# HACKATHON UNILIVER - DEEP TECH ML (PROBLEM STATEMENT 3)

Saurav Suman,Anurag Kedia,Dinesh D(Group - ThePredictors)

### Read the training dataset (Training-Data-Sets.xlsx)

```
library(readxl)
sales_data <- read_excel("/Users/dinesh/Downloads/Training-Data-Sets.xlsx")

head(sales_data)

## # A tibble: 6 x 39
##     Day     EQ Social_Search_I… Social_Search_W… Digital_Impress…
##   <dbl>  <dbl>            <dbl>            <dbl>            <dbl>
## 1     1  718.          22256928            56812          7724107
## 2     2   25.5          4239408           105695          5844288
## 3     3  268.           6708500            87686         13008485
## 4     4  209.          36835247            70791          2520814
## 5     5 3482.          23693467            75610          9276779
## 6     6   55.2         13925382           114740          2733356
## # … with 34 more variables: Digital_Working_cost <dbl>,
## #   Print_Impressions.Ads40 <dbl>, Print_Working_Cost.Ads50 <dbl>,
## #   OOH_Impressions <dbl>, OOH_Working_Cost <dbl>, SOS_pct <dbl>,
## #   Digital_Impressions_pct <dbl>, CCFOT <dbl>, Median_Temp <dbl>,
## #   Median_Rainfall <dbl>, Fuel_Price <dbl>, Inflation <dbl>,
## #   Trade_Invest <dbl>, Brand_Equity <dbl>, Avg_EQ_Price <dbl>,
## #   Any_Promo_pct_ACV <dbl>, Any_Feat_pct_ACV <dbl>,
## #   Any_Disp_pct_ACV <dbl>, EQ_Base_Price <dbl>, Est_ACV_Selling <dbl>,
## #   pct_ACV <dbl>, Avg_no_of_Items <dbl>,
## #   pct_PromoMarketDollars_Category <dbl>, RPI_Category <dbl>,
## #   Magazine_Impressions_pct <dbl>, TV_GRP <dbl>, Competitor1_RPI <dbl>,
## #   Competitor2_RPI <dbl>, Competitor3_RPI <dbl>, Competitor4_RPI <dbl>,
## #   EQ_Category <dbl>, EQ_Subcategory <dbl>,
## #   pct_PromoMarketDollars_Subcategory <dbl>, RPI_Subcategory <dbl>
```

### Check the dimension of the dataset(rows and columns)

```
dim(sales_data)

## [1] 12000    39
```

### Check for the NA values in the dataset

```
sapply(sales_data,function(x){sum(is.na(x))})

##                                 Day                                  EQ
##                                   0                                   0
##          Social_Search_Impressions         Social_Search_Working_cost
##                                   0                                   0
```

```
##                     Digital_Impressions                      Digital_Working_cost
##                                       0                                         0
##                Print_Impressions.Ads40              Print_Working_Cost.Ads50
##                                       0                                         0
##                         OOH_Impressions                          OOH_Working_Cost
##                                       0                                         0
##                                 SOS_pct                    Digital_Impressions_pct
##                                       0                                         0
##                                   CCFOT                               Median_Temp
##                                       0                                         0
##                         Median_Rainfall                                Fuel_Price
##                                       0                                         0
##                               Inflation                              Trade_Invest
##                                       0                                         0
##                            Brand_Equity                              Avg_EQ_Price
##                                       0                                         0
##                       Any_Promo_pct_ACV                          Any_Feat_pct_ACV
##                                       0                                         0
##                        Any_Disp_pct_ACV                             EQ_Base_Price
##                                       0                                         0
##                         Est_ACV_Selling                                   pct_ACV
##                                       0                                         0
##                         Avg_no_of_Items      pct_PromoMarketDollars_Category
##                                       0                                         0
##                            RPI_Category                Magazine_Impressions_pct
##                                       0                                         0
##                                  TV_GRP                           Competitor1_RPI
##                                       0                                         0
##                         Competitor2_RPI                           Competitor3_RPI
##                                       0                                         0
##                         Competitor4_RPI                               EQ_Category
##                                       0                                         0
##                          EQ_Subcategory pct_PromoMarketDollars_Subcategory
##                                       0                                         0
##                         RPI_Subcategory
##                                       0
```

### Check the summary of the dataset

```r
summary(sales_data)
```

```
##       Day              EQ              Social_Search_Impressions
##  Min.   :    1   Min.   :    0.019   Min.   :  874111
##  1st Qu.: 3001   1st Qu.:   57.604   1st Qu.:10213887
##  Median : 6000   Median :  210.732   Median :19494577
##  Mean   : 6000   Mean   :  638.008   Mean   :19620560
##  3rd Qu.: 9000   3rd Qu.:  665.094   3rd Qu.:29138522
##  Max.   :12000   Max.   :18557.564   Max.   :38272395
##  Social_Search_Working_cost Digital_Impressions Digital_Working_cost
##  Min.   :  3546             Min.   :   23440    Min.   :  3493
##  1st Qu.: 33164             1st Qu.: 3330268    1st Qu.:112318
```

```
##  Median : 62888           Median : 6715113    Median :218230  
##  Mean   : 63132           Mean   : 6663405    Mean   :218973  
##  3rd Qu.: 92462           3rd Qu.: 9956033    3rd Qu.:326631  
##  Max.   :123421           Max.   :13238741    Max.   :432340  
##  Print_Impressions.Ads40 Print_Working_Cost.Ads50 OOH_Impressions    
##  Min.   : 46372          Min.   :   462           Min.   : 54350613  
##  1st Qu.:120125          1st Qu.: 47610           1st Qu.:251976016  
##  Median :193610          Median : 95586           Median :454057868  
##  Mean   :194404          Mean   : 95406           Mean   :452681237  
##  3rd Qu.:268844          3rd Qu.:143790           3rd Qu.:655791129  
##  Max.   :342242          Max.   :190389           Max.   :849360949  
##  OOH_Working_Cost     SOS_pct      Digital_Impressions_pct
##  Min.   : 237429   Min.   : 1.00   Min.   : 1.00          
##  1st Qu.:1095344   1st Qu.:13.00   1st Qu.:13.00          
##  Median :1959212   Median :25.00   Median :25.00          
##  Mean   :1975918   Mean   :25.42   Mean   :25.54          
##  3rd Qu.:2854418   3rd Qu.:38.00   3rd Qu.:38.00          
##  Max.   :3748194   Max.   :50.00   Max.   :50.00          
##      CCFOT         Median_Temp    Median_Rainfall     Fuel_Price   
##  Min.   : 11.00   Min.   :32.00   Min.   :0.01004   Min.   :7.234  
##  1st Qu.: 34.00   1st Qu.:43.00   1st Qu.:0.25907   1st Qu.:7.886  
##  Median : 56.00   Median :55.00   Median :0.50480   Median :8.518  
##  Mean   : 55.68   Mean   :54.97   Mean   :0.50505   Mean   :8.535  
##  3rd Qu.: 78.00   3rd Qu.:67.00   3rd Qu.:0.74992   3rd Qu.:9.199  
##  Max.   :100.00   Max.   :78.00   Max.   :0.99998   Max.   :9.837  
##    Inflation         Trade_Invest    Brand_Equity    Avg_EQ_Price  
##  Min.   :0.009931   Min.   :  508   Min.   :42.41   Min.   :42.22  
##  1st Qu.:0.039376   1st Qu.: 2955   1st Qu.:42.79   1st Qu.:46.76  
##  Median :0.069566   Median : 5424   Median :43.19   Median :51.24  
##  Mean   :0.069431   Mean   : 5401   Mean   :43.20   Mean   :51.19  
##  3rd Qu.:0.099335   3rd Qu.: 7841   3rd Qu.:43.60   3rd Qu.:55.64  
##  Max.   :0.129155   Max.   :10291   Max.   :43.99   Max.   :60.00  
##  Any_Promo_pct_ACV Any_Feat_pct_ACV Any_Disp_pct_ACV EQ_Base_Price  
##  Min.   : 0.334    Min.   :1.87     Min.   :0.178    Min.   :1.412  
##  1st Qu.: 4.440    1st Qu.:2.84     1st Qu.:1.213    1st Qu.:1.479  
##  Median : 8.460    Median :3.87     Median :2.260    Median :1.547  
##  Mean   : 8.491    Mean   :3.86     Mean   :2.258    Mean   :1.548  
##  3rd Qu.:12.534    3rd Qu.:4.86     3rd Qu.:3.293    3rd Qu.:1.616  
##  Max.   :16.738    Max.   :5.87     Max.   :4.324    Max.   :1.682  
##  Est_ACV_Selling        pct_ACV      Avg_no_of_Items
##  Min.   :238518835   Min.   :13.89   Min.   :2.222  
##  1st Qu.:404127338   1st Qu.:21.79   1st Qu.:2.393  
##  Median :566115033   Median :29.78   Median :2.561  
##  Mean   :566818008   Mean   :29.76   Mean   :2.561  
##  3rd Qu.:731055504   3rd Qu.:37.76   3rd Qu.:2.731  
##  Max.   :893820548   Max.   :45.67   Max.   :2.898  
##  pct_PromoMarketDollars_Category  RPI_Category   Magazine_Impressions_pct
##  Min.   :0.01233                 Min.   :35.62   Min.   :21.89           
##  1st Qu.:0.12157                 1st Qu.:38.15   1st Qu.:36.10           
##  Median :0.23260                 Median :40.59   Median :50.19           
```

```
##  Mean   :0.23249                  Mean   :40.63   Mean   :50.35
##  3rd Qu.:0.34173                  3rd Qu.:43.18   3rd Qu.:64.57
##  Max.   :0.45360                  Max.   :45.63   Max.   :78.73
##      TV_GRP      Competitor1_RPI  Competitor2_RPI Competitor3_RPI
##  Min.   :12.34   Min.   : 91.90   Min.   :31.90   Min.   :42.90
##  1st Qu.:20.72   1st Qu.: 99.03   1st Qu.:35.26   1st Qu.:44.52
##  Median :28.96   Median :106.15   Median :38.62   Median :46.17
##  Mean   :28.90   Mean   :106.17   Mean   :38.66   Mean   :46.16
##  3rd Qu.:37.24   3rd Qu.:113.28   3rd Qu.:42.03   3rd Qu.:47.79
##  Max.   :45.34   Max.   :120.44   Max.   :45.44   Max.   :49.44
##  Competitor4_RPI  EQ_Category       EQ_Subcategory
##  Min.   :61.90   Min.   : 1234920   Min.   :209473
##  1st Qu.:66.29   1st Qu.: 3965402   1st Qu.:380095
##  Median :70.67   Median : 6619380   Median :547465
##  Mean   :70.64   Mean   : 6688170   Mean   :549686
##  3rd Qu.:75.03   3rd Qu.: 9430345   3rd Qu.:719805
##  Max.   :79.44   Max.   :12234003   Max.   :893820
##  pct_PromoMarketDollars_Subcategory RPI_Subcategory
##  Min.   :0.0472                     Min.   :31.23
##  1st Qu.:0.1332                     1st Qu.:35.73
##  Median :0.2200                     Median :40.20
##  Mean   :0.2189                     Mean   :40.16
##  3rd Qu.:0.3054                     3rd Qu.:44.56
##  Max.   :0.3893                     Max.   :49.02
```

### Check the datatype of the dataset

```
str(sales_data)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    12000 obs. of  39 variables:
##  $ Day                                : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ EQ                                 : num  718.5 25.5 268.3 209.1 3482.2 ...
##  $ Social_Search_Impressions          : num  22256928 4239408 6708500 36835247
23693467 ...
##  $ Social_Search_Working_cost         : num  56812 105695 87686 70791 75610 ...
##  $ Digital_Impressions                : num  7724107 5844288 13008485 2520814
9276779 ...
##  $ Digital_Working_cost               : num  238700 188902 19704 200111 65532 ...
##  $ Print_Impressions.Ads40            : num  151438 264008 150505 253458 278877 ...
##  $ Print_Working_Cost.Ads50           : num  1044 113582 38501 53719 95178 ...
##  $ OOH_Impressions                    : num  1.12e+08 2.85e+08 8.08e+08 6.67e+08
7.40e+07 ...
##  $ OOH_Working_Cost                   : num  2133614 1719318 1569740 922723 1834970
...
##  $ SOS_pct                            : num  5 38 9 9 26 6 21 11 46 44 ...
##  $ Digital_Impressions_pct            : num  11 14 33 43 22 8 20 23 16 21 ...
##  $ CCFOT                              : num  62 59 51 56 48 97 70 81 57 74 ...
##  $ Median_Temp                        : num  55 61 33 51 54 44 32 71 42 67 ...
##  $ Median_Rainfall                    : num  0.493 0.0781 0.9486 0.7092 0.9655 ...
##  $ Fuel_Price                         : num  8.07 9.33 9.55 7.84 8.09 ...
##  $ Inflation                          : num  0.0676 0.0462 0.027 0.1066 0.1291 ...
##  $ Trade_Invest                       : num  7708 6693 2699 4898 8678 ...
##  $ Brand_Equity                       : num  42.8 42.8 43 43.5 43.8 ...
```

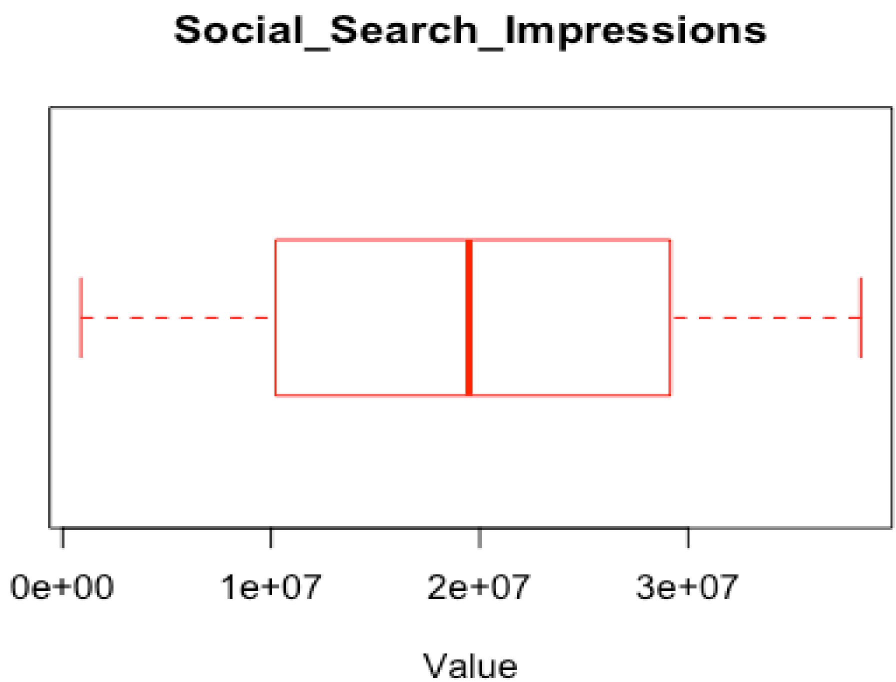
```
##  $ Avg_EQ_Price                       : num  43.1 44.3 48.3 59.1 48.5 ...
##  $ Any_Promo_pct_ACV                  : num  13.663 9.632 14.728 0.465 9.217 ...
##  $ Any_Feat_pct_ACV                   : num  5.25 1.87 5.64 2.86 4.28 2.01 4.42
4.05 5.6 3.66 ...
##  $ Any_Disp_pct_ACV                   : num  1.58 0.806 3.026 1.006 3.681 ...
##  $ EQ_Base_Price                      : num  1.68 1.65 1.62 1.62 1.43 ...
##  $ Est_ACV_Selling                    : num  4.46e+08 8.56e+08 5.04e+08 4.63e+08
6.92e+08 ...
##  $ pct_ACV                            : num  20.6 26.5 14.9 28.9 36.1 ...
##  $ Avg_no_of_Items                    : num  2.81 2.36 2.84 2.79 2.81 ...
##  $ pct_PromoMarketDollars_Category    : num  0.1996 0.2939 0.3148 0.0767 0.3639 ...
##  $ RPI_Category                       : num  36.2 43 42 41.4 38.2 ...
##  $ Magazine_Impressions_pct           : num  54.2 65.8 45.1 75.2 56.9 ...
##  $ TV_GRP                             : num  16.5 15.6 23.9 13.1 40.7 ...
##  $ Competitor1_RPI                    : num  106 112 110 117 115 ...
##  $ Competitor2_RPI                    : num  36.1 43.3 38.3 39.4 36.9 ...
##  $ Competitor3_RPI                    : num  46.4 47.6 49.3 44.2 45.5 ...
##  $ Competitor4_RPI                    : num  71.8 67.9 72.7 73 75.8 ...
##  $ EQ_Category                        : num  5420048 12155631 11939870 7045541
11488805 ...
##  $ EQ_Subcategory                     : num  475559 371540 225984 551342 254143 ...
##  $ pct_PromoMarketDollars_Subcategory: num  0.3766 0.2515 0.3679 0.0504 0.2219 ...
##  $ RPI_Subcategory                    : num  45.8 35 46.2 38.1 39.2 ...
```

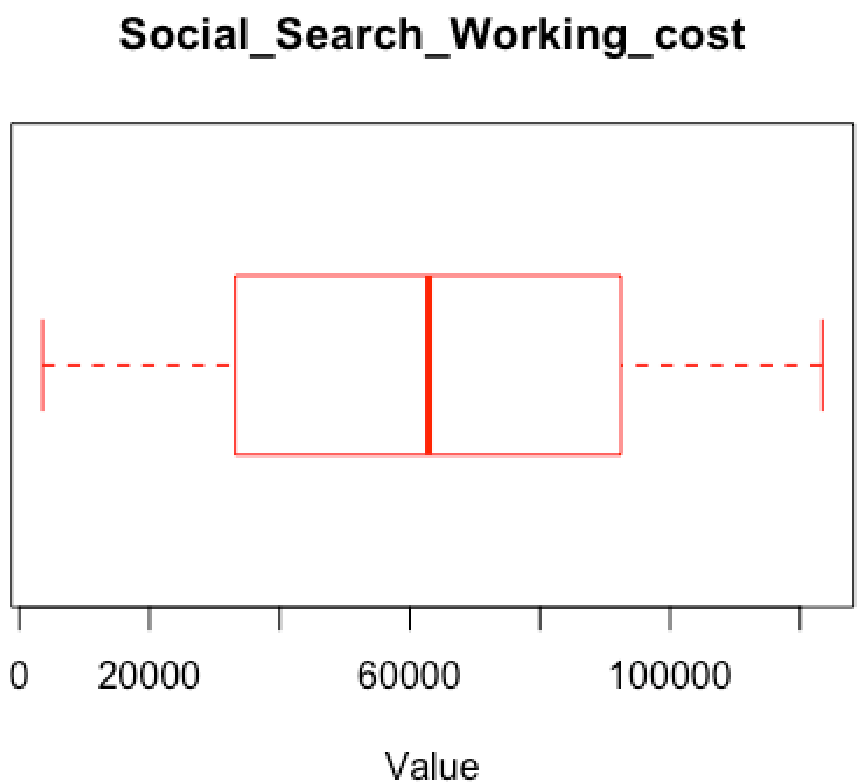
### Plot the boxplot to check the outliers in the target variable - EQ

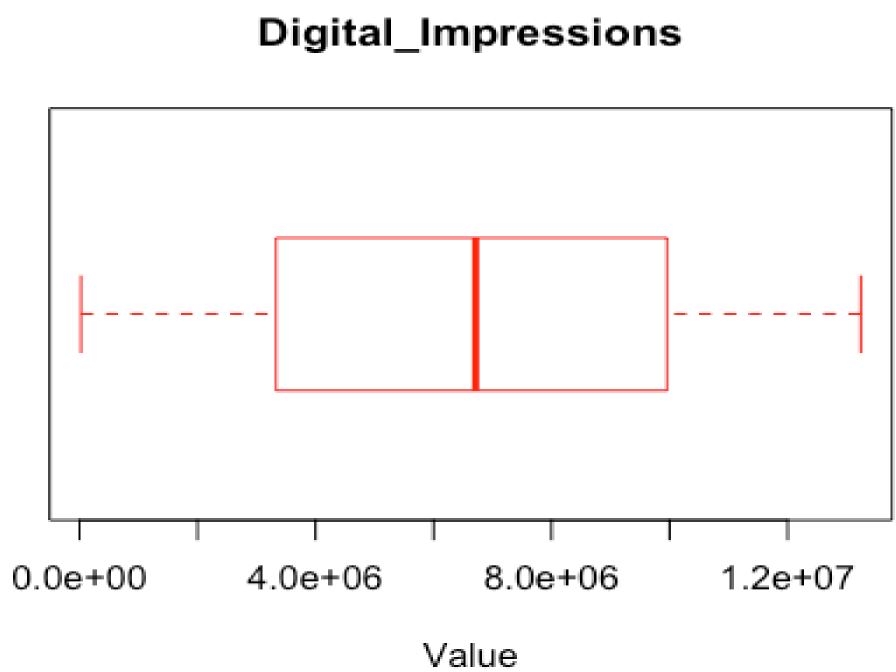
```
library(ggplot2)
#ggplot(data=sales_data, aes(EQ, fill=EQ)) + geom_boxplot(colour="Black")
```

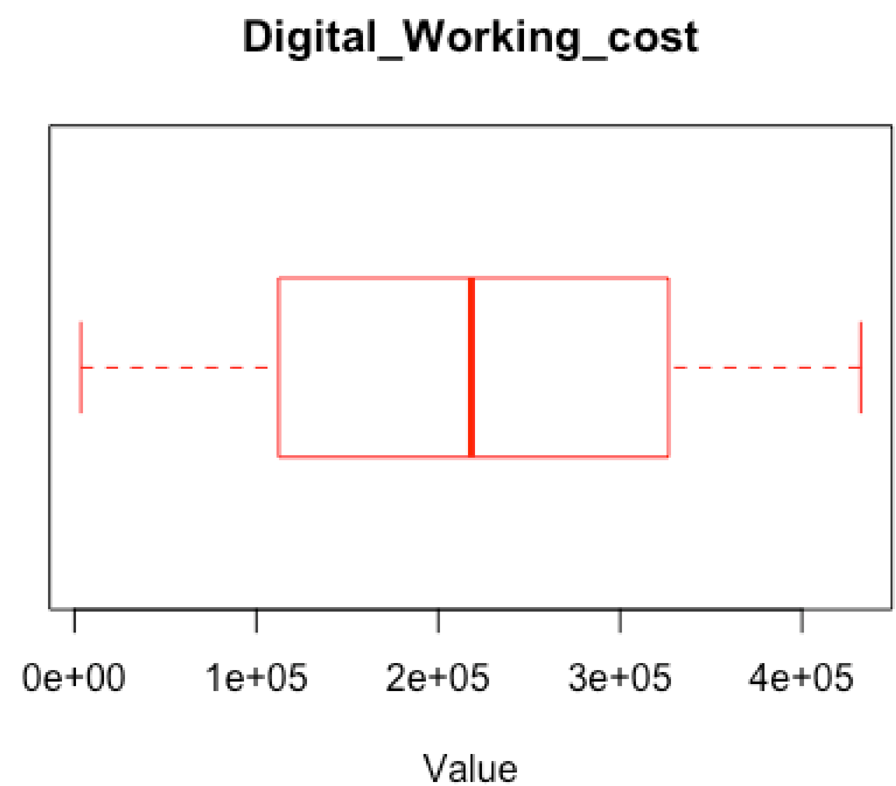
## Box plot of each independent variable ( Univariate Analysis )

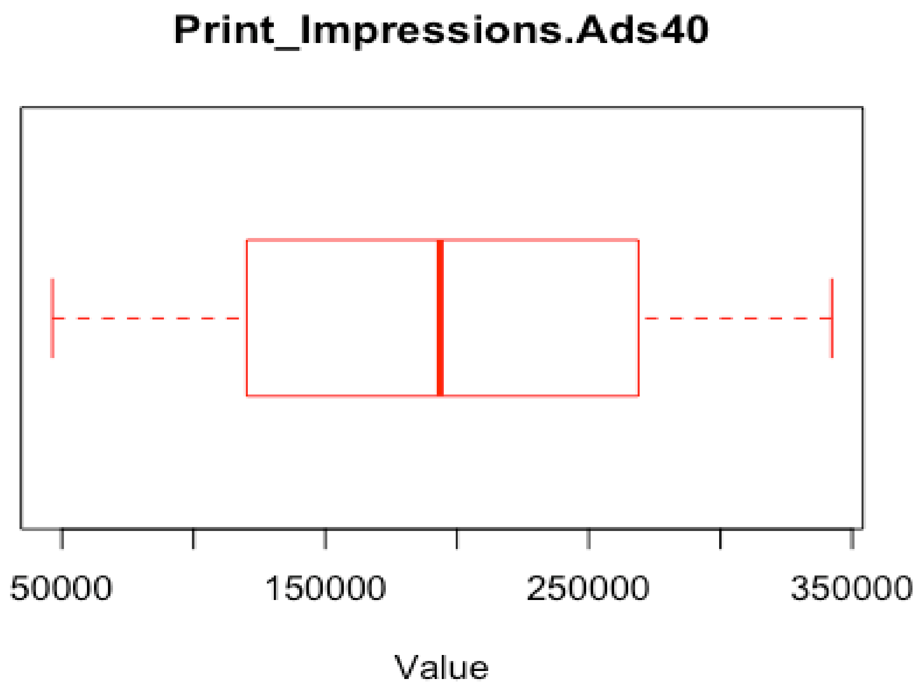
```
for (i in 3:ncol(sales_data))
{
 boxplot(sales_data[,i],horizontal = TRUE, border = 'red',
         xlab = "Value",main = colnames(sales_data[i]))}
```

