting the data, we can clearly see that the data shows both an upward trend and also some seasonal component. This data is currently not displaying white noise and not fit for time series analysis.

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

To make the data fit for time series analysis we need to first decompose the series to understand the components. We need to make the data stationary in order to do a

time series analysis. We, then use the Acf and Pacf functions to check for stationarity of the series.



Interpretation:

By the above graph we can safely say that the data is not stationary yet and we can extract information from it. As mentioned earlier, we can see both a trend and a seasonality component here.

```

139 plot(decompose(ts_data))
140
141 kpss.test(ts_data)
142 adf.test(ts_data, alternative="stationary")
143
144 nsdiffs(ts_data)
145 ndiffs(ts_data)
146
147 kpss.test(diff(log(ts_data)))
148 adf.test(diff(log(ts_data)), alternative="stationary")
149
149:16 (Top Level) >

```

```

> kpss.test(ts_data)

KPSS Test for Level Stationarity

data: ts_data
KPSS Level = 0.49275, Truncation lag parameter = 4, p-value = 0.0433
> adf.test(ts_data, alternative="stationary")

Augmented Dickey-Fuller Test

data: ts_data
Dickey-Fuller = -5.1745, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary

warning message:
In adf.test(ts_data, alternative = "stationary") :
  p-value smaller than printed p-value
>

```

Interpretation:

The KPSS test and ADF test is done to check for stationary data. In KPSS, we reject the null hypothesis, H_0 – Data is stationary if p-value is less than 0.05 and conversely in ADF test we reject, H_0 – Data is not stationary if p-value is less than 0.05.

Here we can see that the output of KPSS test is p-value = 0.043 which is less than the specified 0.05, hence we reject H_0 and conclude that the data is not stationary. Similarly, the output for the ADF test is p-value < 0.01 which is less than the specified 0.05, hence we cannot reject H_0 and conclude that the data is not stationary.

```

139 plot(decompose(ts_data))
140
141 kpss.test(ts_data)
142 adf.test(ts_data, alternative="stationary")
143
144 nsdiffs(ts_data)
145 ndiffs(ts_data)
146
147 kpss.test(diff(log(ts_data)))
148 adf.test(diff(log(ts_data)), alternative="stationary")
149
149:14 (Top Level) >

```

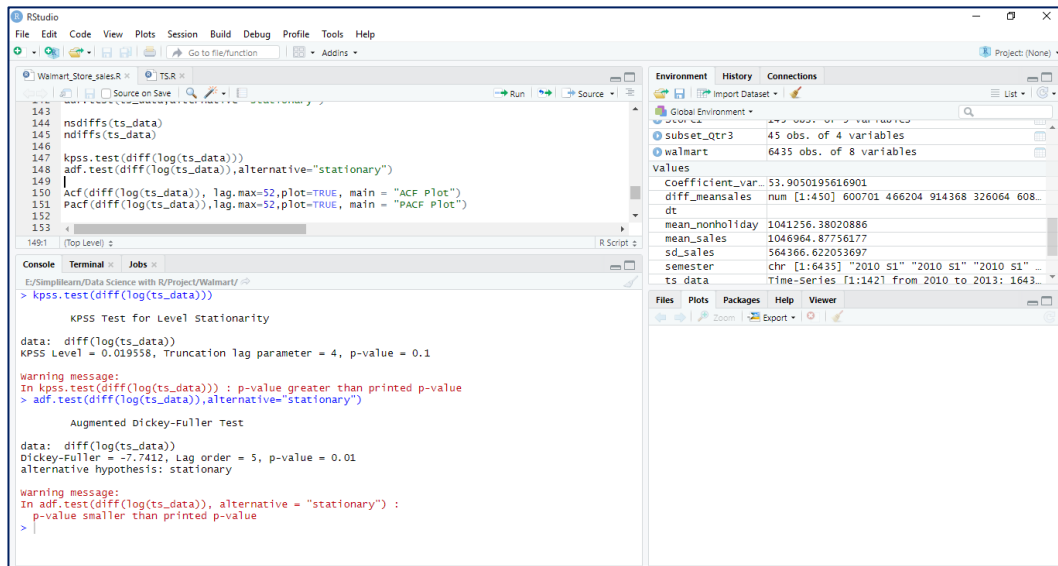
```

> nsdiffs(ts_data)
[1] 1
warning message:
The chosen seasonal unit root test encountered an error when testing for the second difference.
from stl(): series is not periodic or has less than two periods
1 seasonal differences will be used. Consider using a different unit root test.
> ndiffs(ts_data)
[1] 1
>

```

Interpretation:

The `ndiffs` and `nsdiffs` function gives us the value of d which is the number of times we need to difference the data to remove trend or seasonality. Here we the value of d is 1.



```
143
144 nsdiffs(ts_data)
145 ndiffs(ts_data)
146
147 kpss.test(diff(log(ts_data)))
148 adf.test(diff(log(ts_data)), alternative="stationary")
149
150 acf(diff(log(ts_data)), lag.max=52, plot=TRUE, main = "ACF Plot")
151 pacf(diff(log(ts_data)), lag.max=52, plot=TRUE, main = "PACF Plot")
152
153
```

