Source Code: Retail Analysis for Walmart
Canisha Barron

```
#Preparing Libraries
install.packages("pacman")
install.packages (c("vcd","plyr","forecast")
library(vcd)
install.packages("datetime")
library(datetime)
install.packages("dplyr")
library(dplyr)
install.packages("data.table")
library(data.table)
install.packages("ggplot2")
library(ggplot2)
install.packages("EnvStats")
library(EnvStats)
install.packages(c("raster"))
library(sp)
library(raster)
library(usdm)

#Reading Dataset
walmart <-
read.csv("C:/Users/canis/Downloads/1572585064_walmart_store_sales/Walmart_Store_sales.csv")
#Calculating sum of Sales Group by Store Number
X <- aggregate(walmart$Weekly_Sales, by=list(Category=walmart$Store), FUN=sum)
walmart$Data <- as.Date(walmart$Date, format='%d-%m-%Y')
maxi = max(X)

temporary = unlist(subset(X, select=c(x)))
storeNumber = 1
storeWithMaxSales = 1
maxSales = 0
while(storeNumber<=45){
  x=temporary[storeNumber]
  if(x==maxi){
    storeWithMaxSales= storeNumber
    break
  }
  storeNumber = storeNumber + 1
}
#Solution of Q1 ---------------------------------
sprintf("Store with max sales is %i",storeNumber)

ggplot(walmart, aes(x=Store  ,  y=Weekly_Sales)) +
  geom_bar(stat="identity")
```

```
storeNumber = 1
#Standard deviation Store wise
walmart = walmart[with(walmart, order(Data)),]

storeWithMaxSD = 1
maxSD = 0
while(storeNumber<=45){
  s<-subset(walmart, Store == storeNumber)
  p<-subset(s,select=c(Weekly_Sales))
  p = unlist(p)
  x = sd(p)
  print(x)
  if(x>maxSD){
    maxSD = x
    storeWithMaxSD = storeNumber
  }
  storeNumber = storeNumber + 1
}


#Store with Max Standard Deviation
#Solution of Q2 ---------------------------------
sprintf("Store Number %i with SD %f", storeWithMaxSD, maxSD)

Q3onDateBasis2012 = subset(walmart, Data >= '2012-07-01' & Data <= '2012-09-30')
Q3onDateBasis2011 = subset(walmart, Data >= '2011-07-01' & Data <= '2011-09-30')
Q2onDateBasis2012 = subset(walmart, Data >= '2012-04-01' & Data <= '2012-06-30')

print(length(Q3onDateBasis2012$Weekly_Sales))

Q3SalesSum2012 = aggregate(Q3onDateBasis2012$Weekly_Sales, by=list(Q3onDateBasis2012$Store),
FUN=sum)
Q3SalesSum2011 = aggregate(Q3onDateBasis2011$Weekly_Sales, by=list(Q3onDateBasis2011$Store),
FUN=sum)
Q2SalesSum2012 = aggregate(Q2onDateBasis2012$Weekly_Sales, by=list(Q2onDateBasis2012$Store),
FUN=sum)


names(Q3SalesSum2012)[names(Q3SalesSum2012) == "Group.1"] <- "Store"
names(Q3SalesSum2012)[names(Q3SalesSum2012) == "x"] <- "Sales"

names(Q3SalesSum2011)[names(Q3SalesSum2011) == "Group.1"] <- "Store"
names(Q3SalesSum2011)[names(Q3SalesSum2011) == "x"] <- "Sales"


names(Q2SalesSum2012)[names(Q2SalesSum2012) == "Group.1"] <- "Store"
names(Q2SalesSum2012)[names(Q2SalesSum2012) == "x"] <- "Sales"
```

```
Q3SalesSum2012 = Q3SalesSum2012[with(Q3SalesSum2012, order(Store)),]
Q3SalesSum2011 = Q3SalesSum2011[with(Q3SalesSum2011, order(Store)),]
Q2SalesSum2012 = Q2SalesSum2012[with(Q2SalesSum2012, order(Store)),]

resultlist <- vector(mode = "list", length = 46)

storeNumber = 1
while(storeNumber<=45){
  temp = ((Q3SalesSum2012$Sales[storeNumber] -
Q3SalesSum2011$Sales[storeNumber])/(Q3SalesSum2011$Sales[storeNumber]))*100
  print(temp)
  resultlist[storeNumber] = temp
  storeNumber = storeNumber + 1
}

resultlist
highestScore = max(unlist(resultlist))
index = 0

for(x in resultlist){
  if(x==highestScore){
    index = index+1
    break
  }
  index=index+1
}