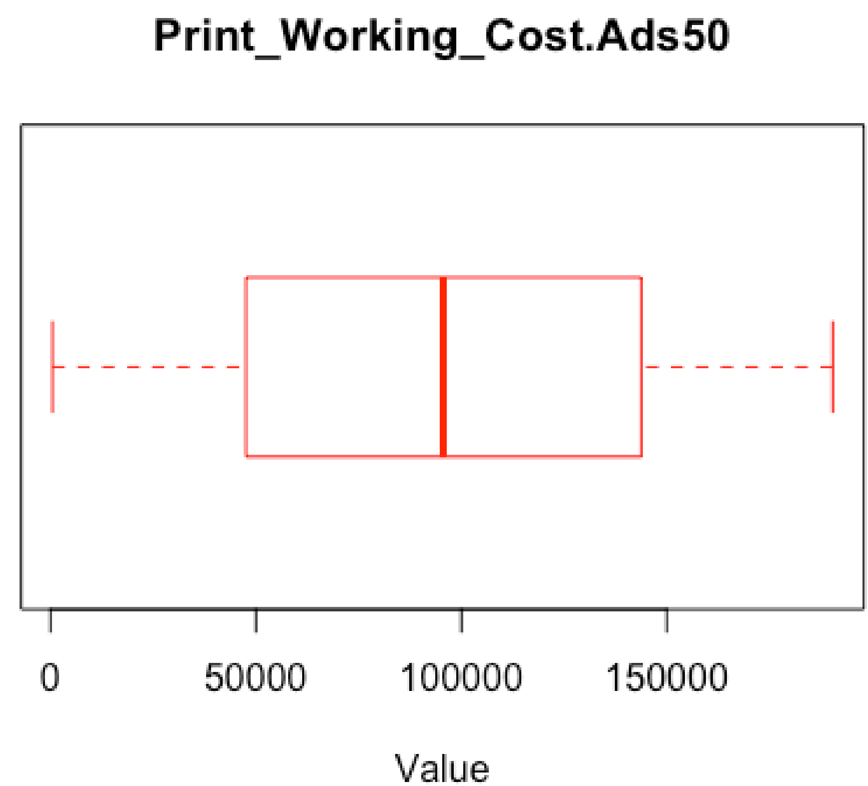

Social_Search_Impressions
0e+00
1e+07
2e+07
3e+07
Value

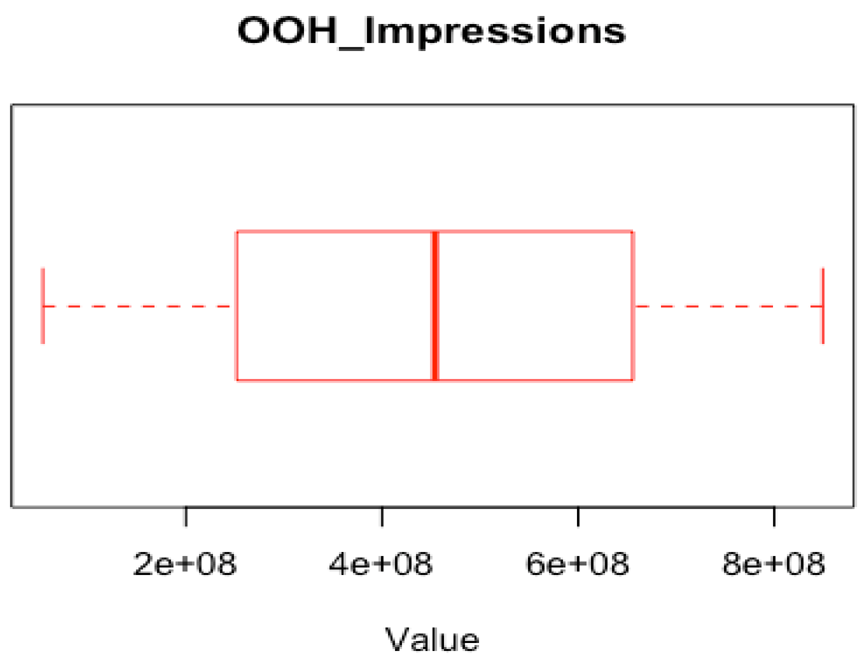

Social_Search_Working_cost
0
20000
60000
100000
Value

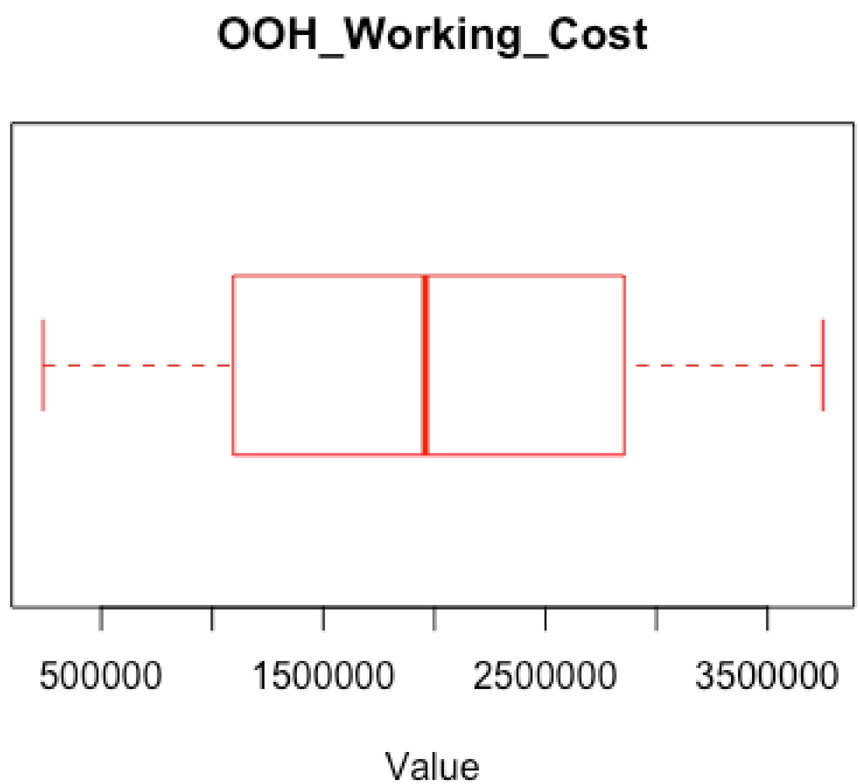

Digital_Impressions
0.0e+00
4.0e+06
8.0e+06
1.2e+07
Value


Digital_Working_cost
0e+00
1e+05
2e+05
3e+05
4e+05
Value

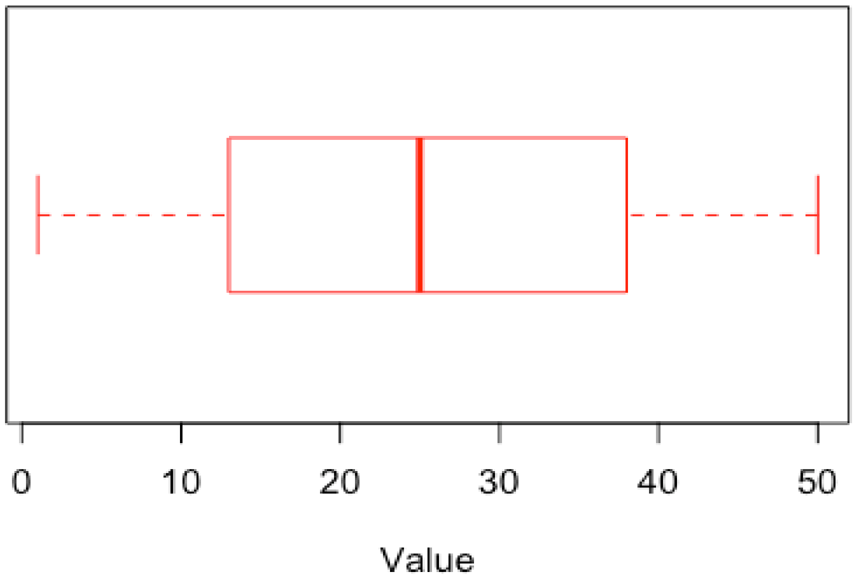

Print_Impressions.Ads40
50000
150000
250000
350000
Value


Print_Working_Cost.Ads50
0
50000
100000
150000
Value

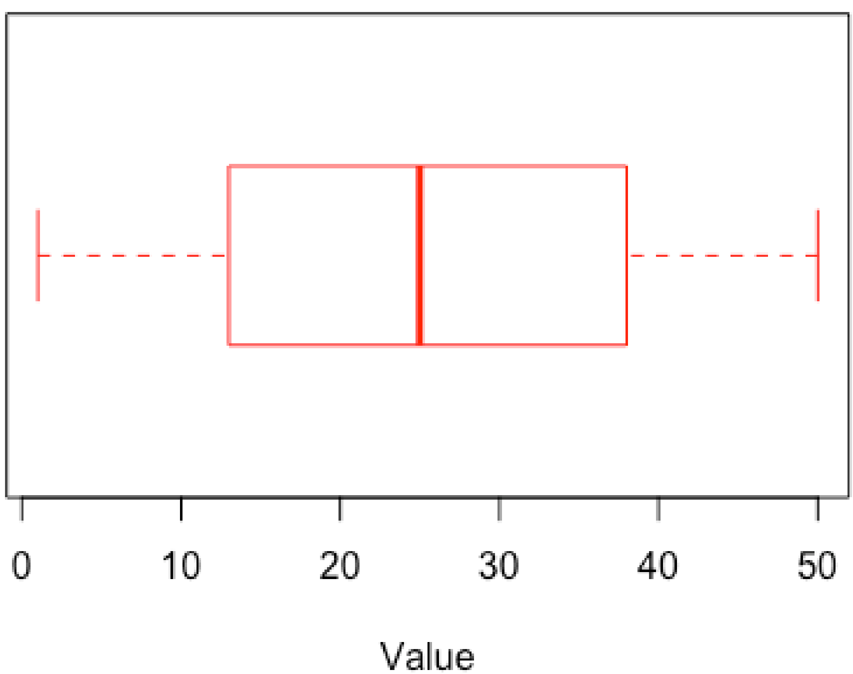

OOH_Impressions
2e+08
4e+08
6e+08
8e+08
Value

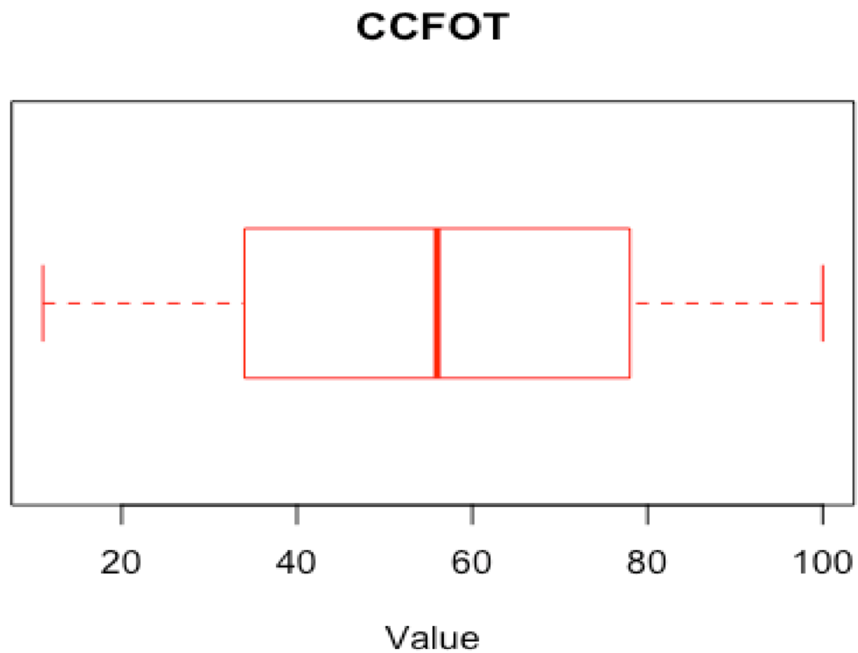

OOH_Working_Cost
500000
1500000
2500000
3500000
Value

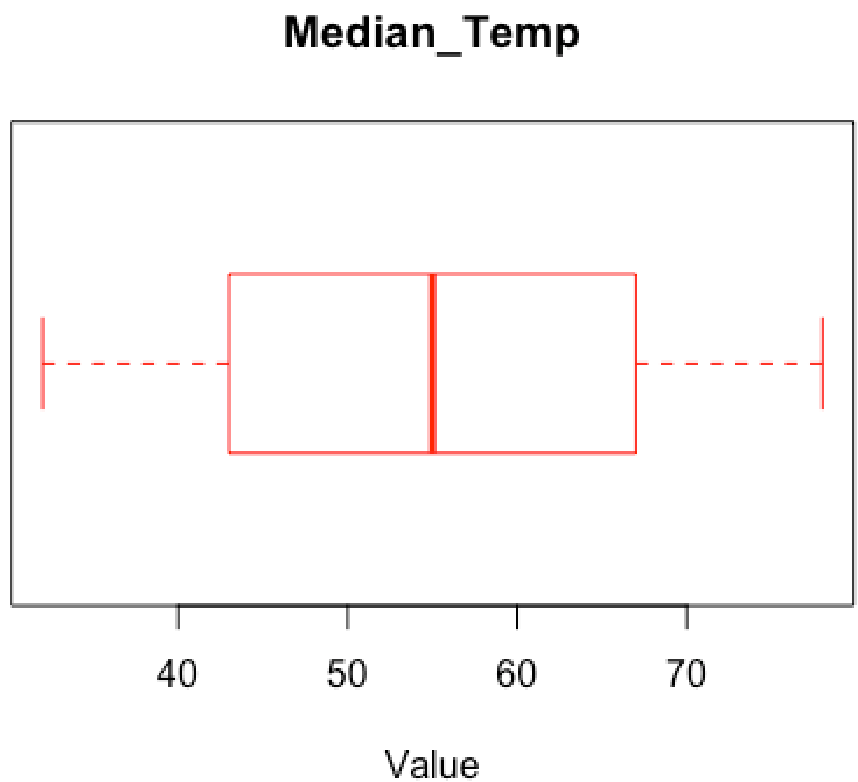
**SOS_pct**

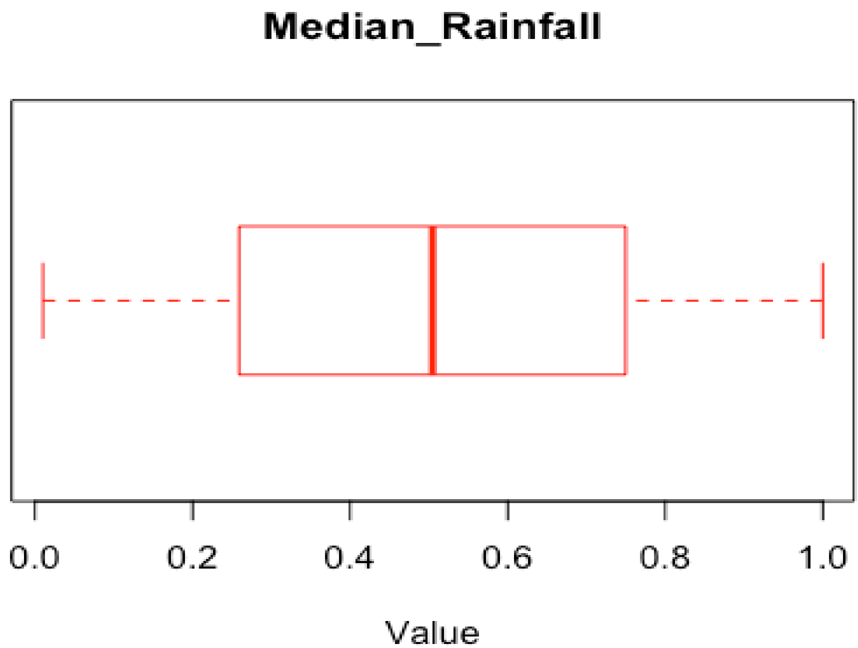


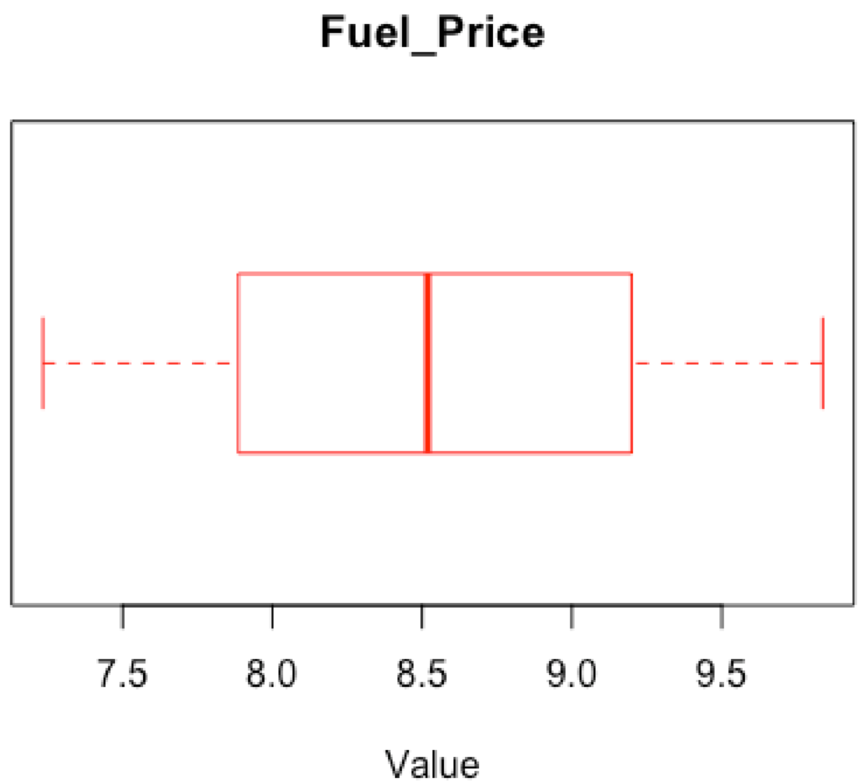

**Digital_Impressions_pct**

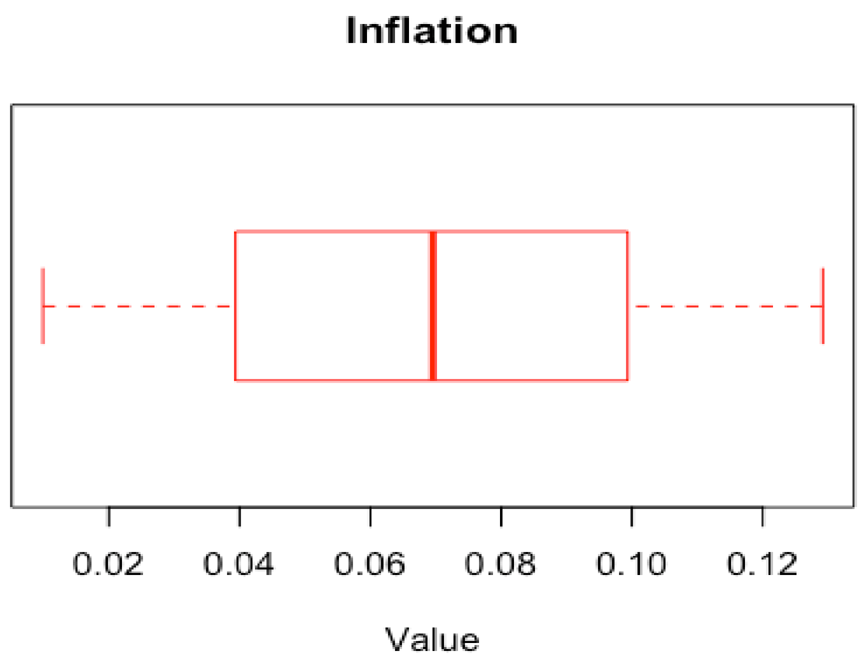


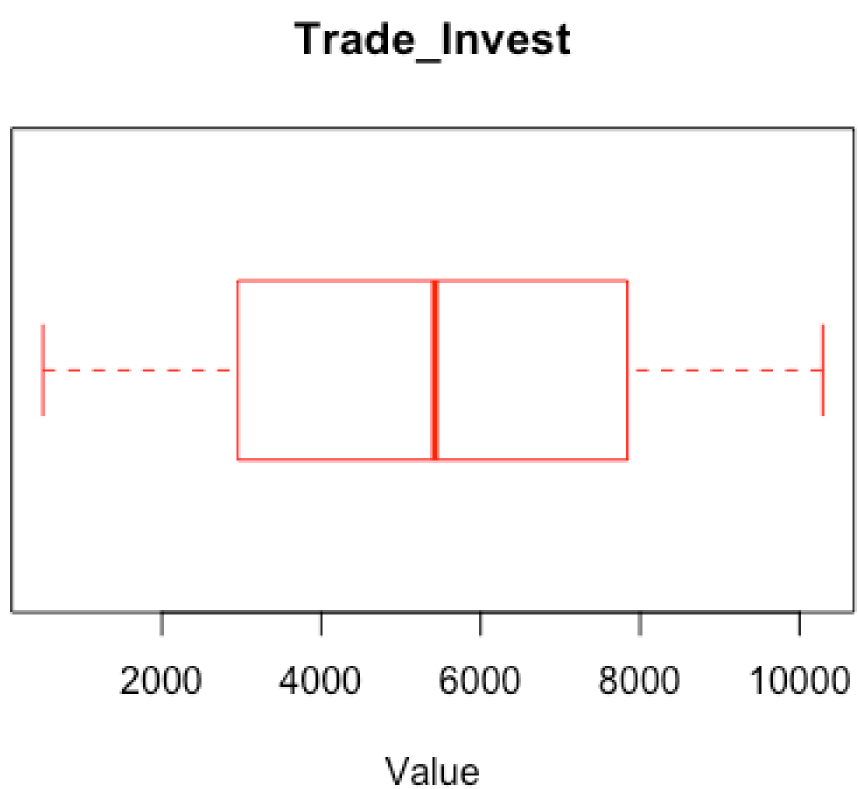

CCFOT
20
40
60
80
100
Value

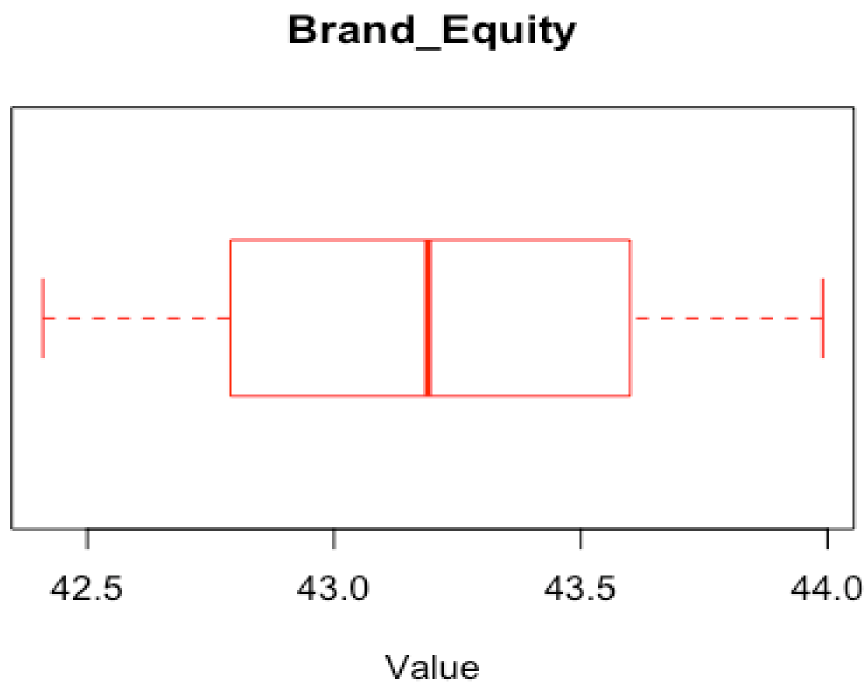

Median_Temp
40
50
60
70
Value

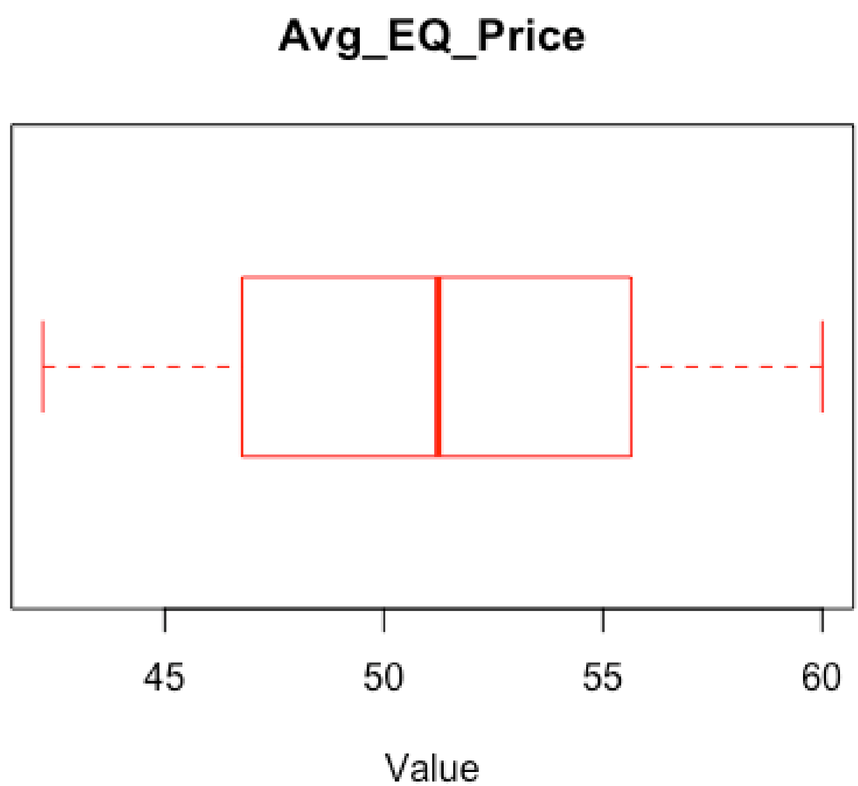

Median_Rainfall
0.0
0.2
0.4
0.6
0.8
1.0
Value


Fuel_Price
7.5
8.0
8.5
9.0
9.5
Value

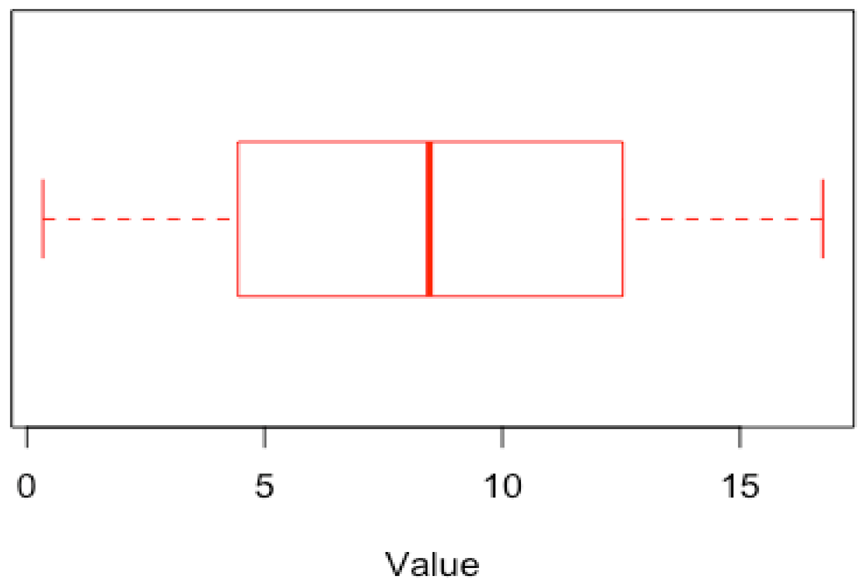

Inflation
0.02
0.04
0.06
0.08
0.10
0.12
Value


Trade_Invest
2000
4000
6000
8000
10000
Value

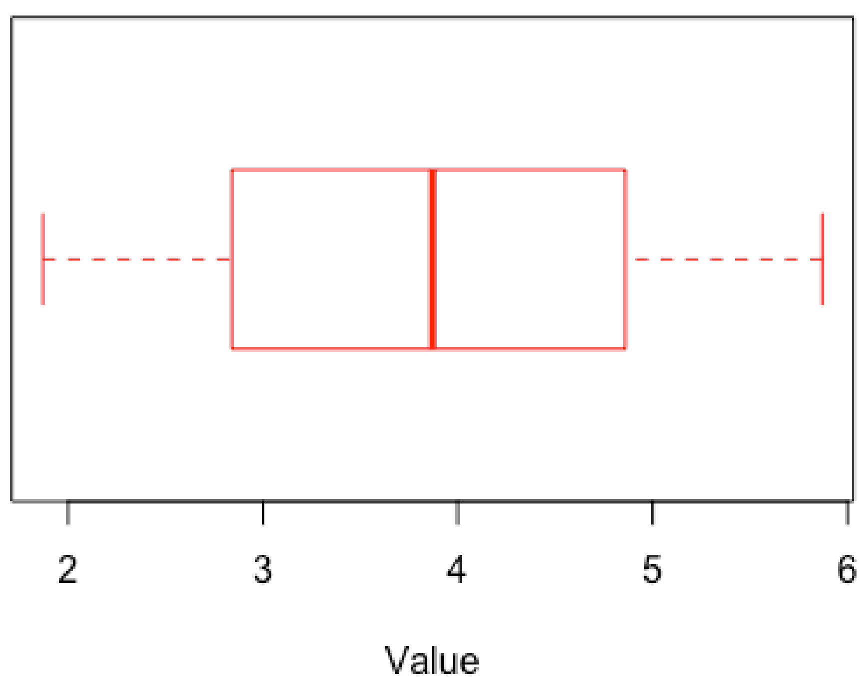

Brand_Equity
42.5
43.0
43.5
44.0
Value

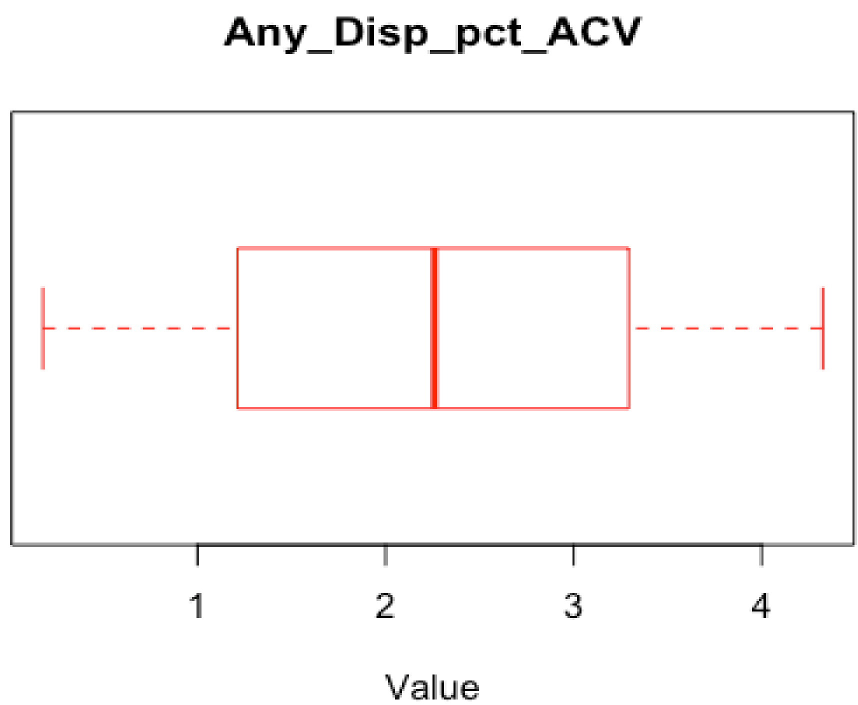

Avg_EQ_Price
45
50
55
60
Value

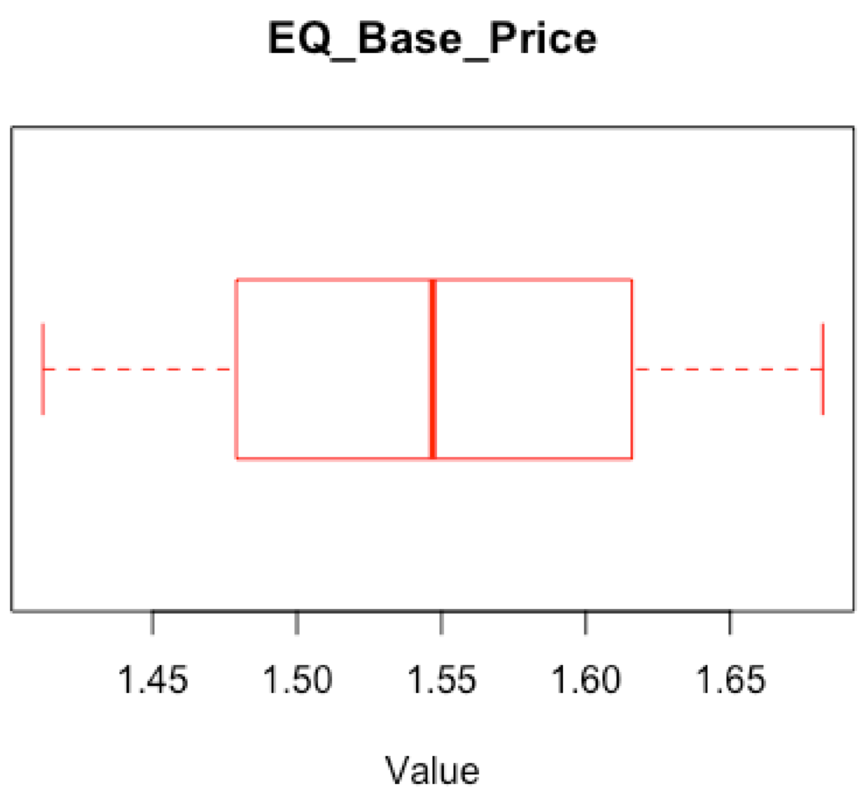
## Any_Promo_pct_ACV

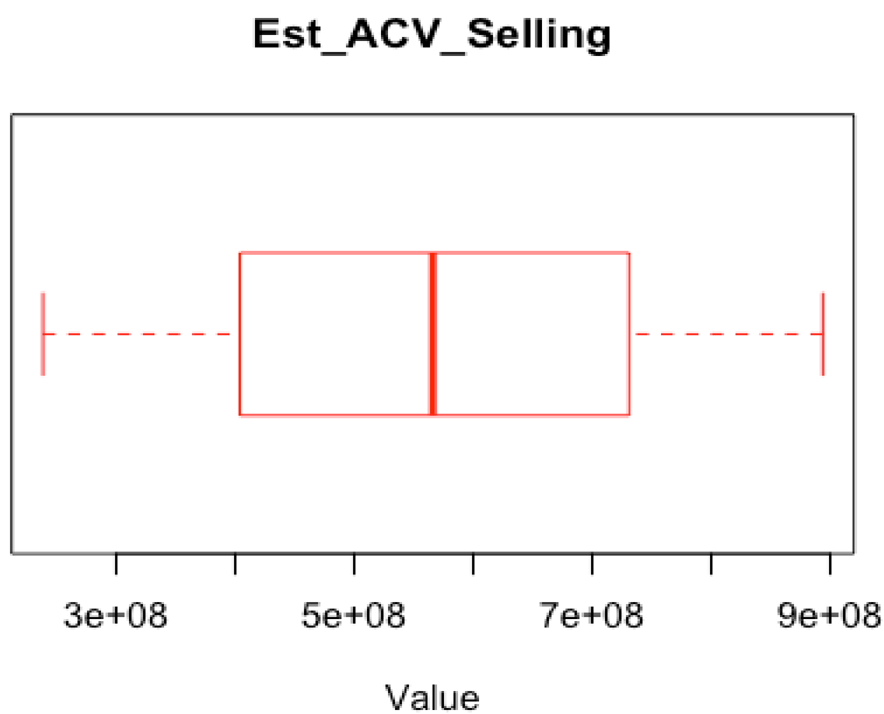


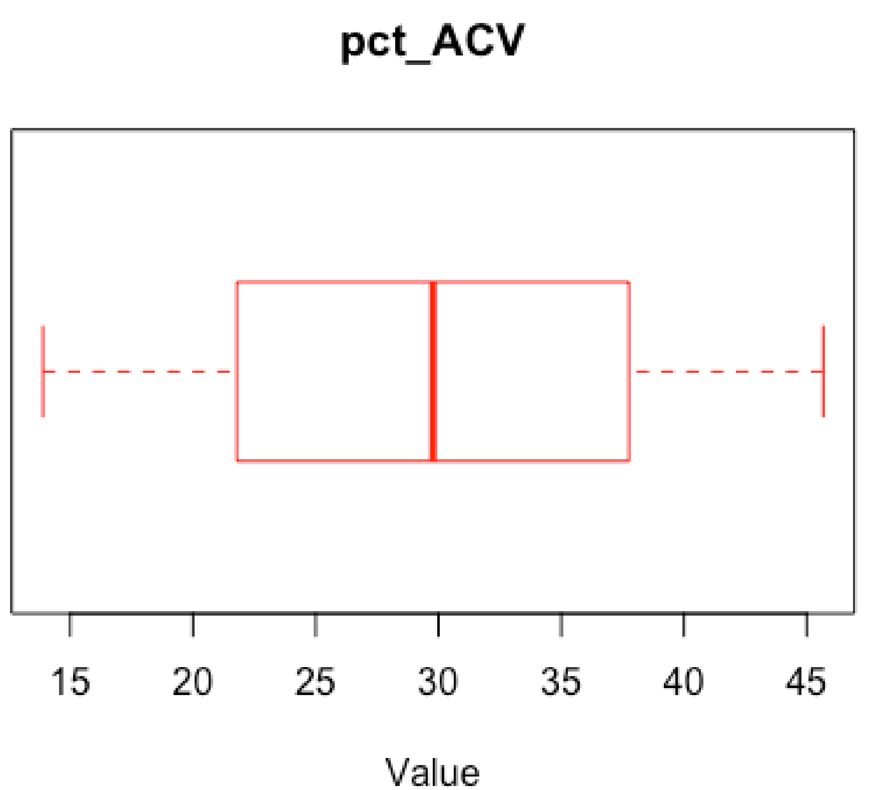

## Any_Feat_pct_ACV

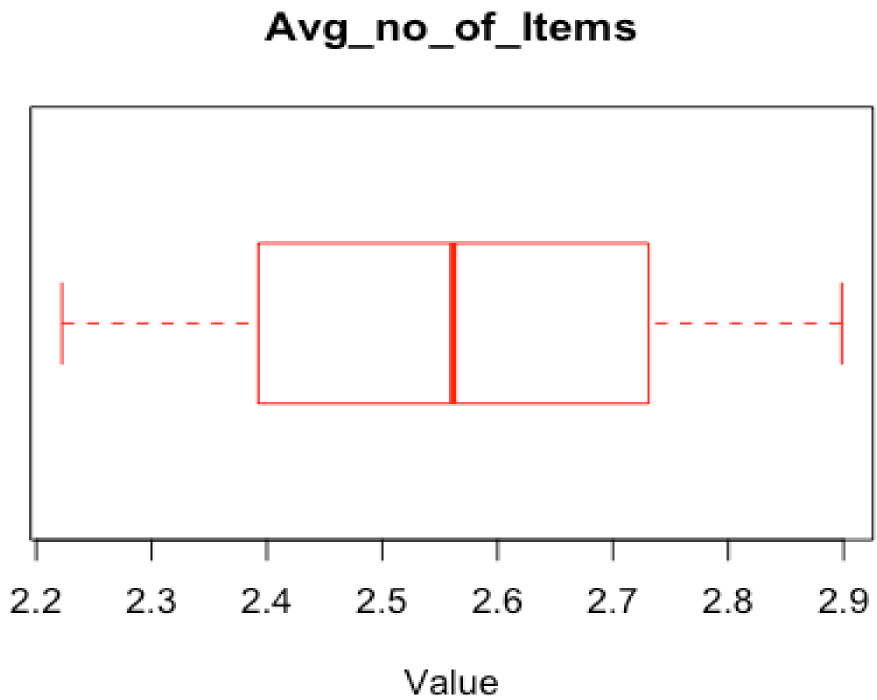


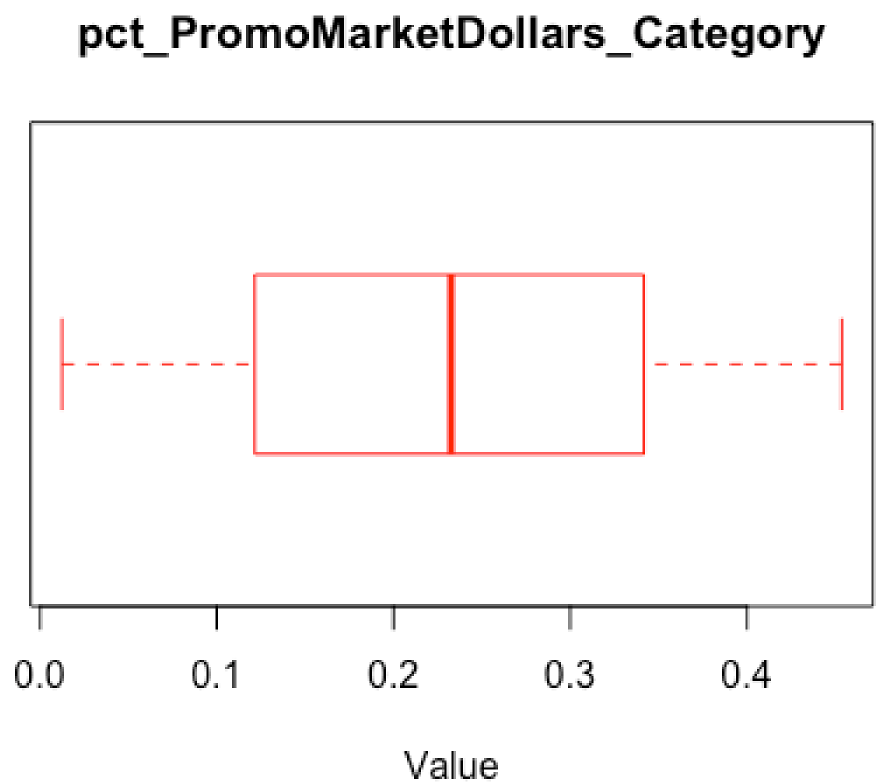

Any_Disp_pct_ACV
1
2
3
4
Value

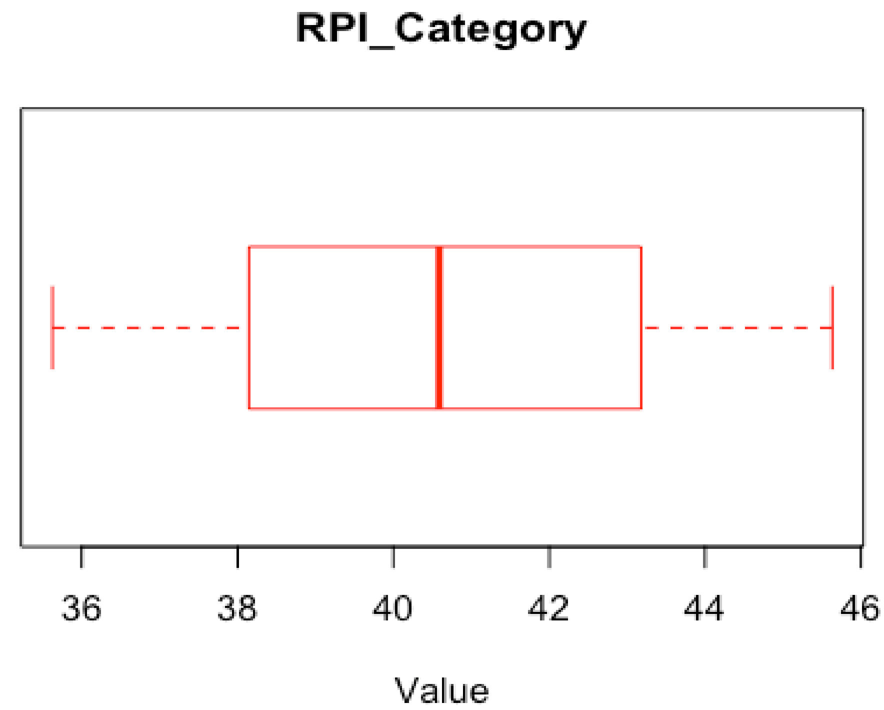

EQ_Base_Price
1.45
1.50
1.55
1.60
1.65
Value

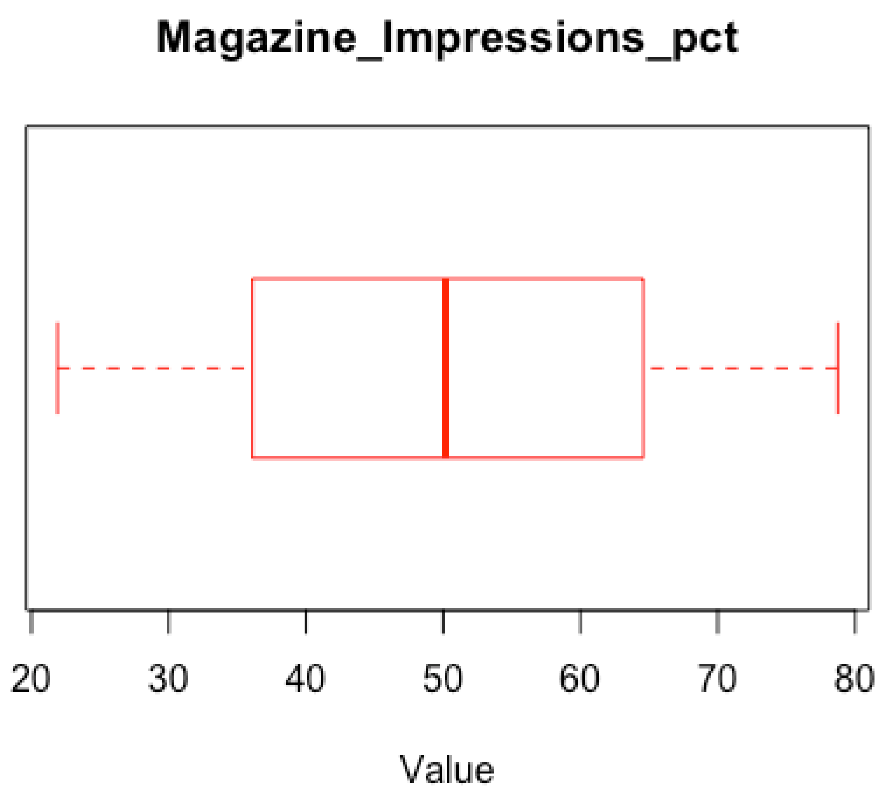

Est_ACV_Selling
3e+08
5e+08
7e+08
9e+08
Value


pct_ACV
15
20
25
30
35
40
45
Value

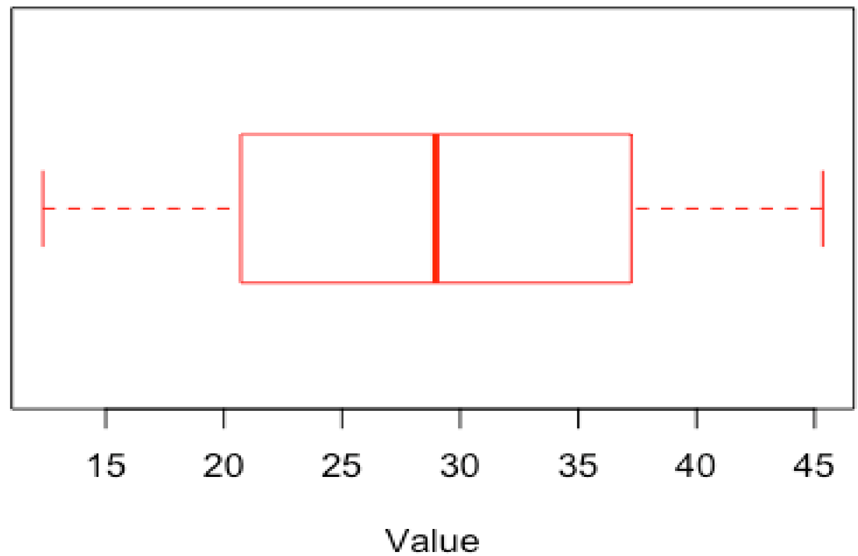

Avg_no_of_Items
2.2
2.3
2.4
2.5
2.6
2.7
2.8
2.9
Value


pct_PromoMarketDollars_Category
0.0
0.1
0.2
0.3
0.4
Value

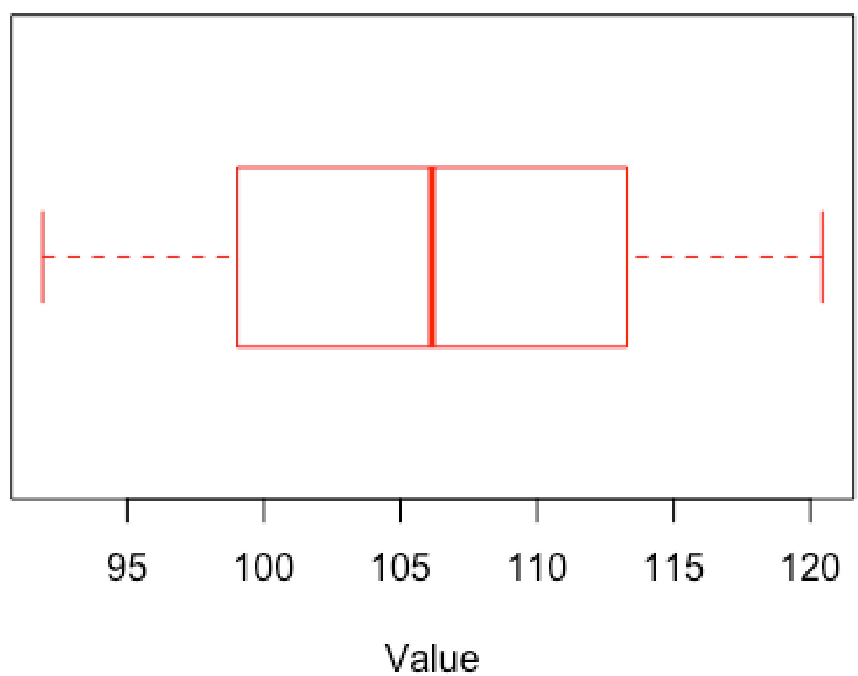

RPI_Category
36
38
40
42
44
46
Value

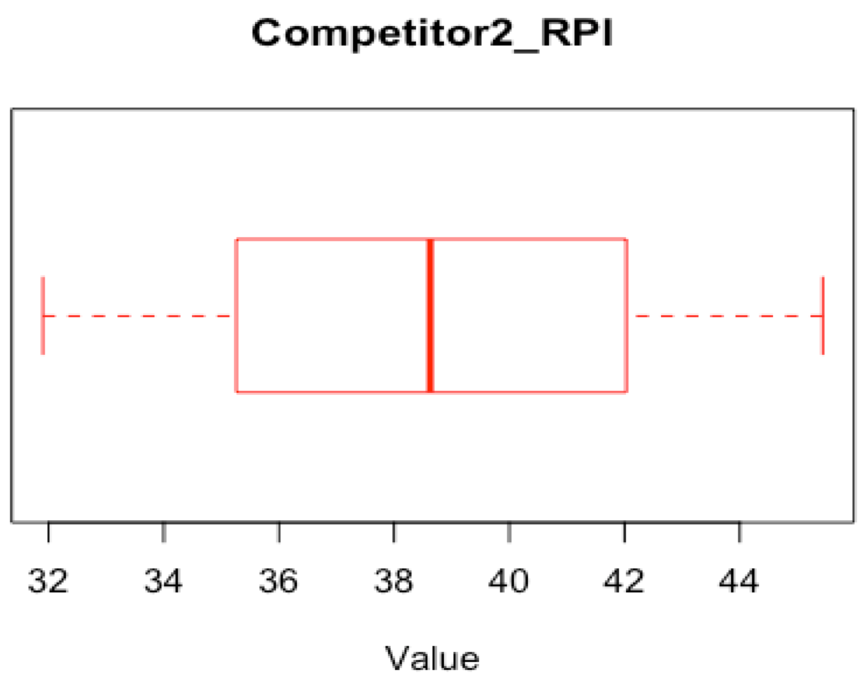

Magazine_Impressions_pct
20
30
40
50
60
70
80
Value

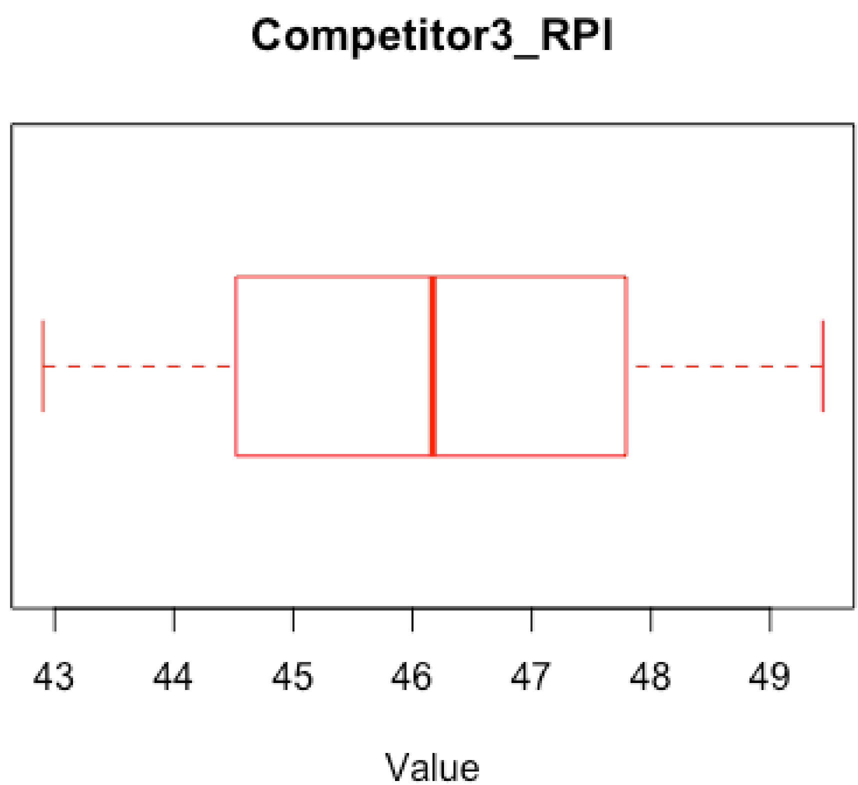
**TV_GRP**

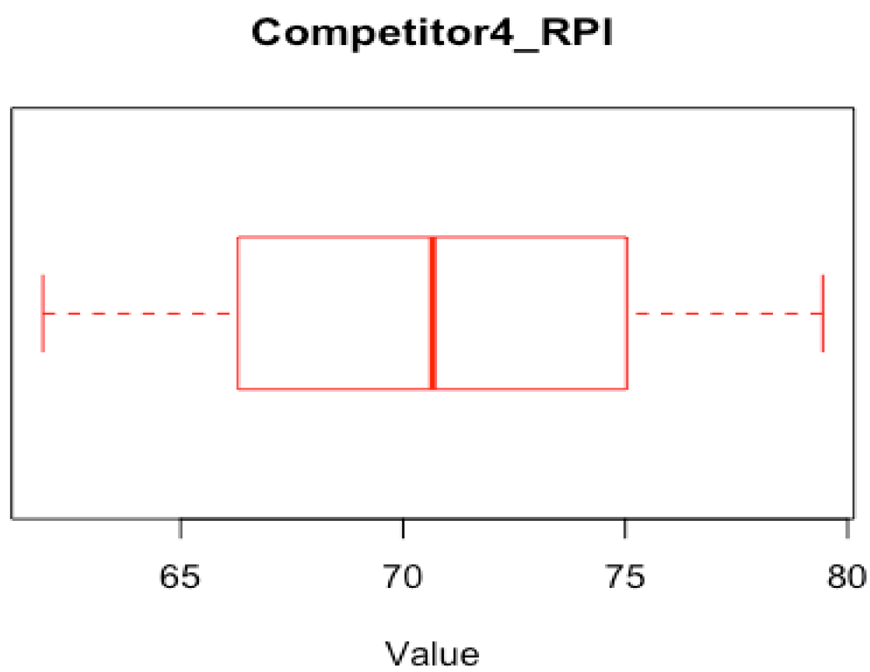


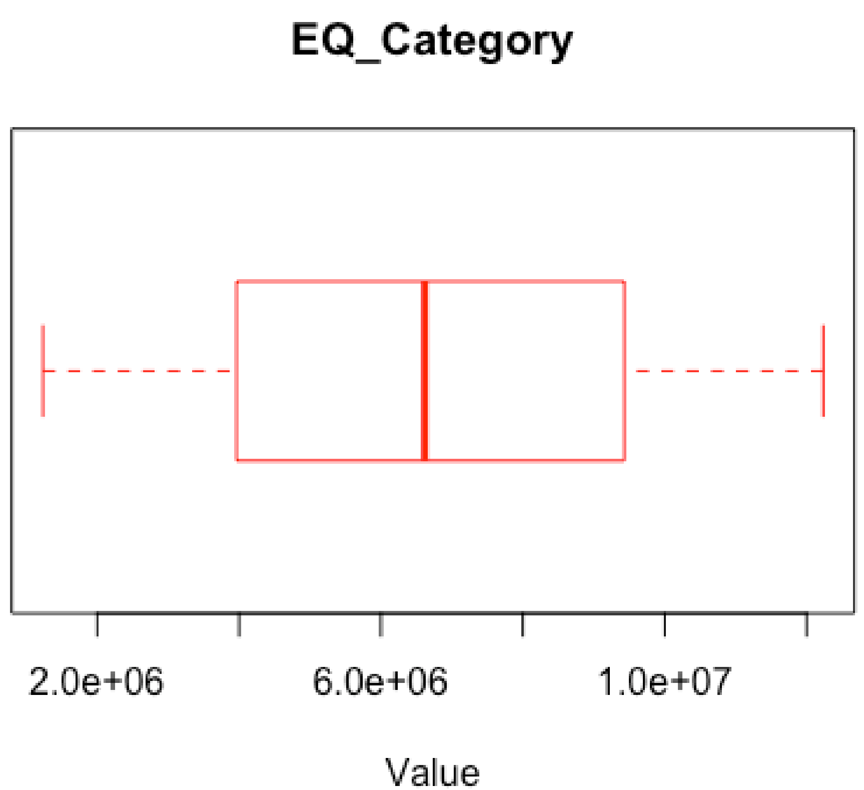

**Competitor1_RPI**

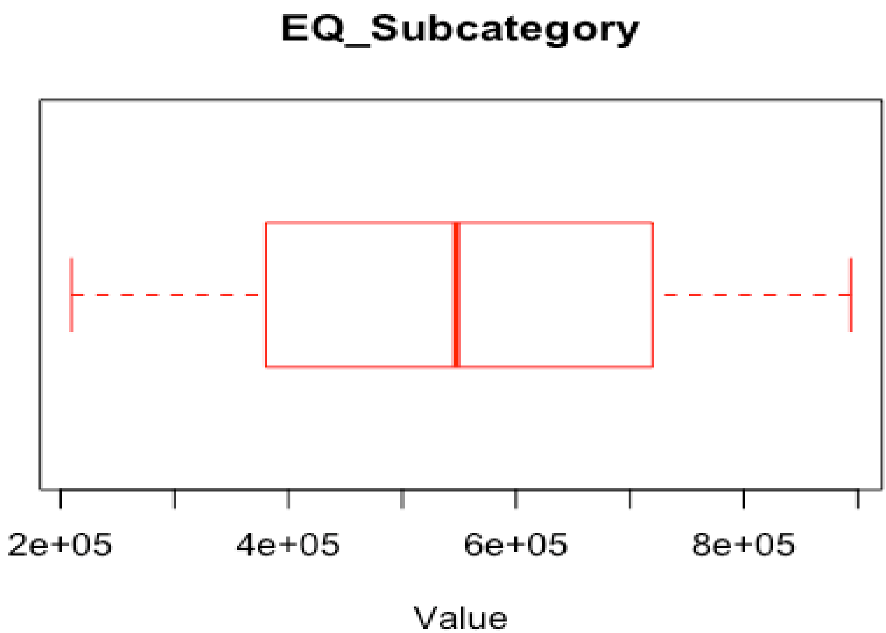


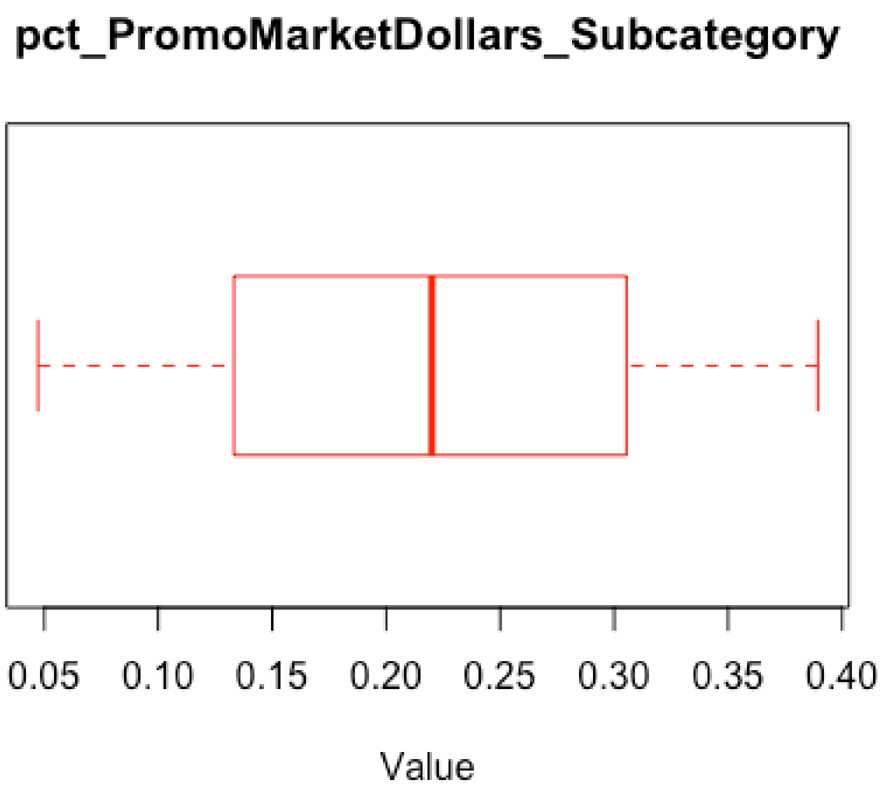

Competitor2_RPI
32
34
36
38
40
42
44
Value

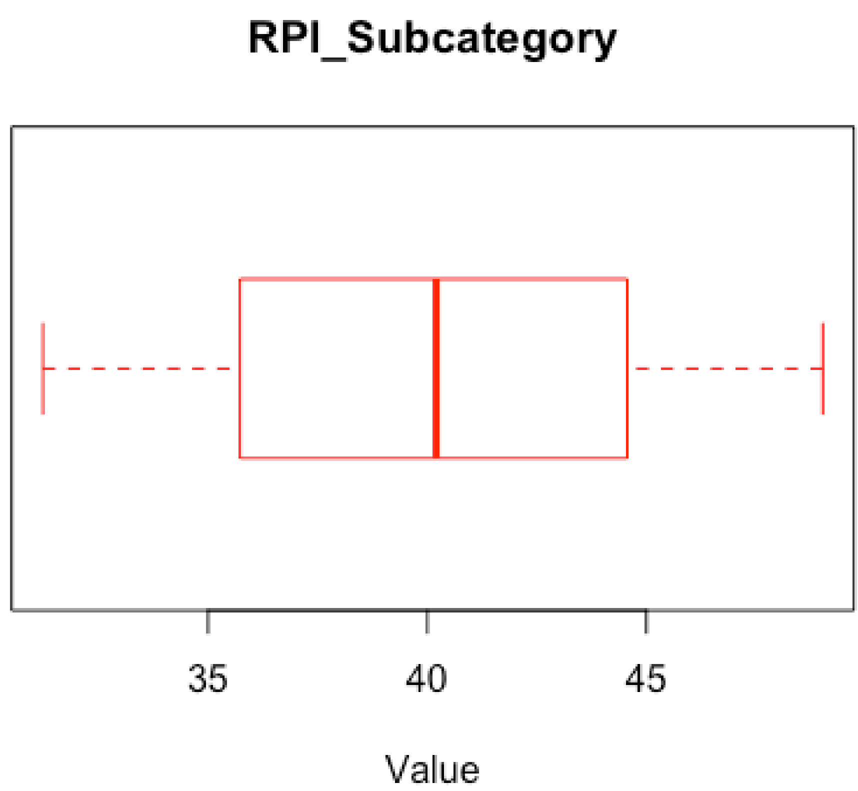

Competitor3_RPI
43
44
45
46
47
48
49
Value


Competitor4_RPI
65
70
75
80
Value


EQ_Category
2.0e+06
6.0e+06
1.0e+07
Value


EQ_Subcategory
2e+05
4e+05
6e+05
8e+05
Value


pct_PromoMarketDollars_Subcategory
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
Value




Remove the 1st column "Da"y from the day as it's not required for regression

```
sales_data_train <- sales_data[,-1]
head(sales_data_train)

## # A tibble: 6 x 38
##        EQ Social_Search_I… Social_Search_W… Digital_Impress…
##     <dbl>            <dbl>            <dbl>            <dbl>
## 1  718.           22256928            56812          7724107
## 2   25.5           4239408           105695          5844288
## 3  268.            6708500            87686         13008485
## 4  209.           36835247            70791          2520814
## 5 3482.           23693467            75610          9276779
## 6   55.2          13925382           114740          2733356
## # … with 34 more variables: Digital_Working_cost <dbl>,
## #   Print_Impressions.Ads40 <dbl>, Print_Working_Cost.Ads50 <dbl>,
## #   OOH_Impressions <dbl>, OOH_Working_Cost <dbl>, SOS_pct <dbl>,
## #   Digital_Impressions_pct <dbl>, CCFOT <dbl>, Median_Temp <dbl>,
## #   Median_Rainfall <dbl>, Fuel_Price <dbl>, Inflation <dbl>,
## #   Trade_Invest <dbl>, Brand_Equity <dbl>, Avg_EQ_Price <dbl>,
## #   Any_Promo_pct_ACV <dbl>, Any_Feat_pct_ACV <dbl>,
## #   Any_Disp_pct_ACV <dbl>, EQ_Base_Price <dbl>, Est_ACV_Selling <dbl>,
## #   pct_ACV <dbl>, Avg_no_of_Items <dbl>,
## #   pct_PromoMarketDollars_Category <dbl>, RPI_Category <dbl>,
```

```
## #   Magazine_Impressions_pct <dbl>, TV_GRP <dbl>, Competitor1_RPI <dbl>,
## #   Competitor2_RPI <dbl>, Competitor3_RPI <dbl>, Competitor4_RPI <dbl>,
## #   EQ_Category <dbl>, EQ_Subcategory <dbl>,
## #   pct_PromoMarketDollars_Subcategory <dbl>, RPI_Subcategory <dbl>
```

### Scale the data to bring it to standard normal scale

```
sales_data_scl_train <- scale(sales_data_train[,-1])

sales_data_train_new <-
data.frame(cbind(EQ=sales_data$EQ,sales_data_scl_train))
```

### Remove the outlier from the target variable “EQ” whose value is greater thne 1.5*IQR of “EQ”

```
sales_data_train_latest <- sales_data_train[sales_data_train$EQ<=912,]

#ggplot(data=sales_data_train_latest, aes(EQ, fill=EQ)) +
geom_boxplot(colour="Black")
```

### Split the train dataset into 70-30 ratio for Train and Test

```
library(caTools)

set.seed(777)

spl = sample.split(sales_data_train_latest$EQ, SplitRatio = 0.7)

train_data = subset(sales_data_train_latest, spl == TRUE)
test_data = subset(sales_data_train_latest, spl == FALSE)
```

### Build the Linear Regression Model on the train data

```