Console Output:

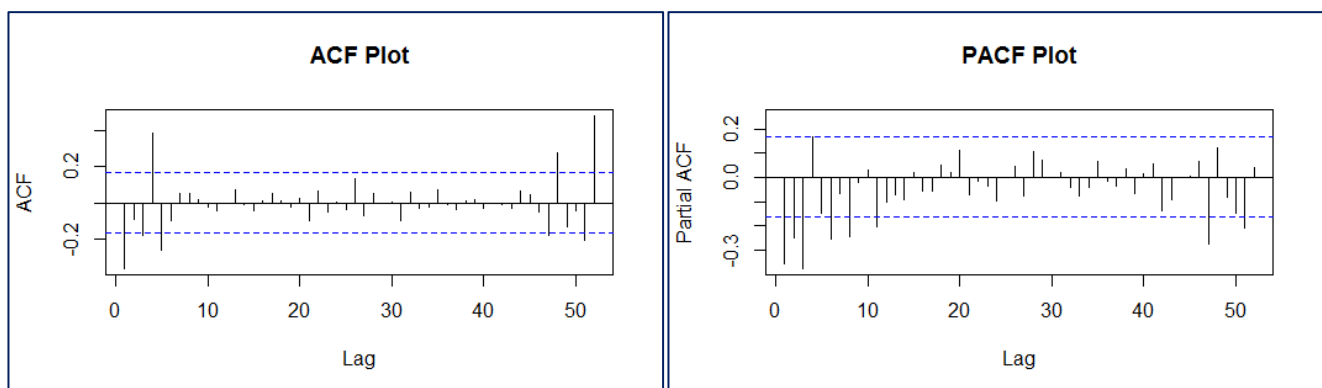
```
> kpss.test(diff(log(ts_data)))
KPSS Test for Level Stationarity
data: diff(log(ts_data))
KPSS Level = 0.019558, Truncation lag parameter = 4, p-value = 0.1
warning message:
In kpss.test(diff(log(ts_data))) : p-value greater than printed p-value
> adf.test(diff(log(ts_data)), alternative="stationary")
Augmented Dickey-Fuller Test
data: diff(log(ts_data))
Dickey-Fuller = -7.7412, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
warning message:
In adf.test(diff(log(ts_data)), alternative = "stationary") :
p-value smaller than printed p-value
```

Interpretation:

We use the log transform data to make data stationary on variance.

The KPSS test and ADF test is done to check for stationary data. In KPSS, we reject the null hypothesis, H_0 – Data is stationary if p-value is less than 0.05 and conversely in ADF test we reject, H_0 – Data is not stationary if p-value is less than 0.05.

Here we can see that the output of KPSS test is p-value > 0.1 which is greater than the specified 0.05, hence we fail to reject H_0 and conclude that the data is now stationary.



Interpretation:

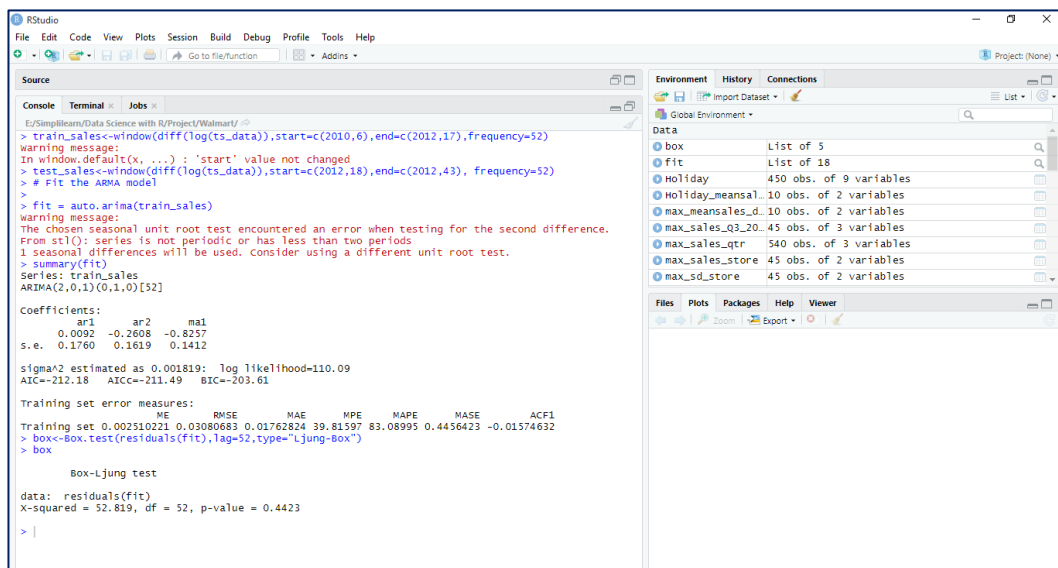
A good way to understand if the data has noise or not is the ACF and PACF functions. The dotted lines in the plot show the threshold for insignificant regions, i.e., for a significant

correlation, the horizontal line should be outside the dotted line. This is mostly done to identify the p and q values or the AR and MA components needed for ARIMA.