#Solution of Q3: comparing 2011 Q3 and 2012 Q3---------------------------------
sprintf("Store Number %i has highest Quaterly Growth Rate of %f", index, highestScore)


storeNumber = 1
while(storeNumber<=45){
  temp = ((Q3SalesSum2012$Sales[storeNumber] -
Q2SalesSum2012$Sales[storeNumber])/(Q2SalesSum2012$Sales[storeNumber]))*100
  print(temp)
  resultlist[storeNumber] = temp
  storeNumber = storeNumber + 1
}

resultlist
highestScore = max(unlist(resultlist))
index = 0

for(x in resultlist){
  if(x==highestScore){
    index = index+1
    break
```

```
  }
  index=index+1
}
#Solution of Q3: comparing Q2 and Q3 2012---------------------------------
sprintf("Store Number %i has highest Quarterly Growth Rate of %f", index, highestScore)


storeNumber = 1
while(storeNumber<=45){
  temp = ((Q3SalesSum2012$Sales[storeNumber] -
Q2SalesSum2012$Sales[storeNumber])/(Q2SalesSum2012$Sales[storeNumber]))*100
  print(temp)
  resultlist[storeNumber] = temp
  storeNumber = storeNumber + 1
}

resultlist
highestScore = max(unlist(resultlist))
index = 0

for(x in resultlist){
  if(x==highestScore){
    index = index+1
    break
  }
  index=index+1
}
#Solution of Q3: comparing Q2 and Q3 2012---------------------------------
sprintf("Store Number %i has highest Quarterly Growth Rate of %f", index, highestScore)

nonHolidayDateFrame = subset(walmart, Data!='2010-02-12' & Data!='2011-02-11' & Data!='2012-02-
10' & Data!='2013-02-08' & Data!='2010-09-10' & Data!='2011-09-09' & Data!='2012-09-07' &
Data!='2013-09-06' &
                       Data!='2010-11-26' & Data!='2011-11-25' & Data!='2012-11-23' & Data!='2013-11-29'
&
                       Data!='2010-12-31' & Data!='2011-12-30' & Data!='2012-12-28' & Data!='2013-12-27')

nonHolidaySales = subset(nonHolidayDateFrame, select=c(Weekly_Sales))
nonHolidaySales = nonHolidaySales[-1,]
meanNonHoliday = mean(nonHolidaySales)

HolidayDateFrame = subset(walmart, Data=='2010-02-12' | Data=='2011-02-11' | Data=='2012-02-10' |
Data=='2013-02-08' | Data=='2010-09-10' | Data=='2011-09-09' | Data=='2012-09-07' | Data=='2013-09-
06' |
                     Data=='2010-11-26' | Data=='2011-11-25' | Data=='2012-11-23' | Data=='2013-11-29' |
                     Data=='2010-12-31' | Data=='2011-12-30' | Data=='2012-12-28' | Data=='2013-12-27')
```

```
#Solution of Q4 ---------------------------------
SalesGreater = subset(HolidayDateFrame, Weekly_Sales>meanNonHoliday)
holidayWithHigherSales = subset(SalesGreater, select=c(Data))

print(holidayWithHigherSales)

ggplot(SalesGreater, aes(x=Date  ,  y=Weekly_Sales)) +
  geom_bar(stat="identity")


#Solution of Q5 ---------------------------------

#Visualization of monthly sales of year 2010

tempo = subset(walmart, format(Data, format = "%Y")=="2010", select=c(Weekly_Sales,Data))
tempo = tempo[with(tempo, order(Data)),]
tempo2 <- aggregate(tempo$Weekly_Sales, by=list(Category=months(tempo$Data)), FUN=sum)
subset(tempo2, select=c(x))
names(tempo2)
subset(tempo2, select=c(Category))

sort.month <- function(x, dataframe = NULL, abbreviated = FALSE){
  y <- data.frame(m1 = month.name, m2 = month.abb, n = 1:12)
  z <- if(abbreviated) match(x, y[, 'm2']) else match(x, y[, 'm1'])
  x <- if(is.null(dataframe)) x else dataframe
  h <- data.frame(z, x)
  h[order(z), ][, -1]
}


tempo2=sort.month(tempo2$Category, tempo2)

ws = unlist(subset(tempo2, select=c(x)))
cat = unlist(subset(tempo2, select=c('Category')))