lr_model <- lm(train_data$EQ ~ . , data = train_data, )

print(lr_model)

##
## Call:
## lm(formula = train_data$EQ ~ ., data = train_data)
##
## Coefficients:
##                    (Intercept)           Social_Search_Impressions
##                     -8.018e+02                           9.227e-06
##        Social_Search_Working_cost                 Digital_Impressions
##                     -1.021e-05                          -1.489e-06
##              Digital_Working_cost             Print_Impressions.Ads40
##                     -1.570e-05                           1.748e-05
##          Print_Working_Cost.Ads50                     OOH_Impressions
##                      9.028e-06                          -2.080e-09
##                  OOH_Working_Cost                             SOS_pct
##                     -2.783e-07                          -1.064e-02
##           Digital_Impressions_pct                               CCFOT
##                     -1.234e-01                          -4.507e-02
```

```
##                            Median_Temp                          Median_Rainfall
##                              8.856e-02                                3.656e+02
##                             Fuel_Price                                Inflation
##                              1.009e+00                                2.425e+03
##                           Trade_Invest                             Brand_Equity
##                             -8.300e-04                               -1.617e+00
##                           Avg_EQ_Price                        Any_Promo_pct_ACV
##                             -2.383e-01                               -4.382e-01
##                       Any_Feat_pct_ACV                         Any_Disp_pct_ACV
##                             -7.494e-01                                3.105e-01
##                          EQ_Base_Price                          Est_ACV_Selling
##                              3.488e+01                               -1.535e-08
##                                pct_ACV                           Avg_no_of_Items
##                             -1.700e-01                               -4.232e+00
##    pct_PromoMarketDollars_Category                             RPI_Category
##                              7.679e+02                                7.623e-01
##               Magazine_Impressions_pct                                   TV_GRP
##                             -1.235e-01                               -2.190e-02
##                        Competitor1_RPI                          Competitor2_RPI
##                             -1.918e-01                                1.419e-01
##                        Competitor3_RPI                          Competitor4_RPI
##                             -2.666e-01                                4.294e-02
##                            EQ_Category                           EQ_Subcategory
##                              2.276e-05                                2.764e-04
## pct_PromoMarketDollars_Subcategory                          RPI_Subcategory
##                              6.934e+02                               -1.806e-01
```

```
summary(lr_model)
```

```
##
## Call:
## lm(formula = train_data$EQ ~ ., data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -477.29  -92.28  -20.61   71.53  514.18
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -8.018e+02  1.815e+02  -4.416 1.02e-05
## Social_Search_Impressions          9.227e-06  1.657e-07  55.692  < 2e-16
## Social_Search_Working_cost        -1.021e-05  5.018e-05  -0.203  0.83877
## Digital_Impressions               -1.489e-06  4.562e-07  -3.263  0.00111
## Digital_Working_cost              -1.570e-05  1.401e-05  -1.120  0.26262
## Print_Impressions.Ads40            1.748e-05  2.033e-05   0.860  0.38985
## Print_Working_Cost.Ads50           9.028e-06  3.168e-05   0.285  0.77567
## OOH_Impressions                   -2.080e-09  7.538e-09  -0.276  0.78258
## OOH_Working_Cost                  -2.783e-07  1.711e-06  -0.163  0.87083
## SOS_pct                           -1.064e-02  1.206e-01  -0.088  0.92967
## Digital_Impressions_pct           -1.234e-01  1.192e-01  -1.035  0.30062
```

```
## CCFOT                               -4.507e-02  6.685e-02  -0.674  0.50019
## Median_Temp                          8.856e-02  1.288e-01   0.688  0.49160
## Median_Rainfall                      3.656e+02  6.319e+00  57.857  < 2e-16
## Fuel_Price                           1.009e+00  2.318e+00   0.435  0.66344
## Inflation                            2.425e+03  5.145e+01  47.125  < 2e-16
## Trade_Invest                        -8.300e-04  6.117e-04  -1.357  0.17489
## Brand_Equity                        -1.617e+00  3.766e+00  -0.429  0.66769
## Avg_EQ_Price                        -2.383e-01  3.391e-01  -0.703  0.48220
## Any_Promo_pct_ACV                   -4.382e-01  3.652e-01  -1.200  0.23018
## Any_Feat_pct_ACV                    -7.494e-01  1.499e+00  -0.500  0.61704
## Any_Disp_pct_ACV                     3.105e-01  1.466e+00   0.212  0.83229
## EQ_Base_Price                        3.488e+01  2.229e+01   1.565  0.11772
## Est_ACV_Selling                     -1.535e-08  9.321e-09  -1.647  0.09964
## pct_ACV                             -1.700e-01  1.876e-01  -0.906  0.36492