Since, there are enough spikes in the plots outside the insignificant zone (dotted horizontal lines) we can conclude that the residuals are not random. This implies that there is information available in residuals to be extracted by AR and MA models. Also, there is a seasonal component available in the residuals at the lag 52 (represented by spikes at lag 52). This makes sense since we are analyzing weekly data that does have seasonality due to Holidays.

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

We first divide the data into train (input till April 2012) and test data (predict next 6 months) and then use the auto.arima function to find the most fitted model for forecasting.



```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source
Console Terminal Jobs
E:\Similearn\Data Science with R\Project\Walmart\
> train_sales<-window(diff(log(ts_data)),start=c(2010,6),end=c(2012,17),frequency=52)
warning message:
In window.default(x, ...) : 'start' value not changed
> 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)
warning message:
The chosen seasonal unit root test encountered an error when testing for the second difference.
From stl(): series is not periodic or has less than two periods
1 seasonal differences will be used. consider using a different unit root test.
> summary(fit)
Series: train_sales
ARIMA(2,0,1)(0,1,0)[52]

Coefficients:
ar1      ar2      ma1
0.0092   -0.2608  -0.8257
s.e.    0.1760    0.1619    0.1412

sigma^2 estimated as 0.001819: log likelihood=110.09
AIC=-212.18   AICc=-211.49   BIC=-203.61

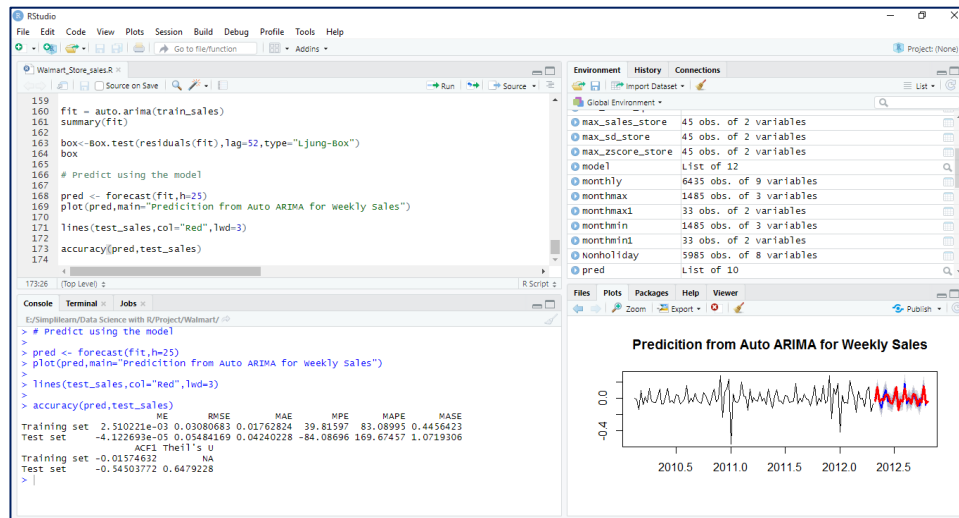
Training set error measures:
      ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
Training set 0.002510221 0.03080683 0.01762824 39.81597 83.08995 0.4456423 -0.01574632
> box<-Box.test(residuals(fit),lag=52,type='Ljung-Box')
> box
Box-Ljung test

data: residuals(fit)
X-squared = 52.819, df = 52, p-value = 0.4423
> |
```

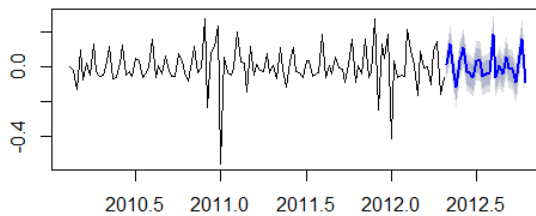
Interpretation:

A best model is one that has a minimum AIC and BIC values. Here the values for AIC and BIC are quite small. Also, the p-value is greater than 0.05 in the Box-Ljung test indicating that the model is independent of residuals.

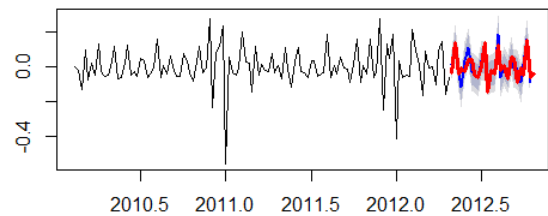
10. Predict next 6 months i.e., June to Oct 2010. Check for MAPE.



Prediction from Auto ARIMA for Weekly Sales



Prediction from Auto ARIMA for Weekly Sales



Interpretation:

The above graph indicates that the model built is the best fit model for the time series constructed.

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="Prediction from Auto ARIMA for Weekly Sales")

lines(test_sales,col="Red",lwd=3)

accuracy(pred,test_sales)
```

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