## Avg_no_of_Items                     -4.232e+00  8.957e+00  -0.472  0.63661
## pct_PromoMarketDollars_Category      7.679e+02  1.407e+01  54.583  < 2e-16
## RPI_Category                         7.623e-01  6.015e-01   1.267  0.20508
## Magazine_Impressions_pct            -1.235e-01  1.056e-01  -1.169  0.24236
## TV_GRP                              -2.190e-02  1.827e-01  -0.120  0.90459
## Competitor1_RPI                     -1.918e-01  2.105e-01  -0.911  0.36223
## Competitor2_RPI                      1.419e-01  4.434e-01   0.320  0.74896
## Competitor3_RPI                     -2.666e-01  9.204e-01  -0.290  0.77210
## Competitor4_RPI                      4.294e-02  3.417e-01   0.126  0.89999
## EQ_Category                          2.276e-05  5.534e-07  41.133  < 2e-16
## EQ_Subcategory                       2.764e-04  8.908e-06  31.023  < 2e-16
## pct_PromoMarketDollars_Subcategory  6.934e+02  1.794e+01  38.652  < 2e-16
## RPI_Subcategory                     -1.806e-01  3.380e-01  -0.534  0.59315
##
## (Intercept)                         ***
## Social_Search_Impressions           ***
## Social_Search_Working_cost
## Digital_Impressions                 **
## Digital_Working_cost
## Print_Impressions.Ads40
## Print_Working_Cost.Ads50
## OOH_Impressions
## OOH_Working_Cost
## SOS_pct
## Digital_Impressions_pct
## CCFOT
## Median_Temp
## Median_Rainfall                     ***
## Fuel_Price
## Inflation                           ***
## Trade_Invest
## Brand_Equity
## Avg_EQ_Price
## Any_Promo_pct_ACV
## Any_Feat_pct_ACV
## Any_Disp_pct_ACV
```

```
## EQ_Base_Price
## Est_ACV_Selling                    .
## pct_ACV
## Avg_no_of_Items
## pct_PromoMarketDollars_Category    ***
## RPI_Category
## Magazine_Impressions_pct
## TV_GRP
## Competitor1_RPI
## Competitor2_RPI
## Competitor3_RPI
## Competitor4_RPI
## EQ_Category                        ***
## EQ_Subcategory                     ***
## pct_PromoMarketDollars_Subcategory ***
## RPI_Subcategory
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 142.9 on 6745 degrees of freedom
## Multiple R-squared:  0.5997, Adjusted R-squared:  0.5975
## F-statistic: 273.1 on 37 and 6745 DF,  p-value: < 2.2e-16
```

We see that the Adjusted R-square of the model is 59% and RSE is 142.9

### Predict on test data using the built Linear Regression model

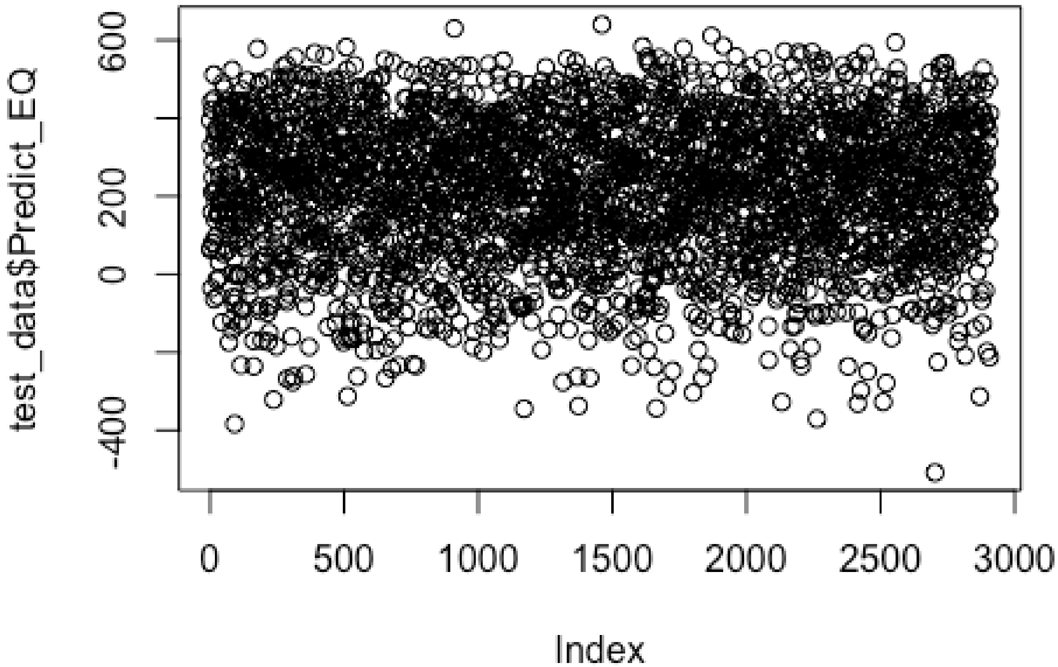
```
test_data$Predict_EQ <- predict(lr_model,newdata=test_data)
```

### Find the MAPE value of the model

```
#install.packages("MLmetrics")
library(MLmetrics)

##
## Attaching package: 'MLmetrics'

## The following object is masked from 'package:base':
##
##     Recall

MAPE(y_pred = test_data$Predict_EQ,y_true = test_data$EQ)

## [1] 19.38867
```

### Scatter plot of the predicted values

```
plot(test_data$Predict_EQ)
```



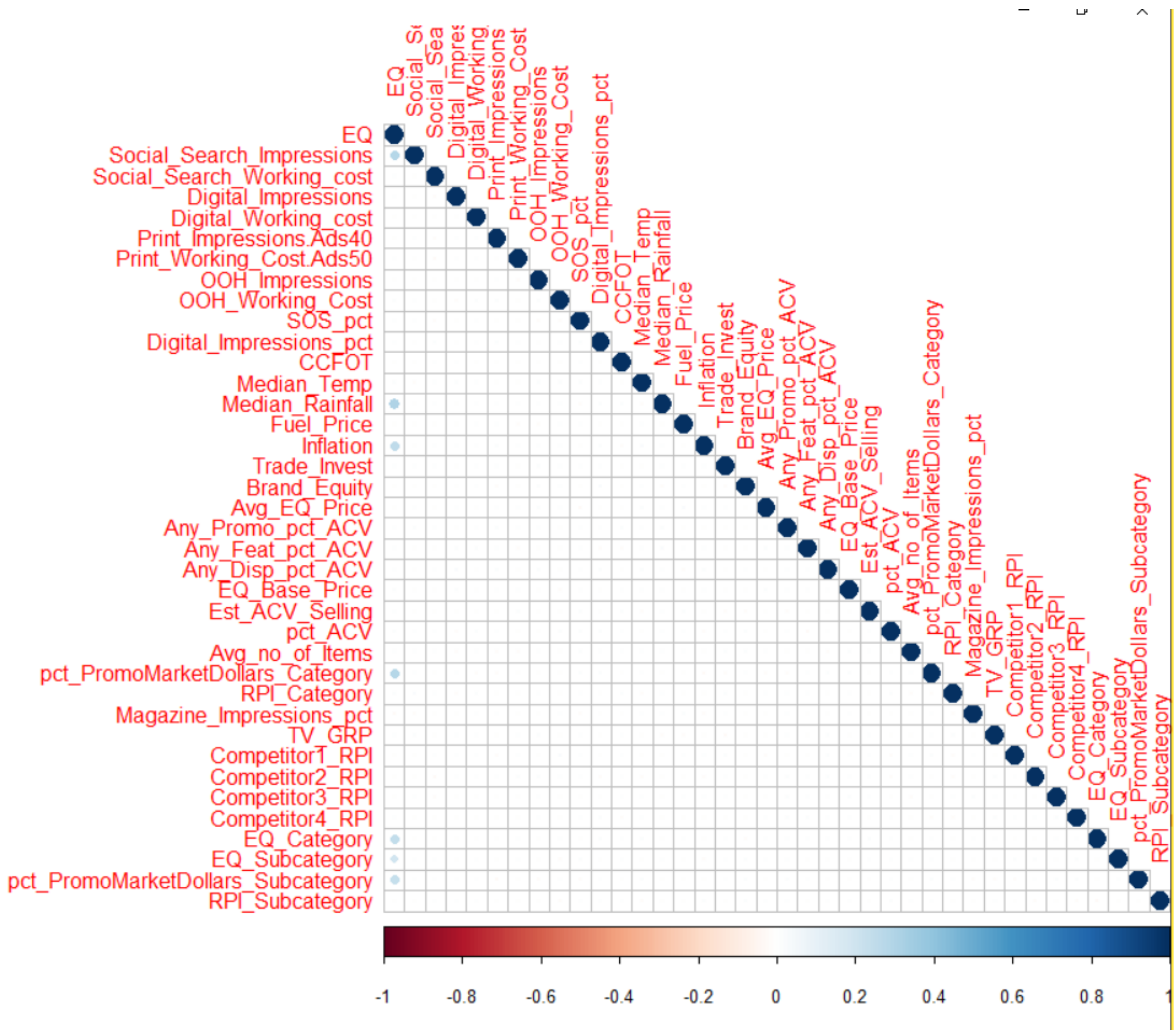

### Show the variable Importance Plot of the Linear Regression Model

```
library(caret)

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following objects are masked from 'package:MLmetrics':
##
##     MAE, RMSE

varimpplot <- as.data.frame(varImp(lr_model))

varimpplot <- data.frame(overall = varimpplot$Overall,names =
rownames(varimpplot))

varimpplot[order(varimpplot$overall,decreasing = T),]

##        overall                              names
## 13 57.85684286                    Median_Rainfall
## 1  55.69239058           Social_Search_Impressions
## 26 54.58251341    pct_PromoMarketDollars_Category
## 15 47.12488573                          Inflation
```

```
## 34 41.13266131                         EQ_Category
## 36 38.65219365 pct_PromoMarketDollars_Subcategory
## 35 31.02274737                      EQ_Subcategory
## 3   3.26269135                 Digital_Impressions
## 23  1.64682456                     Est_ACV_Selling
## 22  1.56460558                       EQ_Base_Price
## 16  1.35678772                        Trade_Invest
## 27  1.26733803                        RPI_Category
## 19  1.19999141                   Any_Promo_pct_ACV
## 28  1.16920761              Magazine_Impressions_pct
## 4   1.12031184                 Digital_Working_cost
## 10  1.03519026               Digital_Impressions_pct
## 30  0.91118215                     Competitor1_RPI
## 24  0.90609440                             pct_ACV
## 5   0.85995253              Print_Impressions.Ads40
## 18  0.70280053                        Avg_EQ_Price
## 12  0.68780770                         Median_Temp
## 11  0.67422342                               CCFOT
## 37  0.53430305                     RPI_Subcategory
## 20  0.50006794                    Any_Feat_pct_ACV
## 25  0.47246518                     Avg_no_of_Items
## 14  0.43519184                          Fuel_Price
## 17  0.42934301                        Brand_Equity
## 31  0.32002209                     Competitor2_RPI
## 32  0.28964346                     Competitor3_RPI
## 6   0.28498341             Print_Working_Cost.Ads50
## 7   0.27596947                     OOH_Impressions
## 21  0.21177796                    Any_Disp_pct_ACV
## 2   0.20348080           Social_Search_Working_cost
## 8   0.16260535                    OOH_Working_Cost
## 33  0.12567633                     Competitor4_RPI
## 29  0.11987122                              TV_GRP
## 9   0.08825811                             SOS_pct
```

### Build the correlation plat and see the orrelation of the variables

```
library(corrplot)

## corrplot 0.84 loaded

sales_data_cor <- as.matrix(sales_data[,2:39])

corplot <- corrplot(cor(sales_data_cor),type = "lower")

write.csv(corrplot(corr_re) , "cor_plot.csv")
```




**Based on the varible importance and seeing the significant varibales from the model which p-value are very low build the linear regression model again**

Important Variables :-

Median_Rainfall
Social_Search_Impressions
pct_PromoMarketDollars_Category
Inflation
EQ_Category
pct_PromoMarketDollars_Subcategory
EQ_Subcategory
Digital_Impressions
Est_ACV_Selling

```r
lr_model_new <- lm(train_data$EQ ~
Median_Rainfall+Social_Search_Impressions+pct_PromoMarketDollars_Category+Inf
lation+EQ_Category+pct_PromoMarketDollars_Subcategory+EQ_Subcategory+Digital_
Impressions+Est_ACV_Selling, data = train_data)

print(lr_model_new)

##
## Call:
## lm(formula = train_data$EQ ~ Median_Rainfall + Social_Search_Impressions +
##     pct_PromoMarketDollars_Category + Inflation + EQ_Category +
##     pct_PromoMarketDollars_Subcategory + EQ_Subcategory +
Digital_Impressions +
##     Est_ACV_Selling, data = train_data)
##
## Coefficients:
##                        (Intercept)                      Median_Rainfall
##                         -8.564e+02                            3.650e+02
##          Social_Search_Impressions      pct_PromoMarketDollars_Category
##                          9.217e-06                            7.665e+02
##                          Inflation                          EQ_Category
##                          2.421e+03                            2.275e-05
## pct_PromoMarketDollars_Subcategory                       EQ_Subcategory
##                          6.923e+02                            2.761e-04
##                Digital_Impressions                      Est_ACV_Selling
##                         -1.445e-06                           -1.497e-08

summary(lr_model_new)

##
## Call:
## lm(formula = train_data$EQ ~ Median_Rainfall + Social_Search_Impressions +
##     pct_PromoMarketDollars_Category + Inflation + EQ_Category +
##     pct_PromoMarketDollars_Subcategory + EQ_Subcategory +
Digital_Impressions +
##     Est_ACV_Selling, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -481.65  -92.92  -20.55   70.46  513.18
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        -8.564e+02  1.320e+01 -64.877  < 2e-16
## Median_Rainfall                     3.650e+02  6.298e+00  57.957  < 2e-16
## Social_Search_Impressions           9.217e-06  1.653e-07  55.772  < 2e-16
## pct_PromoMarketDollars_Category     7.665e+02  1.402e+01  54.660  < 2e-16
## Inflation                           2.421e+03  5.134e+01  47.159  < 2e-16
## EQ_Category                         2.275e-05  5.521e-07  41.207  < 2e-16
## pct_PromoMarketDollars_Subcategory  6.923e+02  1.786e+01  38.770  < 2e-16
```

```
## EQ_Subcategory                       2.761e-04  8.881e-06  31.092  < 2e-16
## Digital_Impressions                 -1.445e-06  4.552e-07  -3.174  0.00151
## Est_ACV_Selling                     -1.497e-08  9.296e-09  -1.610  0.10746
##
## (Intercept)                         ***
## Median_Rainfall                     ***
## Social_Search_Impressions           ***
## pct_PromoMarketDollars_Category     ***
## Inflation                           ***
## EQ_Category                         ***
## pct_PromoMarketDollars_Subcategory ***
## EQ_Subcategory                      ***
## Digital_Impressions                 **
## Est_ACV_Selling
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 142.8 on 6773 degrees of freedom
## Multiple R-squared:  0.5987, Adjusted R-squared:  0.5982
## F-statistic:  1123 on 9 and 6773 DF,  p-value: < 2.2e-16
```

### Predict on testset using the above built Linear Regression model

```
test_data$Predict_EQ <- NULL

test_data$Predict_EQ <- predict(lr_model_new,newdata=test_data)

MAPE(y_pred = test_data$Predict_EQ,y_true = test_data$EQ)

## [1] 19.44759
```

### Read the validation dataset given (Test-dataset-v1.xlsx)

```
sales_data_test <- read_excel("/Users/dinesh/Downloads/Test dataset v1.xlsx")

head(sales_data_test)

## # A tibble: 6 x 39
##   Period    EQ Social_Search_I… Social_Search_W… Digital_Impress…
##   <chr>  <dbl>            <dbl>            <dbl>            <dbl>
## 1 2016 …  505.          2019283             5493           37148.
## 2 2016 …  490.          4564738            12938           50887.
## 3 2016 …  479.          1029384             6546          253333.
## 4 2016 …  489.           902938             3928         3426239
## 5 2016 …  477.          1343454            28374          552198.
## 6 2016 …  488.          2434564            59483           29892.
## # … with 34 more variables: Digital_Working_cost <dbl>,
## #   Print_Impressions.Ads40 <dbl>, Print_Working_Cost.Ads50 <dbl>,
## #   OOH_Impressions <dbl>, OOH_Working_Cost <dbl>, SOS_pct <dbl>,
## #   Digital_Impressions_pct <dbl>, CCFOT <dbl>, Median_Temp <dbl>,
## #   Median_Rainfall <dbl>, Fuel_Price <dbl>, Inflation <dbl>,
## #   Trade_Invest <dbl>, Brand_Equity <dbl>, Avg_EQ_Price <dbl>,
```

```
## #   Any_Promo_pct_ACV <dbl>, Any_Feat_pct_ACV <dbl>,
## #   Any_Disp_pct_ACV <dbl>, EQ_Base_Price <dbl>, Est_ACV_Selling <dbl>,
## #   pct_ACV <dbl>, Avg_no_of_Items <dbl>,
## #   pct_PromoMarketDollars_Category <dbl>, RPI_Category <dbl>,
## #   Magazine_Impressions_pct <dbl>, TV_GRP <dbl>, Competitor1_RPI <dbl>,
## #   Competitor2_RPI <dbl>, Competitor3_RPI <dbl>, Competitor4_RPI <dbl>,
## #   EQ_Category <dbl>, EQ_Subcategory <dbl>,
## #   pct_PromoMarketDollars_Subcategory <dbl>, RPI_Subcategory <dbl>
```

**Remove the Period column**

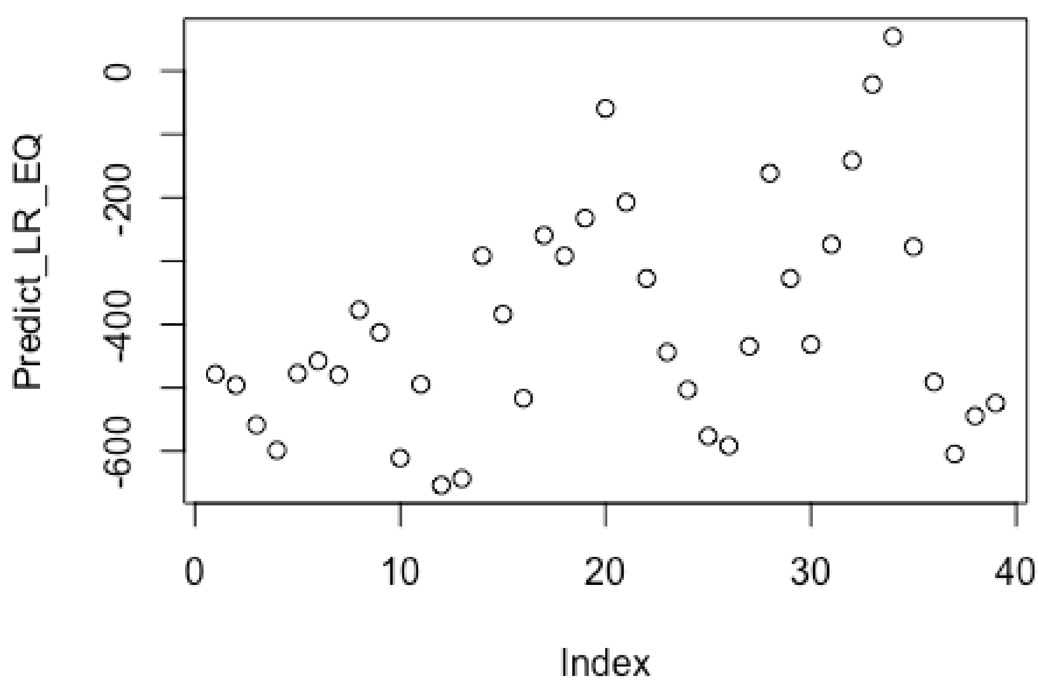
```
sales_data_test <- sales_data_test[,-1]
```

**Predict on validation data using the built Final Linear Regression model**

```
Predict_LR_EQ <-predict(lr_model_new,newdata=sales_data_test)
```

**Plot the predicted sales**

```
plot(Predict_LR_EQ)
```




**Buid a Bayesian Model**

```
#install.packages("BAS")
library(BAS)

model_bays <- bas.lm(train_data$EQ ~ .,
```

```
                     data = train_data,
                     method = "MCMC",
                     prior = "ZS-null",
                     modelprior = uniform())
```

### Show the summary of the Bayesian model

```
summary(model_bays)
```

```
##                                    P(B != 0 | Y)   model 1     model 2
## Intercept                             1.00000000    1.0000   1.0000000
## Social_Search_Impressions             0.99998283    1.0000   1.0000000
## Social_Search_Working_cost            0.02415085    0.0000   0.0000000
## Digital_Impressions                   0.77088451    1.0000   0.0000000
## Digital_Working_cost                  0.03985252    0.0000   0.0000000
## Print_Impressions.Ads40               0.03178806    0.0000   0.0000000
## Print_Working_Cost.Ads50              0.02404804    0.0000   0.0000000
## OOH_Impressions                       0.02347641    0.0000   0.0000000
## OOH_Working_Cost                      0.02332115    0.0000   0.0000000
## SOS_pct                               0.02273903    0.0000   0.0000000
## Digital_Impressions_pct               0.04181232    0.0000   0.0000000
## CCFOT                                 0.02974339    0.0000   0.0000000
## Median_Temp                           0.02960758    0.0000   0.0000000
## Median_Rainfall                       0.99999599    1.0000   1.0000000
## Fuel_Price                            0.02347202    0.0000   0.0000000
## Inflation                             0.99995174    1.0000   1.0000000
## Trade_Invest                          0.05688896    0.0000   0.0000000
## Brand_Equity                          0.02477093    0.0000   0.0000000
## Avg_EQ_Price                          0.02776051    0.0000   0.0000000
## Any_Promo_pct_ACV                     0.04089680    0.0000   0.0000000
## Any_Feat_pct_ACV                      0.02514954    0.0000   0.0000000
## Any_Disp_pct_ACV                      0.02337551    0.0000   0.0000000
## EQ_Base_Price                         0.07177105    0.0000   0.0000000
## Est_ACV_Selling                       0.07777843    0.0000   0.0000000
## pct_ACV                               0.03377285    0.0000   0.0000000
## Avg_no_of_Items                       0.02601204    0.0000   0.0000000
## pct_PromoMarketDollars_Category       0.99999104    1.0000   1.0000000
## RPI_Category                          0.04816437    0.0000   0.0000000
## Magazine_Impressions_pct              0.04128017    0.0000   0.0000000
## TV_GRP                                0.02316818    0.0000   0.0000000
## Competitor1_RPI                       0.03620682    0.0000   0.0000000
## Competitor2_RPI                       0.02393131    0.0000   0.0000000
## Competitor3_RPI                       0.02380466    0.0000   0.0000000
## Competitor4_RPI                       0.02381229    0.0000   0.0000000
## EQ_Category                           0.99997025    1.0000   1.0000000
## EQ_Subcategory                        0.99997959    1.0000   1.0000000
## pct_PromoMarketDollars_Subcategory    0.99999466    1.0000   1.0000000
## RPI_Subcategory                       0.02681236    0.0000   0.0000000
## BF                                            NA    1.0000   0.3210587
## PostProbs                                     NA    0.2891   0.0928000
## R2                                            NA    0.5986   0.5980000
```

```
## dim                                               NA    9.0000    8.0000000
## logmarg                                           NA 3060.0387 3058.9025475
##                                         model 3      model 4      model 5
## Intercept                          1.000000e+00 1.000000e+00 1.000000e+00
## Social_Search_Impressions          1.000000e+00 1.000000e+00 1.000000e+00
## Social_Search_Working_cost         0.000000e+00 0.000000e+00 0.000000e+00
## Digital_Impressions                1.000000e+00 1.000000e+00 1.000000e+00
## Digital_Working_cost               0.000000e+00 0.000000e+00 0.000000e+00
## Print_Impressions.Ads40            0.000000e+00 0.000000e+00 0.000000e+00
## Print_Working_Cost.Ads50           0.000000e+00 0.000000e+00 0.000000e+00
## OOH_Impressions                    0.000000e+00 0.000000e+00 0.000000e+00
## OOH_Working_Cost                   0.000000e+00 0.000000e+00 0.000000e+00
## SOS_pct                            0.000000e+00 0.000000e+00 0.000000e+00
## Digital_Impressions_pct            0.000000e+00 0.000000e+00 0.000000e+00
## CCFOT                              0.000000e+00 0.000000e+00 0.000000e+00
## Median_Temp                        0.000000e+00 0.000000e+00 0.000000e+00
## Median_Rainfall                    1.000000e+00 1.000000e+00 1.000000e+00
## Fuel_Price                         0.000000e+00 0.000000e+00 0.000000e+00
## Inflation                          1.000000e+00 1.000000e+00 1.000000e+00
## Trade_Invest                       0.000000e+00 0.000000e+00 1.000000e+00
## Brand_Equity                       0.000000e+00 0.000000e+00 0.000000e+00
## Avg_EQ_Price                       0.000000e+00 0.000000e+00 0.000000e+00
## Any_Promo_pct_ACV                  0.000000e+00 0.000000e+00 0.000000e+00
## Any_Feat_pct_ACV                   0.000000e+00 0.000000e+00 0.000000e+00
## Any_Disp_pct_ACV                   0.000000e+00 0.000000e+00 0.000000e+00
## EQ_Base_Price                      0.000000e+00 1.000000e+00 0.000000e+00
## Est_ACV_Selling                    1.000000e+00 0.000000e+00 0.000000e+00
## pct_ACV                            0.000000e+00 0.000000e+00 0.000000e+00
## Avg_no_of_Items                    0.000000e+00 0.000000e+00 0.000000e+00
## pct_PromoMarketDollars_Category    1.000000e+00 1.000000e+00 1.000000e+00
## RPI_Category                       0.000000e+00 0.000000e+00 0.000000e+00
## Magazine_Impressions_pct           0.000000e+00 0.000000e+00 0.000000e+00
## TV_GRP                             0.000000e+00 0.000000e+00 0.000000e+00
## Competitor1_RPI                    0.000000e+00 0.000000e+00 0.000000e+00
## Competitor2_RPI                    0.000000e+00 0.000000e+00 0.000000e+00
## Competitor3_RPI                    0.000000e+00 0.000000e+00 0.000000e+00
## Competitor4_RPI                    0.000000e+00 0.000000e+00 0.000000e+00
## EQ_Category                        1.000000e+00 1.000000e+00 1.000000e+00
## EQ_Subcategory                     1.000000e+00 1.000000e+00 1.000000e+00
## pct_PromoMarketDollars_Subcategory 1.000000e+00 1.000000e+00 1.000000e+00
## RPI_Subcategory                    0.000000e+00 0.000000e+00 0.000000e+00
## BF                                 8.363296e-02 7.648941e-02 5.902561e-02
## PostProbs                          2.380000e-02 2.090000e-02 1.750000e-02
## R2                                 5.987000e-01 5.987000e-01 5.987000e-01
## dim                                1.000000e+01 1.000000e+01 1.000000e+01
## logmarg                            3.057557e+03 3.057468e+03 3.057209e+03
```

### Based on the above Bayesian Model below are the important varibales based on the Probablity column.

We see that it's same significant varibales as we got from the Linear Regressio model.

Median_Rainfall pct_PromoMarketDollars_Subcategory pct_PromoMarketDollars_Category Social_Search_Impressions EQ_Subcategory EQ_Category Inflation Digital_Impressions Est_ACV_Selling

### Predict the sales on the validation dataset using the above Bayesian Model

```
Predict_bays_EQ <- predict(model_bays, sales_data_test, estimator="BMA",
interval = "predict", se.fit=TRUE)
```

### Find out the MAPE value of the Bayesian Model

```
MAPE(y_pred = Predict_bays_EQ$Ybma,y_true = sales_data_test$EQ)

## [1] 2.274757
```

### Dataframe for predicted values of Linear and bayesian Models

```
actual_predicted_out <-
data.frame(cbind(ActualEQ=sales_data_test$EQ,LR_Predited_EQ=Predict_LR_EQ,

Bayesian_predited_EQ=Predict_bays_EQ$Ybma))

write.csv(actual_predicted_out,"acutal_predicted_out.csv")
```

## To forcast for the next 6 period on the validation dataset-

### Read the validation dataset

```
sales_data_ts <- read_excel("/Users/dinesh/Downloads/Test dataset v1.xlsx")

head(sales_data_ts)

## # A tibble: 6 x 39
##   Period    EQ Social_Search_I… Social_Search_W… Digital_Impress…
##   <chr>  <dbl>            <dbl>            <dbl>            <dbl>
## 1 2016 …  505.          2019283             5493           37148.
## 2 2016 …  490.          4564738            12938           50887.
## 3 2016 …  479.          1029384             6546          253333.
## 4 2016 …  489.           902938             3928         3426239
## 5 2016 …  477.          1343454            28374          552198.
## 6 2016 …  488.          2434564            59483           29892.
## # … with 34 more variables: Digital_Working_cost <dbl>,
## #   Print_Impressions.Ads40 <dbl>, Print_Working_Cost.Ads50 <dbl>,
## #   OOH_Impressions <dbl>, OOH_Working_Cost <dbl>, SOS_pct <dbl>,
## #   Digital_Impressions_pct <dbl>, CCFOT <dbl>, Median_Temp <dbl>,
## #   Median_Rainfall <dbl>, Fuel_Price <dbl>, Inflation <dbl>,
## #   Trade_Invest <dbl>, Brand_Equity <dbl>, Avg_EQ_Price <dbl>,
## #   Any_Promo_pct_ACV <dbl>, Any_Feat_pct_ACV <dbl>,
## #   Any_Disp_pct_ACV <dbl>, EQ_Base_Price <dbl>, Est_ACV_Selling <dbl>,
```

```
## #   pct_ACV <dbl>, Avg_no_of_Items <dbl>,
## #   pct_PromoMarketDollars_Category <dbl>, RPI_Category <dbl>,
## #   Magazine_Impressions_pct <dbl>, TV_GRP <dbl>, Competitor1_RPI <dbl>,
## #   Competitor2_RPI <dbl>, Competitor3_RPI <dbl>, Competitor4_RPI <dbl>,
## #   EQ_Category <dbl>, EQ_Subcategory <dbl>,
## #   pct_PromoMarketDollars_Subcategory <dbl>, RPI_Subcategory <dbl>
```

### Filter the sales column to forcast

```
sales_data_ts_new <- sales_data_ts[,2]

head(sales_data_ts_new)

## # A tibble: 6 x 1
##       EQ
##    <dbl>
## 1  505.
## 2  490.
## 3  479.
## 4  489.
## 5  477.
## 6  488.
```

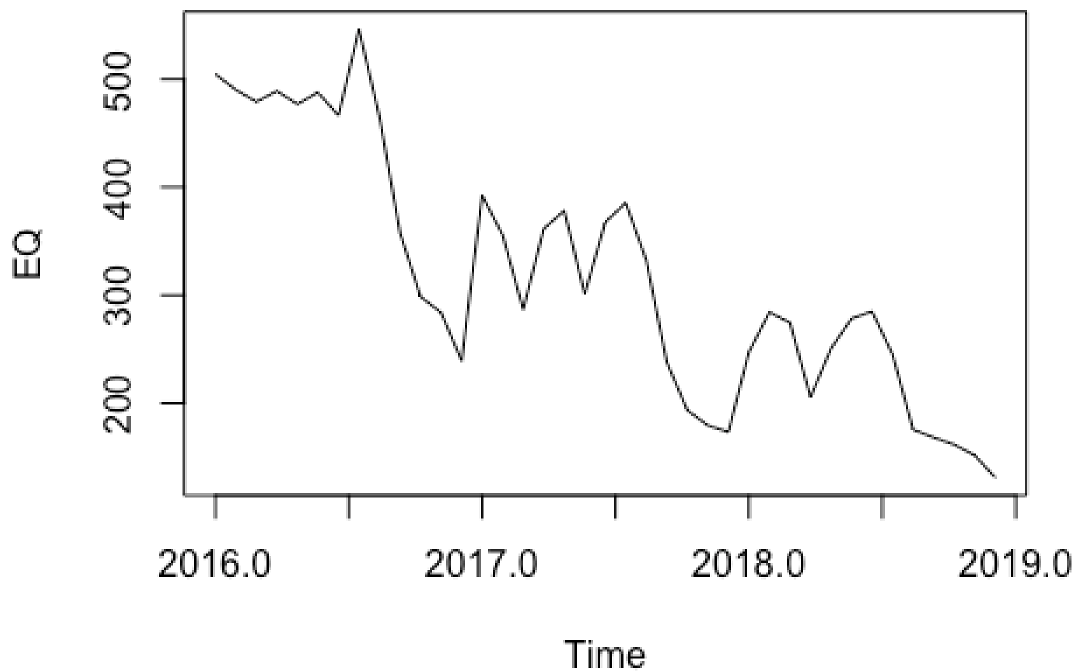
### Create the time series dataframe

```
sales_data.ts <- ts(sales_data_ts_new , start = c(2016,1) , end = c(2018,13)
,frequency = 13)

sales_data.ts

## Time Series:
## Start = c(2016, 1)
## End = c(2018, 13)
## Frequency = 13
##              EQ
##  [1,] 504.7849
##  [2,] 490.2265
##  [3,] 479.2447
##  [4,] 489.0574
##  [5,] 477.0320
##  [6,] 487.8553
##  [7,] 466.3993
##  [8,] 546.0531
##  [9,] 464.9256
## [10,] 357.6487
## [11,] 298.5533
## [12,] 283.7974
## [13,] 239.2316
## [14,] 392.3264
## [15,] 355.6523
## [16,] 286.7056
## [17,] 361.4447
```

```
## [18,] 378.2739
## [19,] 300.9221
## [20,] 367.5470
## [21,] 385.5379
## [22,] 332.1504
## [23,] 237.7136
## [24,] 193.3008
## [25,] 179.2925
## [26,] 173.2373
## [27,] 247.3155
## [28,] 284.1833
## [29,] 274.4308
## [30,] 205.5000
## [31,] 250.5551
## [32,] 278.3175
## [33,] 284.8955
## [34,] 244.9314
## [35,] 175.4323
## [36,] 168.1067
## [37,] 161.5293
## [38,] 151.6422
## [39,] 130.9374
```

### Plot the time service dataframe

```
library(forecast)

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff

plot(sales_data.ts)
```




####

Decompose the above time series df

```
decompose(sales_data.ts)

## $x
## Time Series:
## Start = c(2016, 1)
## End = c(2018, 13)
## Frequency = 13
##             EQ
##  [1,] 504.7849
##  [2,] 490.2265
##  [3,] 479.2447
##  [4,] 489.0574
##  [5,] 477.0320
##  [6,] 487.8553
##  [7,] 466.3993
##  [8,] 546.0531
##  [9,] 464.9256
## [10,] 357.6487
## [11,] 298.5533
## [12,] 283.7974
## [13,] 239.2316
## [14,] 392.3264
```

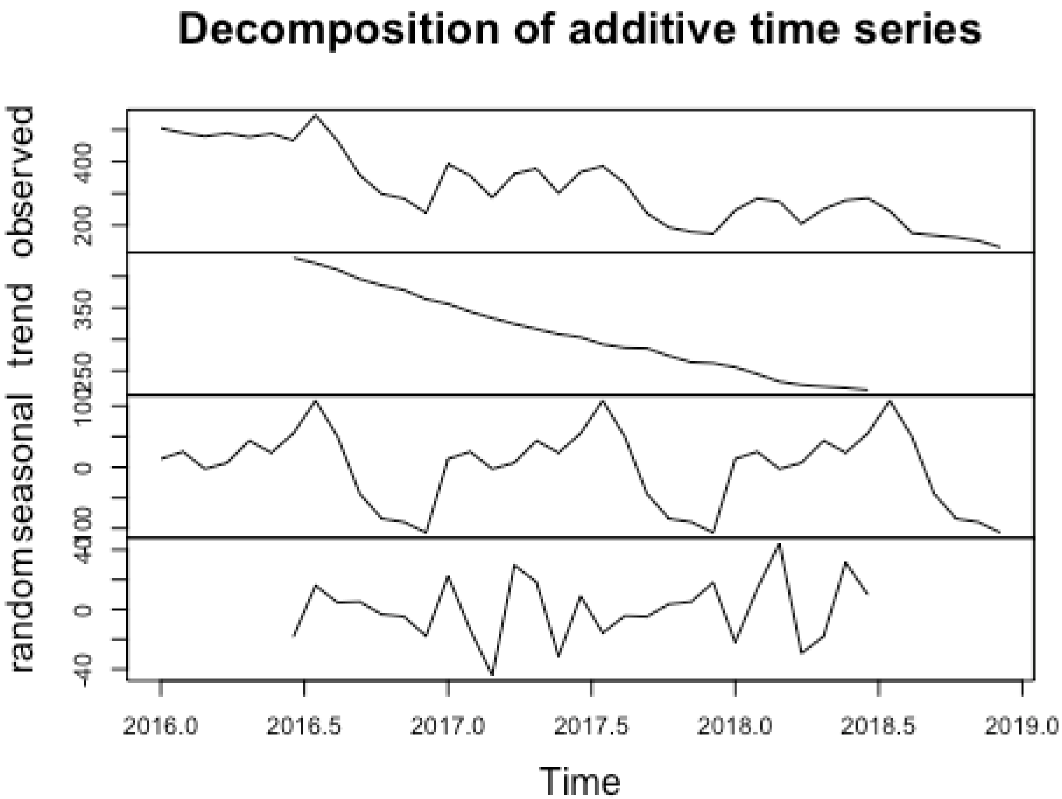
```
## [15,] 355.6523
## [16,] 286.7056
## [17,] 361.4447
## [18,] 378.2739
## [19,] 300.9221
## [20,] 367.5470
## [21,] 385.5379
## [22,] 332.1504
## [23,] 237.7136
## [24,] 193.3008
## [25,] 179.2925
## [26,] 173.2373
## [27,] 247.3155
## [28,] 284.1833
## [29,] 274.4308
## [30,] 205.5000
## [31,] 250.5551
## [32,] 278.3175
## [33,] 284.8955
## [34,] 244.9314
## [35,] 175.4323
## [36,] 168.1067
## [37,] 161.5293
## [38,] 151.6422
## [39,] 130.9374
##
## $seasonal
## Time Series:
## Start = c(2016, 1)
## End = c(2018, 13)
## Frequency = 13
##  [1]   13.549668   25.228097   -2.987152    7.207104   43.419396
##  [6]   23.707591   55.203588  109.065237   49.732480  -43.246914
## [11]  -84.094977  -89.766380 -107.017738   13.549668   25.228097
## [16]   -2.987152    7.207104   43.419396   23.707591   55.203588
## [21]  109.065237   49.732480  -43.246914  -84.094977  -89.766380
## [26] -107.017738   13.549668   25.228097   -2.987152    7.207104
## [31]   43.419396   23.707591   55.203588  109.065237   49.732480
## [36]  -43.246914  -84.094977  -89.766380 -107.017738
##
## $trend
## Time Series:
## Start = c(2016, 1)
## End = c(2018, 13)
## Frequency = 13
##           [,1]
##  [1,]       NA
##  [2,]       NA
##  [3,]       NA
##  [4,]       NA
```

```
##  [5,]       NA
##  [6,]       NA
##  [7,] 429.6008
##  [8,] 420.9501
##  [9,] 410.5982
## [10,] 395.7875
## [11,] 385.9712
## [12,] 378.3744
## [13,] 363.9949
## [14,] 356.3909
## [15,] 344.0436
## [16,] 333.8301
## [17,] 324.6043
## [18,] 316.5080
## [19,] 308.4691
## [20,] 303.3926
## [21,] 292.2380
## [22,] 286.7403
## [23,] 285.7961
## [24,] 273.8004
## [25,] 263.9759
## [26,] 262.2370
## [27,] 255.8792
## [28,] 245.0633
## [29,] 233.0081
## [30,] 227.6537
## [31,] 225.2098
## [32,] 223.0828
## [33,] 219.8290
## [34,]       NA
## [35,]       NA
## [36,]       NA
## [37,]       NA
## [38,]       NA
## [39,]       NA
##
## $random
## Time Series:
## Start = c(2016, 1)
## End = c(2018, 13)
## Frequency = 13
##       x - seasonal
##  [1,]           NA
##  [2,]           NA
##  [3,]           NA
##  [4,]           NA
##  [5,]           NA
##  [6,]           NA
##  [7,]   -18.405062
##  [8,]    16.037745
```

```
##  [9,]     4.594887
## [10,]     5.108068
## [11,]    -3.322913
## [12,]    -4.810591
## [13,]   -17.745561
## [14,]    22.385861
## [15,]   -13.619413
## [16,]   -44.137362
## [17,]    29.633288
## [18,]    18.346540
## [19,]   -31.254669
## [20,]     8.950812
## [21,]   -15.765307
## [22,]    -4.322449
## [23,]    -4.835630
## [24,]     3.595351
## [25,]     5.083030
## [26,]    18.018000
## [27,]   -22.113422
## [28,]    13.891852
## [29,]    44.409801
## [30,]   -29.360849
## [31,]   -18.074102
## [32,]    31.527107
## [33,]     9.862908
## [34,]           NA
## [35,]           NA
## [36,]           NA
## [37,]           NA
## [38,]           NA
## [39,]           NA
## 
## $figure
##  [1]   13.549668   25.228097   -2.987152    7.207104   43.419396
##  [6]   23.707591   55.203588  109.065237   49.732480  -43.246914
## [11]  -84.094977  -89.766380 -107.017738
## 
## $type
## [1] "additive"
## 
## attr(,"class")
## [1] "decomposed.ts"
```

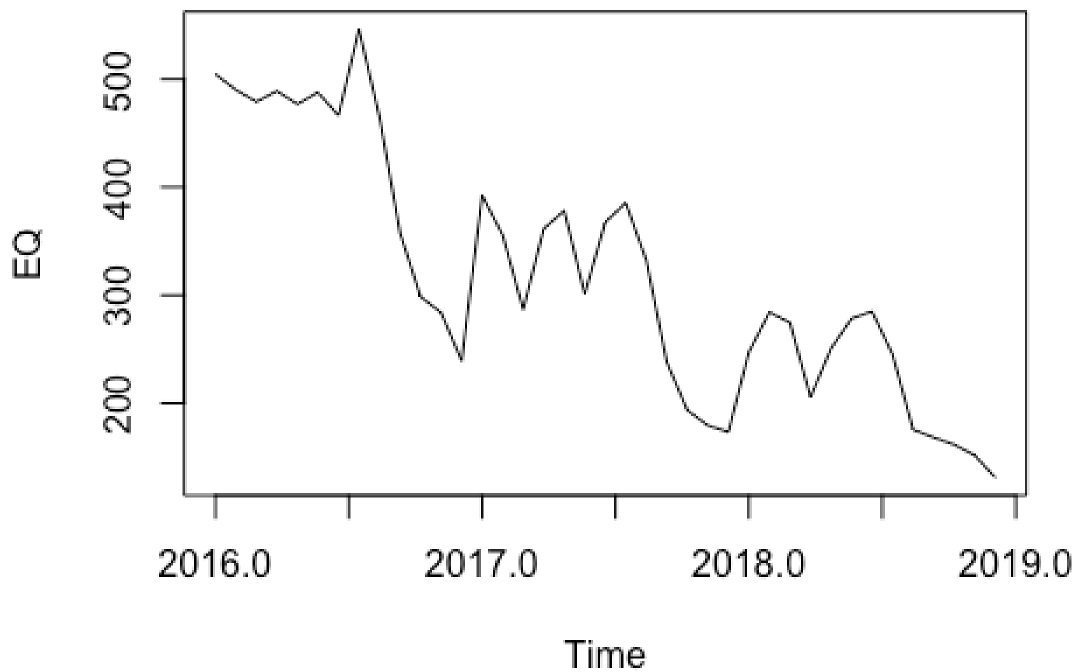
### Plot the above decompose data

```
plot(decompose(sales_data.ts))
```



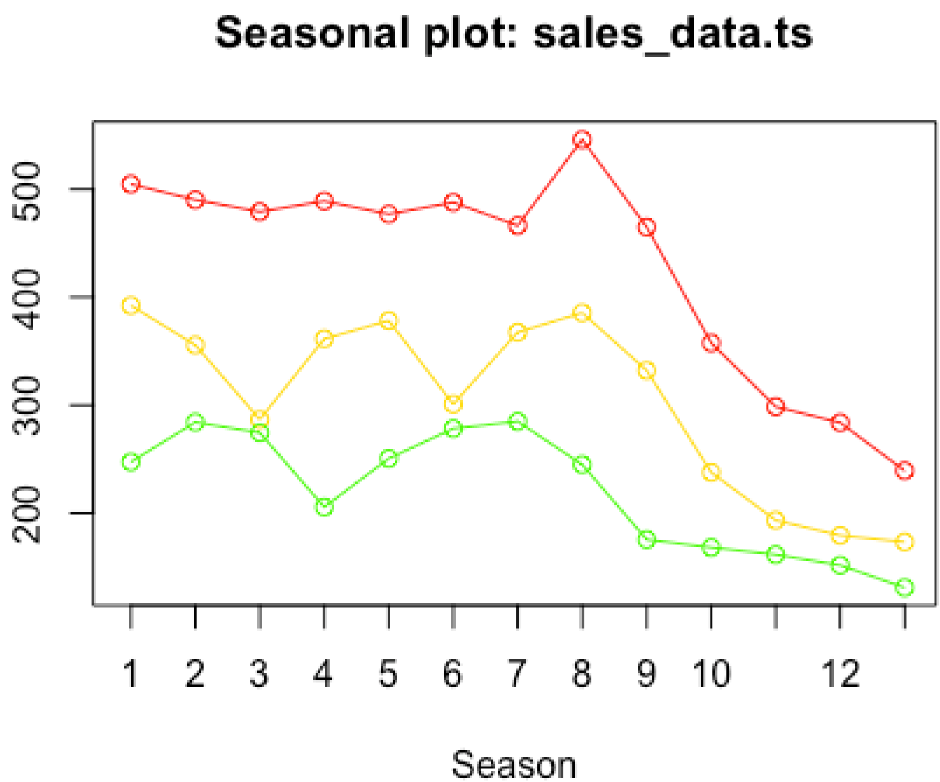

**Plot the actual values of the decompose data**

```
plot(decompose(sales_data.ts)$x)
```



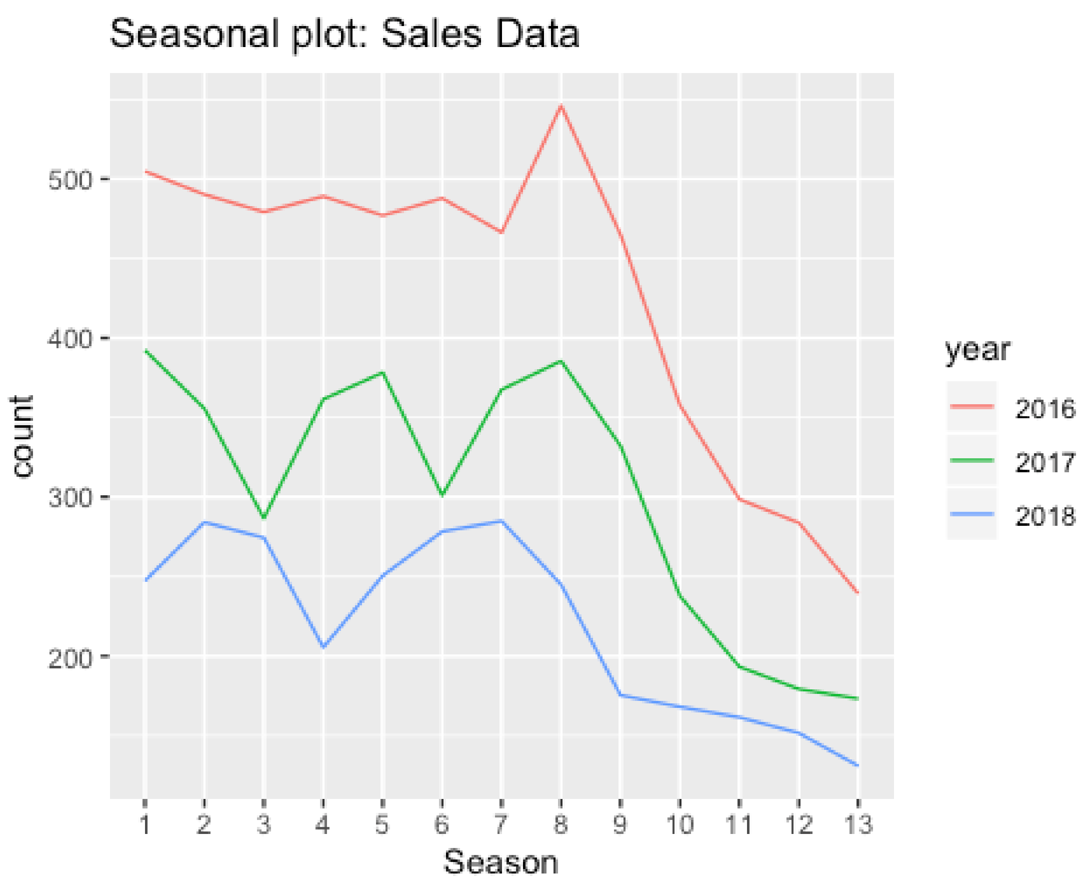

**Plot the time series dataframe based on season**

```
seasonplot(sales_data.ts , col = rainbow(7))
```




**Plot the seasonal sale using ggseasonalplot**

```
ggseasonplot(sales_data.ts , ylab="count" , main="Seasonal plot: Sales Data")
```




**Split the data into train and test set to build multiplicative model**

```
# Split the data for train set

sales_data.ts.train <- window(sales_data.ts , start = c(2016,1) , end =
c(2017,13), frequency = 13)
sales_data.ts.train

## Time Series:
## Start = c(2016, 1)
## End = c(2017, 13)
## Frequency = 13
##             EQ
##  [1,] 504.7849
##  [2,] 490.2265
##  [3,] 479.2447
##  [4,] 489.0574
##  [5,] 477.0320
##  [6,] 487.8553
##  [7,] 466.3993
##  [8,] 546.0531
##  [9,] 464.9256
## [10,] 357.6487
## [11,] 298.5533
```

```
## [12,] 283.7974
## [13,] 239.2316
## [14,] 392.3264
## [15,] 355.6523
## [16,] 286.7056
## [17,] 361.4447
## [18,] 378.2739
## [19,] 300.9221
## [20,] 367.5470
## [21,] 385.5379
## [22,] 332.1504
## [23,] 237.7136
## [24,] 193.3008
## [25,] 179.2925
## [26,] 173.2373
```

### Split the data for test set

```
sales_data.ts.test <- window(sales_data.ts , start = c(2018,1) , frequency =
13)

sales_data.ts.test

## Time Series:
## Start = c(2018, 1)
## End = c(2018, 13)
## Frequency = 13
##             EQ
##  [1,] 247.3155
##  [2,] 284.1833
##  [3,] 274.4308
##  [4,] 205.5000
##  [5,] 250.5551
##  [6,] 278.3175
##  [7,] 284.8955
##  [8,] 244.9314
##  [9,] 175.4323
## [10,] 168.1067
## [11,] 161.5293
## [12,] 151.6422
## [13,] 130.9374
```

### Build the Holt Winter Model ( Multiplicative ) using train set

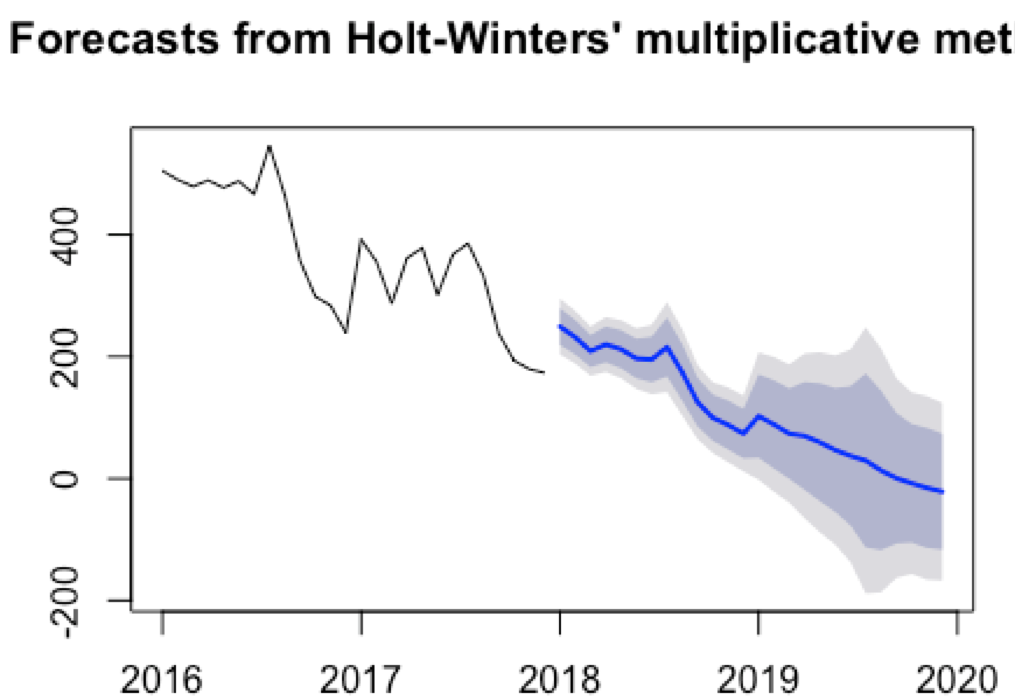
```
hw.model.multi <- hw(sales_data.ts.train , seasonal = "m")

summary(hw.model.multi)

##
## Forecast method: Holt-Winters' multiplicative method
##
## Model Information:
```

```
## Holt-Winters' multiplicative method
##
## Call:
##  hw(y = sales_data.ts.train, seasonal = "m")
##
##   Smoothing parameters:
##     alpha = 0.0816
##     beta  = 0.0804
##     gamma = 1e-04
##
##   Initial states:
##     l = 516.8021
##     b = -11.0241
##     s = 0.6829 0.7412 0.7705 0.8941 1.1531 1.3335
##            1.1323 1.0746 1.0938 1.0746 0.971 1.0251 1.0531
##
##   sigma:  0.0939
##
##      AIC     AICc      BIC
## 274.6586 372.3729 297.3043
##
## Error measures:
##                     ME     RMSE      MAE       MPE   MAPE      MASE
## Training set 0.2601721 18.78671 15.87713 -0.314091 4.6764 0.1258012
##                    ACF1
## Training set -0.2431128
##
## Forecasts:
##          Point Forecast        Lo 80     Hi 80       Lo 95    Hi 95
## 2018.000    249.2295360  219.2263095 279.23276  203.343561 295.1155
## 2018.077    231.6005717  203.3172821 259.88386  188.345013 274.8561
## 2018.154    208.9474649  182.5547858 235.34014  168.583345 249.3116
## 2018.231    219.7258133  190.2393434 249.21228  174.630149 264.8215
## 2018.308    211.9258602  180.7779392 243.07378  164.289226 259.5625
## 2018.385    196.6661937  164.0823765 229.25001  146.833546 246.4988
## 2018.462    195.0868655  157.7942754 232.37946  138.052771 252.1210
## 2018.538    215.4438405  167.1431452 263.74454  141.574302 289.3134
## 2018.615    173.9345859  127.7229561 220.14622  103.259997 244.6092
## 2018.692    125.2727222   85.5787702 164.96667   64.566061 185.9794
## 2018.769     99.6909849   61.8617477 137.52022   41.836159 157.5458
## 2018.846     87.9475332   47.8426067 128.05246   26.612341 149.2827
## 2018.923     73.7077591   33.1583284 114.25719   11.692756 135.7228
## 2019.000    102.3675845   34.0651273 170.67004   -2.092009 206.8272
## 2019.077     88.6469047   16.3633285 160.93048  -21.901285 199.1951
## 2019.154     73.5474118   -0.5548839 147.64971  -39.782270 186.8771
## 2019.231     69.8714789  -18.5194007 158.26236  -65.310706 205.0537
## 2019.308     59.3887103  -37.1981830 155.97560  -88.328196 207.1056
## 2019.385     46.8156945  -54.6714441 148.30283 -108.395490 202.0269
## 2019.462     37.1847876  -76.8089824 151.17856 -137.153639 211.5232
## 2019.538     29.4870588 -113.1704114 172.14453 -188.688714 247.6628
```

```
## 2019.615     13.1291444 -117.5879291 143.84622 -186.785368 213.0437
## 2019.692      0.5888675 -106.5269992 107.70473 -163.230714 164.4084
## 2019.769     -7.7578443 -105.0808120  89.56512 -156.600479 141.0848
## 2019.846    -15.4135809 -113.9023083  83.07515 -166.039091 135.2119
## 2019.923    -21.5276640 -116.8006304  73.74530 -167.235092 124.1798
```

### Plot the above model

```r
plot(hw.model.multi)
```




### Forecast using the train data for next 13 period using above model

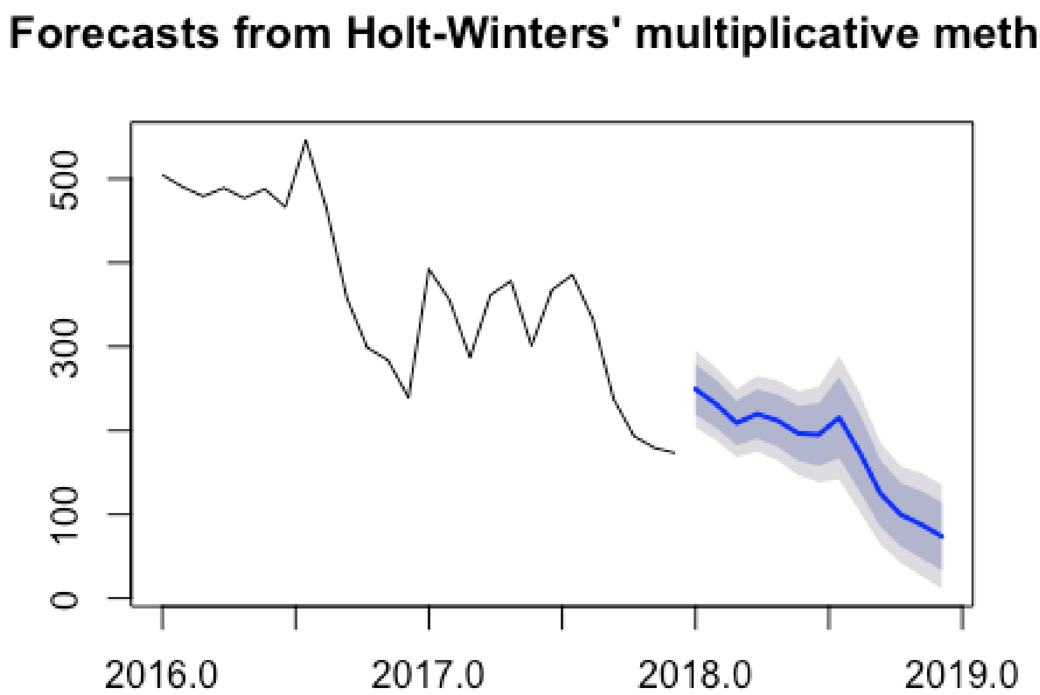
```r
train.forecast.multi <- forecast(hw.model.multi , h=13)
train.forecast.multi
```

```
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## 2018.000      249.22954 219.22631 279.2328 203.34356 295.1155
## 2018.077      231.60057 203.31728 259.8839 188.34501 274.8561
## 2018.154      208.94746 182.55479 235.3401 168.58335 249.3116
## 2018.231      219.72581 190.23934 249.2123 174.63015 264.8215
## 2018.308      211.92586 180.77794 243.0738 164.28923 259.5625
## 2018.385      196.66619 164.08238 229.2500 146.83355 246.4988
## 2018.462      195.08687 157.79428 232.3795 138.05277 252.1210
## 2018.538      215.44384 167.14315 263.7445 141.57430 289.3134
## 2018.615      173.93459 127.72296 220.1462 103.26000 244.6092
```

```
## 2018.692       125.27272  85.57877 164.9667  64.56606 185.9794
## 2018.769        99.69098  61.86175 137.5202  41.83616 157.5458
## 2018.846        87.94753  47.84261 128.0525  26.61234 149.2827
## 2018.923        73.70776  33.15833 114.2572  11.69276 135.7228
```

### Plot the above forecast

```r
plot(train.forecast.multi)
```




### Find the forcasted value for each month.

```r
train.forecast.value <- train.forecast.multi$mean
train.forecast.value
```

```
## Time Series:
## Start = c(2018, 1)
## End = c(2018, 13)
## Frequency = 13
##  [1] 249.22954 231.60057 208.94746 219.72581 211.92586 196.66619 195.08687
##  [8] 215.44384 173.93459 125.27272  99.69098  87.94753  73.70776
```

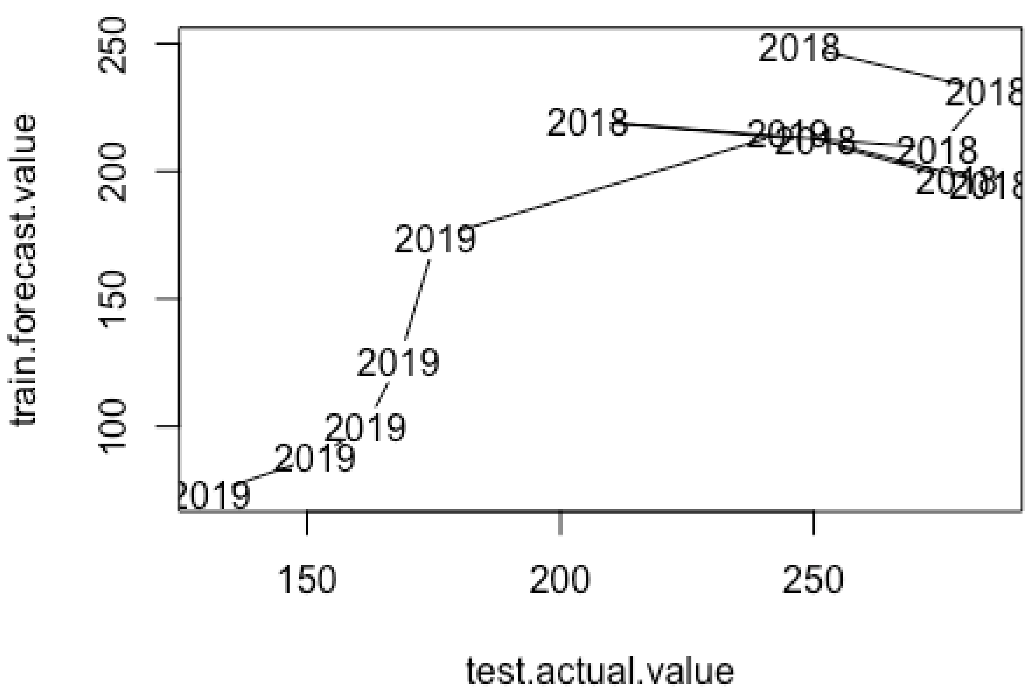
### Actual Test Values

```r
test.actual.value <- sales_data.ts.test
test.actual.value
```

```
## Time Series:
## Start = c(2018, 1)
## End = c(2018, 13)
## Frequency = 13
##               EQ
##  [1,] 247.3155
##  [2,] 284.1833
##  [3,] 274.4308
##  [4,] 205.5000
##  [5,] 250.5551
##  [6,] 278.3175
##  [7,] 284.8955
##  [8,] 244.9314
##  [9,] 175.4323
## [10,] 168.1067
## [11,] 161.5293
## [12,] 151.6422
## [13,] 130.9374
```

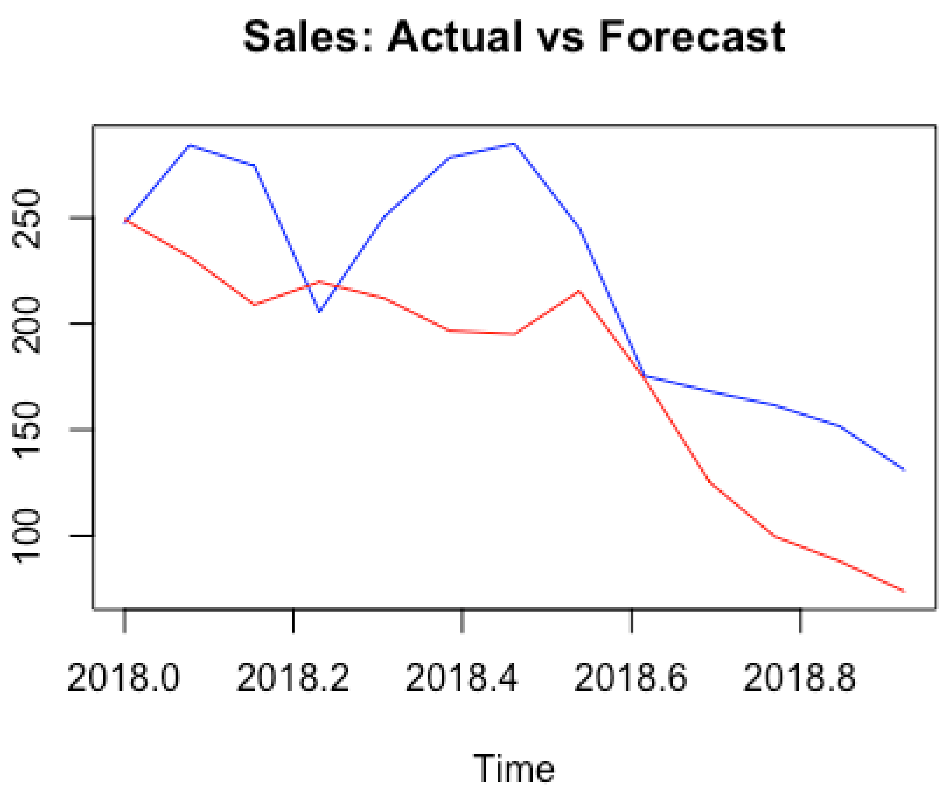
### Plot the actual and forcasted values

```
plot(test.actual.value,train.forecast.value)
```



### Accuracy measures: RMSE and MAPE using HOLT WINTER MODEL (MULTIPLICATIVE)

```
Vec2 <- (cbind(test.actual.value,train.forecast.value))

ts.plot(Vec2, col=c("blue", "red"), main="Sales: Actual vs Forecast")
```




Blue line denotes actual value and red denotes predicted values. Note that predicted values are somewhat lower than the actual observations.

### Find the RMSE and MAPE values

```
RMSE2 <- round(sqrt(sum(((Vec2[,1]-Vec2[,2])^2)/length(Vec2[,1]))),4)
MAPE2 <- round(mean(abs(Vec2[,1]-Vec2[,2])/Vec2[,1]),4) * 100
paste("Accuracy Measures: RMSE:", RMSE2, "and MAPE:", MAPE2)

## [1] "Accuracy Measures: RMSE: 53.5986 and MAPE: 22.21"
```

### Build the Holt Winter Model ( Multiplicative ) on the full validation data

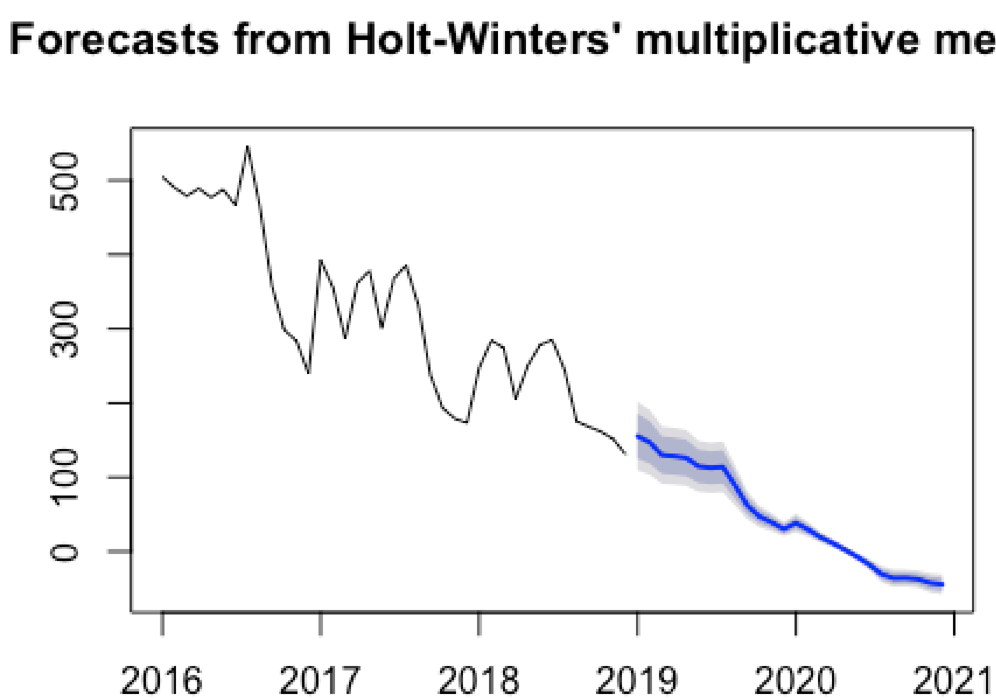
```
hw.model.multi_full <- hw(sales_data.ts , seasonal = "m")

summary(hw.model.multi_full)

##
## Forecast method: Holt-Winters' multiplicative method
##
```

```
## Model Information:
## Holt-Winters' multiplicative method
##
## Call:
##  hw(y = sales_data.ts, seasonal = "m")
##
##   Smoothing parameters:
##     alpha = 0.0397
##     beta  = 1e-04
##     gamma = 1e-04
##
##   Initial states:
##     l = 490.6311
##     b = -8.5534
##     s = 0.6701 0.7356 0.7588 0.8798 1.1232 1.2982
##            1.1704 1.0943 1.1114 1.0529 0.9953 1.0575 1.0524
##
##   sigma:  0.1515
##
##      AIC     AICc      BIC
## 452.5155 486.7155 482.4596
##
## Error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.8630968 24.98351 18.54407 0.2291539 7.433924 0.1768023
##                   ACF1
## Training set 0.1825626
##
## Forecasts:
##          Point Forecast       Lo 80      Hi 80      Lo 95      Hi 95
## 2019.000     155.425428 125.2514613 185.599395 109.278328 201.572529
## 2019.077     147.127785 118.5386284 175.716941 103.404443 190.851126
## 2019.154     129.954647 104.6760603 155.233235  91.294385 168.614910
## 2019.231     128.475197 103.4539282 153.496465  90.208469 166.741925
## 2019.308     126.105589 101.5103382 150.700839  88.490399 163.720778
## 2019.385     114.803697  92.3736450 137.233749  80.499893 149.107501
## 2019.462     112.776565  90.6950415 134.858089  79.005789 146.547342
## 2019.538     113.979637  91.6015516 136.357723  79.755309 148.203966
## 2019.615      89.009367  71.4714741 106.547261  62.187474 115.831261
## 2019.692      62.194157  49.8807797  74.507534  43.362472  81.025842
## 2019.769      47.150143  37.7520220  56.548264  32.776957  61.523329
## 2019.846      39.417600  31.4821120  47.353088  27.281318  51.553882
## 2019.923      30.173049  24.0028942  36.343203  20.736612  39.609485
## 2020.000      38.386467  30.3247642  46.448169  26.057157  50.715777
## 2020.077      29.525016  23.0028389  36.047192  19.550207  39.499824
## 2020.154      19.273378  14.5123899  24.034366  11.992075  26.554681
## 2020.231      11.382676   7.5543543  15.210998   5.527763  17.237589
## 2020.308       2.507541  -0.8373028   5.852384  -2.607956   7.623037
## 2020.385      -6.892279 -10.4220352  -3.362523 -12.290575  -1.493983
## 2020.462     -17.384030 -22.2479161 -12.520144 -24.822702  -9.945358
```

```
## 2020.538      -30.386357 -37.4565353 -23.316180 -41.199261 -19.573453
## 2020.615      -35.899317 -43.6478592 -28.150775 -47.749690 -24.048945
## 2020.692      -35.645792 -43.0624814 -28.229102 -46.988640 -24.302943
## 2020.769      -37.235001 -44.8266370 -29.643365 -48.845406 -25.624596
## 2020.846      -42.391143 -50.9285689 -33.853718 -55.448009 -29.334278
## 2020.923      -44.346123 -53.2068253 -35.485421 -57.897398 -30.794849
```

### Plot the above model

```
plot(hw.model.multi_full)
```




### Forecast using the train data for next 6 period using above model

```
train.forecast.multi_full <- forecast(hw.model.multi_full , h=13)
train.forecast.multi_full

##          Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 2019.000      155.42543 125.25146 185.59940 109.27833 201.57253
## 2019.077      147.12778 118.53863 175.71694 103.40444 190.85113
## 2019.154      129.95465 104.67606 155.23323  91.29438 168.61491
## 2019.231      128.47520 103.45393 153.49647  90.20847 166.74192
## 2019.308      126.10559 101.51034 150.70084  88.49040 163.72078
## 2019.385      114.80370  92.37364 137.23375  80.49989 149.10750
## 2019.462      112.77657  90.69504 134.85809  79.00579 146.54734
## 2019.538      113.97964  91.60155 136.35772  79.75531 148.20397
```

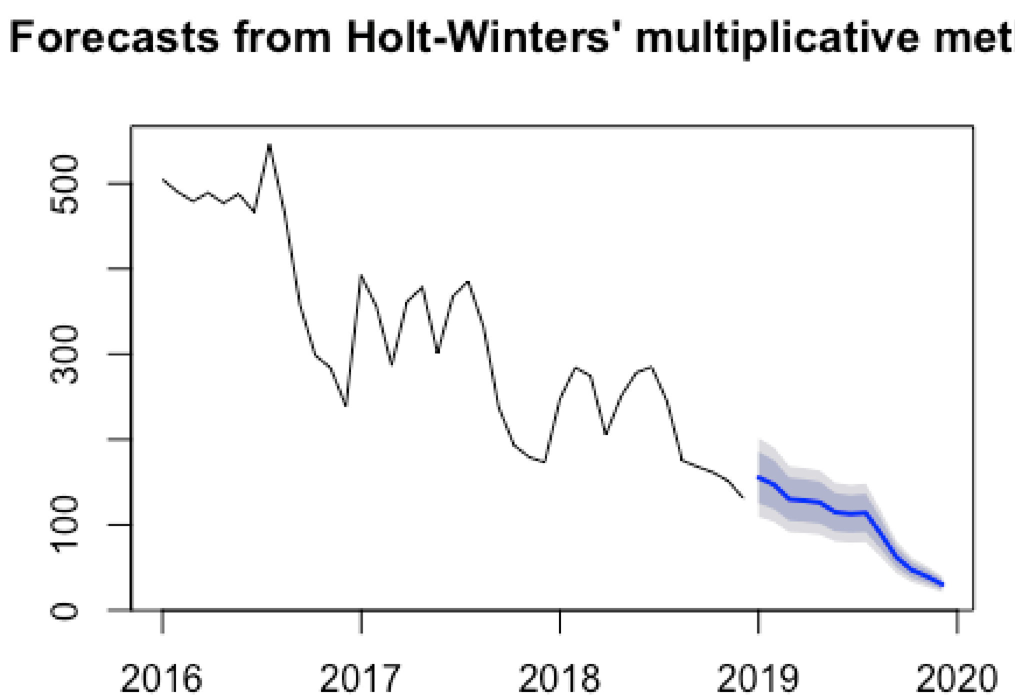
```
## 2019.615       89.00937  71.47147 106.54726  62.18747 115.83126
## 2019.692       62.19416  49.88078  74.50753  43.36247  81.02584
## 2019.769       47.15014  37.75202  56.54826  32.77696  61.52333
## 2019.846       39.41760  31.48211  47.35309  27.28132  51.55388
## 2019.923       30.17305  24.00289  36.34320  20.73661  39.60949
```

```r
summary(train.forecast.multi_full)
```

```
##
## Forecast method: Holt-Winters' multiplicative method
##
## Model Information:
## Holt-Winters' multiplicative method
##
## Call:
##  hw(y = sales_data.ts, seasonal = "m")
##
##   Smoothing parameters:
##     alpha = 0.0397
##     beta  = 1e-04
##     gamma = 1e-04
##
##   Initial states:
##     l = 490.6311
##     b = -8.5534
##     s = 0.6701 0.7356 0.7588 0.8798 1.1232 1.2982
##            1.1704 1.0943 1.1114 1.0529 0.9953 1.0575 1.0524
##
##   sigma:  0.1515
##
##      AIC     AICc      BIC
## 452.5155 486.7155 482.4596
##
## Error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -0.8630968 24.98351 18.54407 0.2291539 7.433924 0.1768023
##                   ACF1
## Training set 0.1825626
##
## Forecasts:
##          Point Forecast     Lo 80     Hi 80     Lo 95     Hi 95
## 2019.000      155.42543 125.25146 185.59940 109.27833 201.57253
## 2019.077      147.12778 118.53863 175.71694 103.40444 190.85113
## 2019.154      129.95465 104.67606 155.23323  91.29438 168.61491
## 2019.231      128.47520 103.45393 153.49647  90.20847 166.74192
## 2019.308      126.10559 101.51034 150.70084  88.49040 163.72078
## 2019.385      114.80370  92.37364 137.23375  80.49989 149.10750
## 2019.462      112.77657  90.69504 134.85809  79.00579 146.54734
## 2019.538      113.97964  91.60155 136.35772  79.75531 148.20397
## 2019.615       89.00937  71.47147 106.54726  62.18747 115.83126
```

```
## 2019.692       62.19416  49.88078  74.50753  43.36247  81.02584
## 2019.769       47.15014  37.75202  56.54826  32.77696  61.52333
## 2019.846       39.41760  31.48211  47.35309  27.28132  51.55388
## 2019.923       30.17305  24.00289  36.34320  20.73661  39.60949
```

```
plot(train.forecast.multi_full)
```




## GET THE ACTUAL SALES VALUES FOR NEXT 6 PERIOD UDING HOLT WINTER MODEL

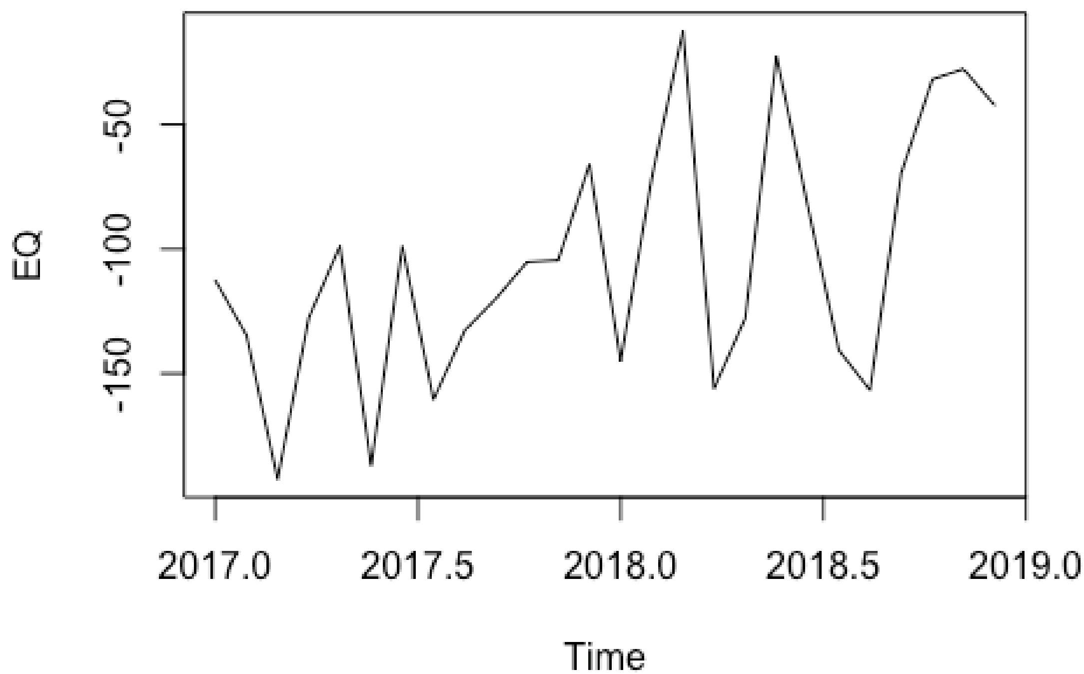
```
forecast(train.forecast.multi_full, h=6)$mean

## Time Series:
## Start = c(2019, 1)
## End = c(2019, 6)
## Frequency = 13
## [1] 155.4254 147.1278 129.9546 128.4752 126.1056 114.8037
```

### CHECK FOR STATIONARY

```
library(tseries)
diff1 <- diff(sales_data.ts , lag = 13)

adf.test(diff1)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff1
## Dickey-Fuller = -2.9168, Lag order = 2, p-value = 0.2231
## alternative hypothesis: stationary
```

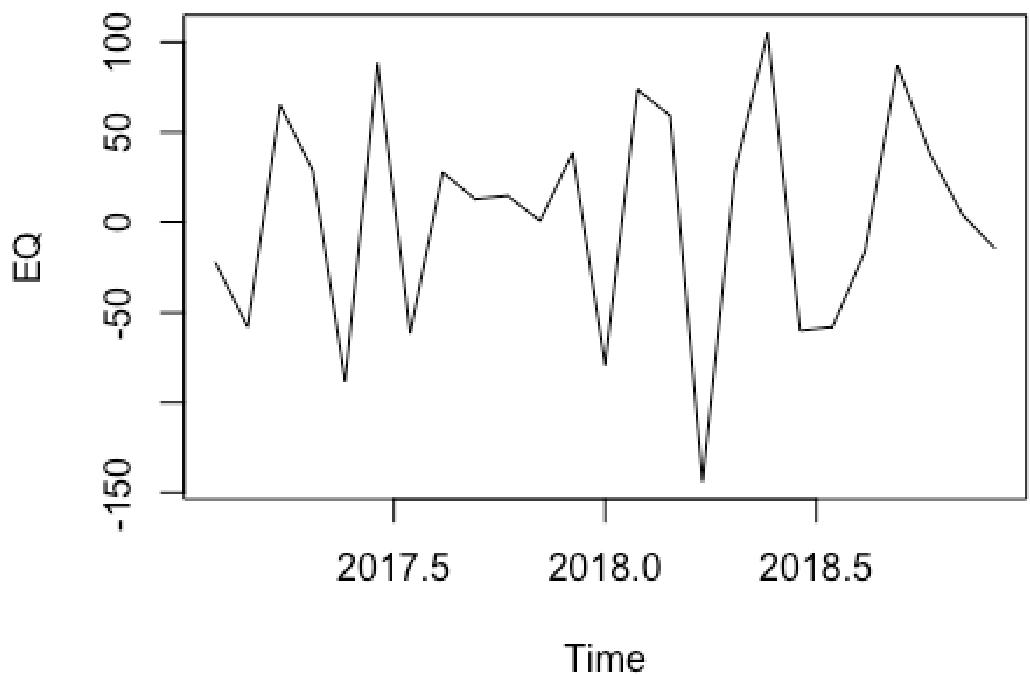
```
plot.ts(diff1)
```




### Lets do the adf test on difference in sales

```
library(tseries)
diff2 <- diff(diff1 , lag = 1)

adf.test(diff2)

## Warning in adf.test(diff2): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  diff2
## Dickey-Fuller = -4.7677, Lag order = 2, p-value = 0.01
## alternative hypothesis: stationary
```

```r
plot.ts(diff2)
```




p = 0.01

, Reject null hythesis - So series is stationary We also see the plot has become stationary.

### Build the ARIMA Model on the full validation dataset using auto.arima function

```r
auto.arima.model <- auto.arima(sales_data.ts)

## Warning: 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.

auto.arima.model

## Series: sales_data.ts
## ARIMA(0,1,1)(0,1,0)[13]
##
## Coefficients:
##           ma1
##       -0.7810
## s.e.   0.1197
##
```

```
## sigma^2 estimated as 2490:  log likelihood=-133.18
## AIC=270.36   AICc=270.91   BIC=272.8
```

```
summary(auto.arima.model)
```

```
## Series: sales_data.ts
## ARIMA(0,1,1)(0,1,0)[13]
##
## Coefficients:
##           ma1
##       -0.7810
## s.e.   0.1197
##
## sigma^2 estimated as 2490:  log likelihood=-133.18
## AIC=270.36   AICc=270.91   BIC=272.8
##
## Training set error measures:
##                    ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 7.101845 39.14123 27.80842 3.879737 12.33453 0.2651302
##                     ACF1
## Training set -0.01529808
```

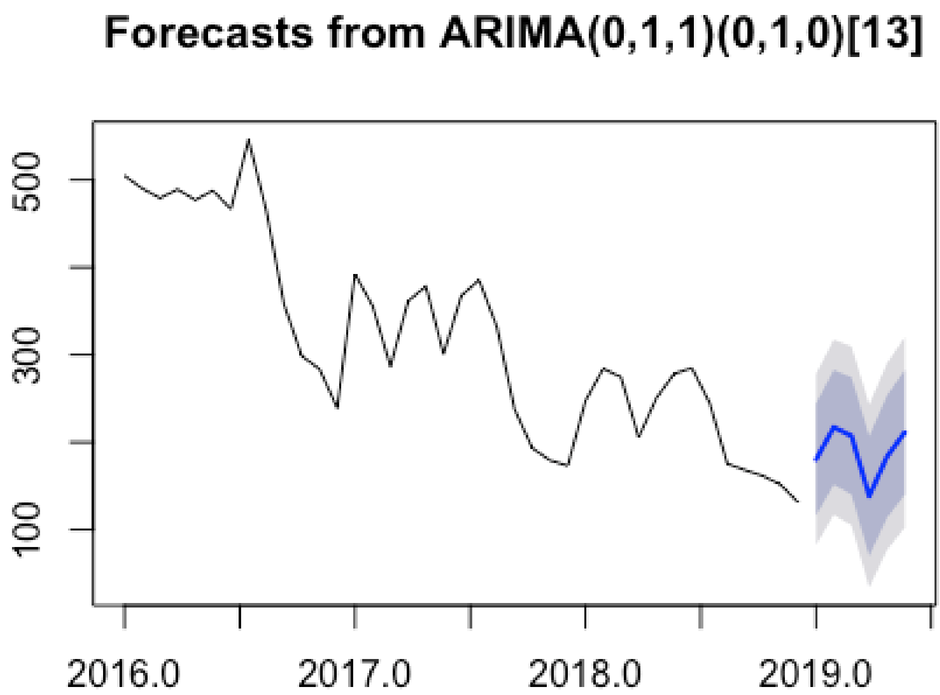
### Forecast for the next 6 period using ARIMA MODEL

```
arima.forecast <- forecast(auto.arima.model , h=6)
arima.forecast
```

```
##          Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## 2019.000       180.2191 116.27544 244.1628  82.42571 278.0125
## 2019.077       217.0869 151.62822 282.5456 116.97648 317.1974
## 2019.154       207.3344 140.39493 274.2738 104.95932 309.7094
## 2019.231       138.4036  70.01546 206.7918  33.81295 242.9943
## 2019.308       183.4587 113.65189 253.2655  76.69841 290.2190
## 2019.385       211.2212 140.02397 282.4183 102.33446 320.1078
```

#### Plot the next Six period forecast

```
plot(arima.forecast)
```




### GET THE ACTUAL SALES VALUES FOR NEXT 6 PERIOD USING ARIMA MODEL

```
forecast(arima.forecast, h=6)$mean

## Time Series:
## Start = c(2019, 1)
## End = c(2019, 6)
## Frequency = 13
## [1] 180.2191 217.0869 207.3344 138.4036 183.4587 211.2212
```

### Read the validation dataset

```
library(readxl)
sales_data_ts <- read_excel("/Users/dinesh/Downloads/Test dataset v1.xlsx")

head(sales_data_ts)

## # A tibble: 6 x 39
##   Period    EQ Social_Search_I… Social_Search_W… Digital_Impress…
##   <chr>  <dbl>            <dbl>            <dbl>            <dbl>
## 1 2016 …  505.          2019283             5493           37148.
## 2 2016 …  490.          4564738            12938           50887.
## 3 2016 …  479.          1029384             6546          253333.
## 4 2016 …  489.           902938             3928         3426239
## 5 2016 …  477.          1343454            28374          552198.
```

```
## 6 2016 …  488.          2434564           59483           29892.
## # … with 34 more variables: Digital_Working_cost <dbl>,
## #   Print_Impressions.Ads40 <dbl>, Print_Working_Cost.Ads50 <dbl>,
## #   OOH_Impressions <dbl>, OOH_Working_Cost <dbl>, SOS_pct <dbl>,
## #   Digital_Impressions_pct <dbl>, CCFOT <dbl>, Median_Temp <dbl>,
## #   Median_Rainfall <dbl>, Fuel_Price <dbl>, Inflation <dbl>,
## #   Trade_Invest <dbl>, Brand_Equity <dbl>, Avg_EQ_Price <dbl>,
## #   Any_Promo_pct_ACV <dbl>, Any_Feat_pct_ACV <dbl>,
## #   Any_Disp_pct_ACV <dbl>, EQ_Base_Price <dbl>, Est_ACV_Selling <dbl>,
## #   pct_ACV <dbl>, Avg_no_of_Items <dbl>,
## #   pct_PromoMarketDollars_Category <dbl>, RPI_Category <dbl>,
## #   Magazine_Impressions_pct <dbl>, TV_GRP <dbl>, Competitor1_RPI <dbl>,
## #   Competitor2_RPI <dbl>, Competitor3_RPI <dbl>, Competitor4_RPI <dbl>,
## #   EQ_Category <dbl>, EQ_Subcategory <dbl>,
## #   pct_PromoMarketDollars_Subcategory <dbl>, RPI_Subcategory <dbl>
```

### Build the dataframe with all the significant predictor variabled and sales(EQ) and Period

```
sales_data_signif <- cbind.data.frame(Period=sales_data_ts$Period,
                                      EQ=sales_data_ts$EQ,

Medium_rainfall=sales_data_ts$Median_Rainfall,

Social_Search_Impressions=sales_data_ts$Social_Search_Impressions,

pct_PromoMarketDollars_Category=sales_data_ts$pct_PromoMarketDollars_Category
,
                                      Inflation=sales_data_ts$Inflation,

EQ_Category=sales_data_ts$EQ_Category,

pct_PromoMarketDollars_Subcategory=sales_data_ts$pct_PromoMarketDollars_Subca
tegory,

EQ_Subcategory=sales_data_ts$EQ_Subcategory,

Digital_Impressions=sales_data_ts$Digital_Impressions,

Est_ACV_Selling=sales_data_ts$Est_ACV_Selling,stringsAsFactors = FALSE)

head(sales_data_signif)

##             Period       EQ Medium_rainfall Social_Search_Impressions
## 1 2016 - Period:1 504.7849          0.5150                   2019283
## 2 2016 - Period:2 490.2265          0.2700                   4564738
## 3 2016 - Period:3 479.2447          0.3900                   1029384
## 4 2016 - Period:4 489.0574          0.3500                    902938
## 5 2016 - Period:5 477.0320          0.5025                   1343454
## 6 2016 - Period:6 487.8553          0.4600                   2434564
##   pct_PromoMarketDollars_Category   Inflation EQ_Category
## 1                          0.0339 0.013258065     1728389
```

```
## 2                              0.0391 0.009938487     1900860
## 3                              0.0228 0.007832481     2036437
## 4                              0.0147 0.010034301     2113635
## 5                              0.0219 0.009546344     2402211
## 6                              0.0107 0.009290323     2796950
##   pct_PromoMarketDollars_Subcategory EQ_Subcategory Digital_Impressions
## 1                         0.16273152       331927.5             37148.2
## 2                         0.23164966       334611.4             50886.8
## 3                         0.12539376       387148.4            253333.2
## 4                         0.05660340       482489.7           3426239.0
## 5                         0.06505878       629826.6            552197.8
## 6                         0.02991243       806075.8             29892.2
##   Est_ACV_Selling
## 1      8696587915
## 2      8682307085
## 3      8706897549
## 4      8660288592
## 5      8644518558
## 6      8627353001
```

```r
str(sales_data_signif)
```

```
## 'data.frame':    39 obs. of  11 variables:
##  $ Period                            : chr  "2016 - Period:1" "2016 - 
Period:2" "2016 - Period:3" "2016 - Period:4" ...
##  $ EQ                                : num  505 490 479 489 477 ...
##  $ Medium_rainfall                   : num  0.515 0.27 0.39 0.35 0.502 ...
##  $ Social_Search_Impressions         : num  2019283 4564738 1029384 902938 
1343454 ...
##  $ pct_PromoMarketDollars_Category   : num  0.0339 0.0391 0.0228 0.0147 
0.0219 0.0107 0.00765 0.0302 0.0304 0.00561 ...
##  $ Inflation                         : num  0.01326 0.00994 0.00783 
0.01003 0.00955 ...
##  $ EQ_Category                       : num  1728389 1900860 2036437 
2113635 2402211 ...
##  $ pct_PromoMarketDollars_Subcategory: num  0.1627 0.2316 0.1254 0.0566 
0.0651 ...
##  $ EQ_Subcategory                    : num  331928 334611 387148 482490 
629827 ...
##  $ Digital_Impressions               : num  37148 50887 253333 3426239 
552198 ...
##  $ Est_ACV_Selling                   : num  8.70e+09 8.68e+09 8.71e+09 
8.66e+09 8.64e+09 ...
```

### Build LM model with time series

```r
#install.packages("fpp2")

library(fpp2)
```

```
## Loading required package: fma
```

```
## Loading required package: expsmooth
```

```r
names(sales_data_signif)
```

```
##  [1] "Period"
##  [2] "EQ"
##  [3] "Medium_rainfall"
##  [4] "Social_Search_Impressions"
##  [5] "pct_PromoMarketDollars_Category"
##  [6] "Inflation"
##  [7] "EQ_Category"
##  [8] "pct_PromoMarketDollars_Subcategory"
##  [9] "EQ_Subcategory"
## [10] "Digital_Impressions"
## [11] "Est_ACV_Selling"
```

**Build the time series data using the above variables**

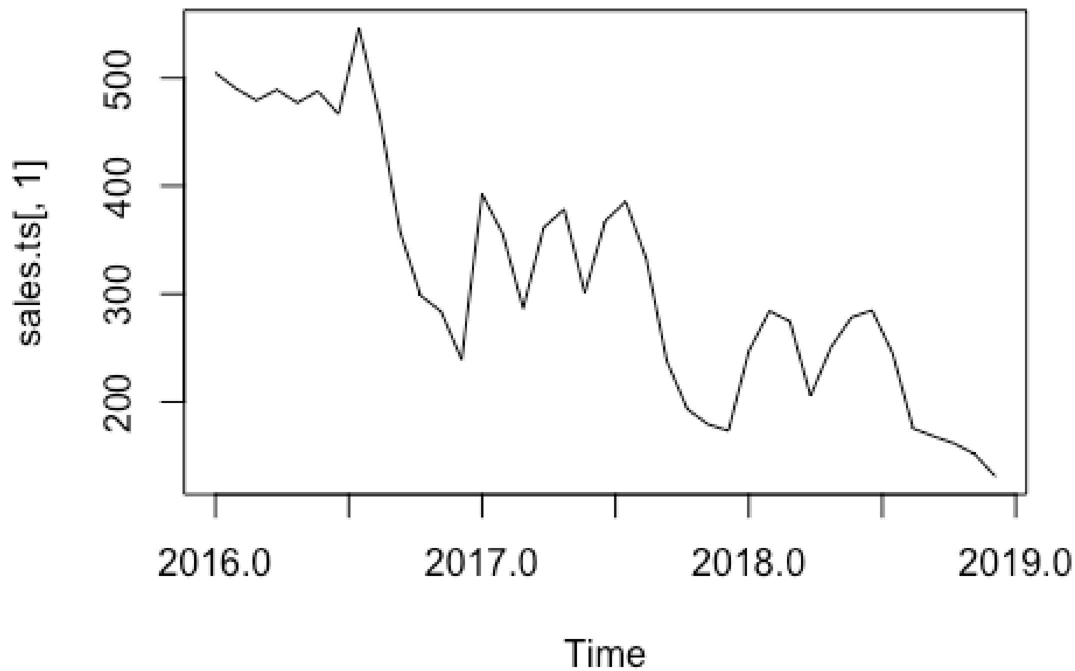
```r
sales.ts <- ts(sales_data_signif[,c(2:11)],start = c(2016,1), end =
c(2018,13) , frequency = 13)
```

```r
head(sales.ts)
```

```
## Time Series:
## Start = c(2016, 1)
## End = c(2016, 6)
## Frequency = 13
##                EQ Medium_rainfall Social_Search_Impressions
## 2016.000 504.7849          0.5150                   2019283
## 2016.077 490.2265          0.2700                   4564738
## 2016.154 479.2447          0.3900                   1029384
## 2016.231 489.0574          0.3500                    902938
## 2016.308 477.0320          0.5025                   1343454
## 2016.385 487.8553          0.4600                   2434564
##          pct_PromoMarketDollars_Category   Inflation EQ_Category
## 2016.000                          0.0339 0.013258065     1728389
## 2016.077                          0.0391 0.009938487     1900860
## 2016.154                          0.0228 0.007832481     2036437
## 2016.231                          0.0147 0.010034301     2113635
## 2016.308                          0.0219 0.009546344     2402211
## 2016.385                          0.0107 0.009290323     2796950
##          pct_PromoMarketDollars_Subcategory EQ_Subcategory
## 2016.000                         0.16273152       331927.5
## 2016.077                         0.23164966       334611.4
## 2016.154                         0.12539376       387148.4
## 2016.231                         0.05660340       482489.7
## 2016.308                         0.06505878       629826.6
## 2016.385                         0.02991243       806075.8
##          Digital_Impressions Est_ACV_Selling
## 2016.000             37148.2      8696587915
## 2016.077             50886.8      8682307085
## 2016.154            253333.2      8706897549
```

```
## 2016.231           3426239.0      8660288592
## 2016.308            552197.8      8644518558
## 2016.385             29892.2      8627353001
```

```r
plot(sales.ts[,1])
```




```r
auto.arima(sales.ts[,1],xreg = sales.ts[,c(2,3)])
```

```
## Series: sales.ts[, 1]
## Regression with ARIMA(0,1,1)(1,0,1)[13] errors
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##           ma1    sar1     sma1  Medium_rainfall  Social_Search_Impressions
##       -0.3674  0.8185  -0.4066          80.0676                          0
## s.e.   0.2443  0.3904   0.6852          15.2082                        NaN
##
## sigma^2 estimated as 2149:  log likelihood=-200.2
## AIC=412.39   AICc=415.1   BIC=422.22
```

```r
summary(auto.arima(sales.ts[,1],xreg = sales.ts[,c(2,3)]))
```

```
## Series: sales.ts[, 1]
## Regression with ARIMA(0,1,1)(1,0,1)[13] errors
##
## Coefficients:

## Warning in sqrt(diag(x$var.coef)): NaNs produced

##           ma1    sar1     sma1  Medium_rainfall  Social_Search_Impressions
##       -0.3674  0.8185  -0.4066          80.0676                          0
## s.e.   0.2443  0.3904   0.6852          15.2082                        NaN
##
## sigma^2 estimated as 2149:  log likelihood=-200.2
## AIC=412.39   AICc=415.1   BIC=422.22
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -5.546375 42.64059 33.28969 -2.501008 12.61145 0.3173896
##                    ACF1
## Training set 0.05069036
```

```
summary(auto.arima(sales.ts[,1],xreg = sales.ts[,c(2:4)]))
```

```
## Series: sales.ts[, 1]
## Regression with ARIMA(0,1,0)(1,0,0)[13] errors
##
## Coefficients:

## Warning in sqrt(diag(x$var.coef)): NaNs produced

##         sar1  Medium_rainfall  Social_Search_Impressions
##       0.2395          43.9398                          0
## s.e.  0.1854          13.2796                        NaN
##       pct_PromoMarketDollars_Category
##                             1979.4117
## s.e.                         257.8143
##
## sigma^2 estimated as 1055:  log likelihood=-184.46
## AIC=378.91   AICc=380.79   BIC=387.1
##
## Training set error measures:
##                     ME     RMSE     MAE       MPE    MAPE      MASE
## Training set -6.748195 30.32761 24.1669 -3.030569 8.39654 0.2304114
##                     ACF1
## Training set -0.07558858
```

```
summary(auto.arima(sales.ts[,1],xreg = sales.ts[,c(2:6)]))
```

```
## Series: sales.ts[, 1]
## Regression with ARIMA(0,1,0)(1,0,0)[13] errors
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##         sar1  Medium_rainfall  Social_Search_Impressions
##       0.1593          26.0389                          0
## s.e.  0.2168           3.2724                        NaN
##       pct_PromoMarketDollars_Category  Inflation  EQ_Category
##                              1909.181  1636.8246        1e-04
## s.e.                          230.672   125.3178          NaN
##
## sigma^2 estimated as 903.3:  log likelihood=-180.14
## AIC=374.27   AICc=378.01   BIC=385.74
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -7.243228 27.22444 22.31574 -2.591191 7.839342 0.2127621
##                    ACF1
## Training set -0.2836857
```

```
summary(auto.arima(sales.ts[,1],xreg = sales.ts[,c(2:10)]))
```

```
## Series: sales.ts[, 1]
## Regression with ARIMA(0,0,0) errors
##
## Coefficients:
##       Medium_rainfall  Social_Search_Impressions
##               79.8147                     0e+00
## s.e.          33.0758                     2e-04
##       pct_PromoMarketDollars_Category  Inflation  EQ_Category
##                             3000.1046  -3889.599        0e+00
## s.e.                         874.1514   1243.941        2e-04
##       pct_PromoMarketDollars_Subcategory  EQ_Subcategory
##                                -253.7832           1e-04
## s.e.                            163.7703           2e-04
##       Digital_Impressions  Est_ACV_Selling
##                     0e+00            0e+00
## s.e.                2e-04            2e-04
##
## sigma^2 estimated as 846.7:  log likelihood=-181.68
## AIC=383.36   AICc=391.22   BIC=399.99
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE     MAPE     MASE
## Training set 0.3421923 25.52118 19.40442 0.002139539 6.881549 0.185005
##                   ACF1
## Training set 0.2502674
```

```
summary(hw(sales.ts[,1]))
```

```
##
## Forecast method: Holt-Winters' additive method
##
```

```
## Model Information:
## Holt-Winters' additive method
##
## Call:
##  hw(y = sales.ts[, 1])
##
##   Smoothing parameters:
##     alpha = 0.0326
##     beta  = 0.0326
##     gamma = 0.0438
##
##   Initial states:
##     l = 504.2814
##     b = -12.681
##     s = -90.9187 -80.9648 -89.1257 -35.5467 50.8659 105.3759
##            60.258 18.2822 46.1876 -0.2688 3.5841 16.9871 -4.7164
##
##   sigma:  43.5088
##
##      AIC     AICc      BIC
## 450.8419 485.0419 480.7860
##
## Error measures:
##                    ME     RMSE      MAE      MPE     MAPE      MASE
## Training set 8.60991 32.67808 26.40321 3.540329 10.76586 0.2517327
##                    ACF1
## Training set 0.1482281
##
## Forecasts:
##          Point Forecast       Lo 80    Hi 80        Lo 95    Hi 95
## 2019.000      178.47744  122.718623 234.2363   93.2016878 263.7532
## 2019.077      196.28802  140.410779 252.1653  110.8311524 281.7449
## 2019.154      180.17842  124.035635 236.3212   94.3154408 266.0414
## 2019.231      176.69727  120.085502 233.3090   90.1170439 263.2775
## 2019.308      218.76020  161.423341 276.0971  131.0710416 306.4494
## 2019.385      192.11412  133.748962 250.4793  102.8523114 281.3759
## 2019.462      230.19117  170.454815 289.9275  138.8323012 321.5500
## 2019.538      271.03586  209.554612 332.5171  177.0084069 365.0633
## 2019.615      214.21288  150.591838 277.8339  116.9128926 311.5129
## 2019.692      129.55974   63.392361 195.7271   28.3654670 230.7540
## 2019.769       77.52462    8.401481 146.6477  -28.1900965 183.2393
## 2019.846       82.12909    9.645270 154.6129  -28.7253454 192.9835
## 2019.923       69.54119   -6.698034 145.7804  -47.0566436 186.1390
## 2020.000      155.94375   74.761867 237.1256   31.7867757 280.1007
## 2020.077      173.75433   88.115076 259.3936   42.7803939 304.7283
## 2020.154      157.64472   67.200738 248.0887   19.3225840 295.9669
## 2020.231      154.16357   58.585304 249.7418    7.9892254 300.3379
## 2020.308      196.22651   95.201937 297.2511   41.7227610 350.7302
## 2020.385      169.58043   62.814336 276.3465    6.2957809 332.8651
## 2020.462      207.65747   94.870415 320.4445   35.1645536 380.1504
```

```
## 2020.538      248.50216  129.429336 367.5750   66.3959898 430.6083
## 2020.615      191.67919   66.069230 317.2891   -0.4246639 383.7830
## 2020.692      107.02605  -25.360129 239.4122  -95.4411370 309.4932
## 2020.769       54.99092  -84.399402 194.3812 -158.1881815 268.1700
## 2020.846       59.59539  -87.016880 206.2077 -164.6287304 283.8195
## 2020.923       47.00750 -107.035350 201.0503 -188.5807089 282.5957
```

```
summary(tslm(EQ ~ Medium_rainfall+Social_Search_Impressions, data =
sales.ts))
```

```
##
## Call:
## tslm(formula = EQ ~ Medium_rainfall + Social_Search_Impressions,
##     data = sales.ts)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -190.945  -66.874    1.931   79.227  186.945
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.522e+02  4.679e+01   5.390 4.54e-06 ***
## Medium_rainfall           2.484e+02  1.061e+02   2.340   0.0249 *
## Social_Search_Impressions -3.507e-06  1.837e-06  -1.909   0.0642 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 108.8 on 36 degrees of freedom
## Multiple R-squared:  0.1884, Adjusted R-squared:  0.1433
## F-statistic: 4.179 on 2 and 36 DF,  p-value: 0.02333
```

```
summary(tslm(EQ ~ trend+season, data = sales.ts))
```

```
##
## Call:
## tslm(formula = EQ ~ trend + season, data = sales.ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -60.088 -24.310   3.526  22.887  54.688
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 494.4296    23.4631  21.073  < 2e-16 ***
## trend        -8.0681     0.5758 -14.013 2.42e-13 ***
## season2       3.2799    31.1676   0.105  0.91703
## season3     -18.5456    31.1835  -0.595  0.55737
## season4      -5.2705    31.2101  -0.169  0.86726
## season5      19.4173    31.2473   0.621  0.53996
## season6      14.5634    31.2950   0.465  0.64570
## season7      39.8805    31.3532   1.272  0.21509
```

```
## season8       67.1755    31.4218   2.138  0.04249 *
## season9        7.2390    31.5009   0.230  0.82011
## season10     -54.3727    31.5902  -1.721  0.09757 .
## season11     -82.9997    31.6897  -2.619  0.01477 *
## season12     -87.8153    31.7994  -2.762  0.01062 *
## season13    -103.5224    31.9190  -3.243  0.00334 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38.17 on 25 degrees of freedom
## Multiple R-squared:  0.9307, Adjusted R-squared:  0.8946
## F-statistic: 25.81 on 13 and 25 DF,  p-value: 2.934e-11
```

### Build the ARIMAX model

```
#attach(sales_data_signif)

tslm_model <- tslm(EQ ~
trend+season+Medium_rainfall+Social_Search_Impressions+pct_PromoMarketDollars
_Category+

Inflation+EQ_Category+pct_PromoMarketDollars_Subcategory+EQ_Subcategory+
                     Digital_Impressions+Est_ACV_Selling, data = sales.ts)

summary(tslm_model)

##
## Call:
## tslm(formula = EQ ~ trend + season + Medium_rainfall +
Social_Search_Impressions +
##     pct_PromoMarketDollars_Category + Inflation + EQ_Category +
##     pct_PromoMarketDollars_Subcategory + EQ_Subcategory +
Digital_Impressions +
##     Est_ACV_Selling, data = sales.ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.744  -6.809  -0.978   7.100  47.014
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        2.507e+02  3.245e+02   0.773   0.4511
## trend                             -8.094e-01  6.117e+00  -0.132   0.8964
## season2                            3.199e+00  3.097e+01   0.103   0.9190
## season3                           -5.944e+01  5.094e+01  -1.167   0.2603
## season4                           -4.573e+01  7.147e+01  -0.640   0.5313
## season5                           -8.896e+01  1.269e+02  -0.701   0.4932
## season6                           -1.043e+02  1.935e+02  -0.539   0.5974
## season7                           -1.014e+02  2.034e+02  -0.499   0.6248
## season8                           -7.343e+01  1.790e+02  -0.410   0.6870
## season9                           -9.120e+01  1.196e+02  -0.763   0.4568
```

```
## season10                             -8.343e+01  6.875e+01  -1.213   0.2426
## season11                             -1.003e+02  6.359e+01  -1.577   0.1344
## season12                             -9.026e+01  5.087e+01  -1.774   0.0950
## season13                             -9.060e+01  5.110e+01  -1.773   0.0953
## Medium_rainfall                       5.073e+01  5.036e+01   1.007   0.3288
## Social_Search_Impressions            -3.783e-07  1.048e-06  -0.361   0.7229
## pct_PromoMarketDollars_Category       3.543e+03  1.681e+03   2.107   0.0512
## Inflation                            -5.998e+03  2.179e+03  -2.752   0.0142
## EQ_Category                           1.105e-05  1.435e-04   0.077   0.9396
## pct_PromoMarketDollars_Subcategory -5.528e+02  3.800e+02  -1.455   0.1650
## EQ_Subcategory                        1.462e-04  4.567e-04   0.320   0.7531
## Digital_Impressions                  -3.100e-06  1.491e-06  -2.079   0.0540
## Est_ACV_Selling                       2.331e-08  3.040e-08   0.767   0.4543
##
## (Intercept)
## trend
## season2
## season3
## season4
## season5
## season6
## season7
## season8
## season9
## season10
## season11
## season12                             .
## season13                             .
## Medium_rainfall
## Social_Search_Impressions
## pct_PromoMarketDollars_Category      .
## Inflation                            *
## EQ_Category
## pct_PromoMarketDollars_Subcategory
## EQ_Subcategory
## Digital_Impressions                  .
## Est_ACV_Selling
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.82 on 16 degrees of freedom
## Multiple R-squared:  0.9781, Adjusted R-squared:  0.948
## F-statistic: 32.46 on 22 and 16 DF,  p-value: 1.913e-09
```

We see that Adjusted R-sqaure = 94% and RSE = 2.72

```
tsml_lm_model <- lm(EQ ~
Medium_rainfall+Social_Search_Impressions+pct_PromoMarketDollars_Category+
```

```r
Inflation+EQ_Category+pct_PromoMarketDollars_Subcategory+EQ_Subcategory+
                     Digital_Impressions+Est_ACV_Selling, data = sales.ts)

predict(tsml_lm_model)
```

```
##        1        2        3        4        5        6        7        8
## 469.2329 446.9421 449.5451 441.8019 500.4576 487.6894 472.1669 531.7673
##        9       10       11       12       13       14       15       16
## 479.1617 360.4109 318.7774 301.4336 313.6139 386.8081 341.3382 306.6429
##       17       18       19       20       21       22       23       24
## 355.1409 381.4386 336.3903 407.6853 382.2948 335.1704 229.0673 221.5473
##       25       26       27       28       29       30       31       32
## 181.0069 140.9491 266.1263 293.2778 306.1385 170.6975 242.2848 260.2051
##       33       34       35       36       37       38       39
## 270.7875 227.9791 170.1948 155.5118 144.0585 151.5801 149.3684
```

### Forecast for the next 6 Period using the ARIMAX model

```r
forecast(tslm_model$x,h=6)
```

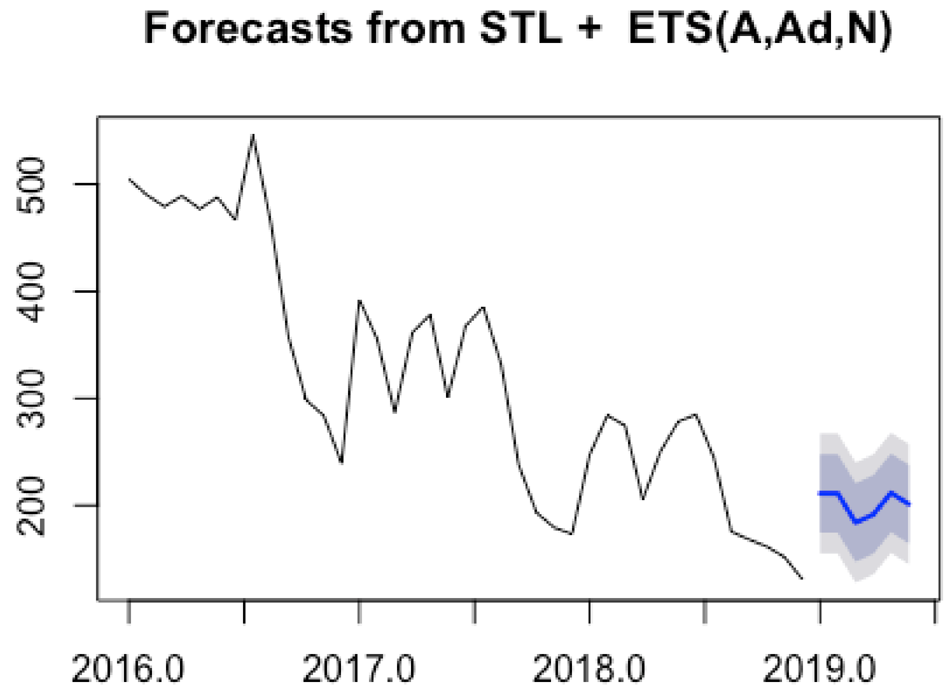
```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2019.000       211.1773 174.4794 247.8753 155.0527 267.3020
## 2019.077       211.1209 174.4230 247.8188 154.9963 267.2455
## 2019.154       183.9475 147.2496 220.6455 127.8229 240.0722
## 2019.231       191.4725 154.7745 228.1704 135.3478 247.5971
## 2019.308       211.7388 175.0409 248.4368 155.6142 267.8635
## 2019.385       201.2157 164.5178 237.9137 145.0910 257.3404
```

```r
forecast(tslm_model$x,h=6)$mean
```

```
## Time Series:
## Start = c(2019, 1)
## End = c(2019, 6)
## Frequency = 13
## [1] 211.1773 211.1209 183.9475 191.4725 211.7388 201.2157
```

### Plot the forecast for next 6 Period

```r
plot(forecast(tslm_model$x,h=6))
```


Forecasts from STL +  ETS(A,Ad,N)
500
400
300
200
2016.0
2017.0
2018.0
2019.0