barplot(ws,names.arg=cat,xlab="Months", ylab="Weekly Sales", main="For Year 2010")

#Visualization of monthly sales of year 2011

tempo = subset(walmart, format(Data, format = "%Y")=="2011", select=c(Weekly_Sales,Data))
tempo = tempo[with(tempo, order(Data)),]
tempo2 <- aggregate(tempo$Weekly_Sales, by=list(Category=months(tempo$Data)), FUN=sum)
subset(tempo2, select=c(x))
subset(tempo2, select=c('Category'))

sort.month <- function(x, dataframe = NULL, abbreviated = FALSE){
  y <- data.frame(m1 = month.name, m2 = month.abb, n = 1:12)
  z <- if(abbreviated) match(x, y[, 'm2']) else match(x, y[, 'm1'])
  x <- if(is.null(dataframe)) x else dataframe
```

```
  h <- data.frame(z, x)
  h[order(z), ][, -1]
}


tempo2=sort.month(tempo2$Category, tempo2)

ws = unlist(subset(tempo2, select=c(x)))
cat = unlist(subset(tempo2, select=c('Category')))
barplot(ws,names.arg=cat,xlab="Months", ylab="Weekly Sales", main="For Year 2011")

#Visualization of monthly sales of year 2012

tempo = subset(walmart, format(Data, format = "%Y")=="2012", select=c(Weekly_Sales,Data))
tempo = tempo[with(tempo, order(Data)),]
tempo2 <- aggregate(tempo$Weekly_Sales, by=list(Category=months(tempo$Data)), FUN=sum)
subset(tempo2, select=c(x))
names(tempo2)
subset(tempo2, select=c(Category))

sort.month <- function(x, dataframe = NULL, abbreviated = FALSE){
  y <- data.frame(m1 = month.name, m2 = month.abb, n = 1:12)
  z <- if(abbreviated) match(x, y[, 'm2']) else match(x, y[, 'm1'])
  x <- if(is.null(dataframe)) x else dataframe
  h <- data.frame(z, x)
  h[order(z), ][, -1]
}


tempo2=sort.month(tempo2$Category, tempo2)

ws = unlist(subset(tempo2, select=c(x)))


cat = unlist(subset(tempo2, select=c('Category')))
barplot(ws,names.arg=cat,xlab="Months", ylab="Weekly Sales", main="For Year 2012")

#Visualization Semester Wise for Year 2010

tempo = subset(walmart, format(Data, format = "%Y")=="2010", select=c(Weekly_Sales,Data))
tempo = tempo[with(tempo, order(Data)),]
tempo2 <- aggregate(tempo$Weekly_Sales, by=list(months(tempo$Data)), FUN=sum)
subset(tempo2, select=c(x))
subset(tempo2, select=c(Group.1))

tempdf = tempo2
tempdf
```

```
tempdf$Group.1[tempdf$Group.1=="February"]<-0
tempdf$Group.1[tempdf$Group.1=="March"]<-0
tempdf$Group.1[tempdf$Group.1=="April"]<-0
tempdf$Group.1[tempdf$Group.1=="May"]<-0
tempdf$Group.1[tempdf$Group.1=="June"]<-0


tempdf$Group.1[tempdf$Group.1=="July"]<-1
tempdf$Group.1[tempdf$Group.1=="August"]<-1
tempdf$Group.1[tempdf$Group.1=="September"]<-1
tempdf$Group.1[tempdf$Group.1=="October"]<-1
tempdf$Group.1[tempdf$Group.1=="November"]<-1
tempdf$Group.1[tempdf$Group.1=="December"]<-1

tempo2 <- aggregate(tempdf$x, by=list(tempdf$Group.1), FUN=sum)

tempo2$Group.1[tempo2$Group.1==0]<-"Semester 1"
tempo2$Group.1[tempo2$Group.1==1]<-"Semester 2"

ws = unlist(subset(tempo2, select=c(x)))
cat = unlist(subset(tempo2, select=c(Group.1)))
barplot(ws,names.arg=cat,xlab="Semesters", ylab="Weekly Sales", main="For Year 2010 - Sem wise")

#Visualization Semester Wise for Year 2011

tempo = subset(walmart, format(Data, format = "%Y")=="2011", select=c(Weekly_Sales,Data))
tempo = tempo[with(tempo, order(Data)),]
tempo2 <- aggregate(tempo$Weekly_Sales, by=list(months(tempo$Data)), FUN=sum)
subset(tempo2, select=c(x))
subset(tempo2, select=c(Group.1))

tempdf = tempo2
tempdf

tempdf$Group.1[tempdf$Group.1=="January"]<-0
tempdf$Group.1[tempdf$Group.1=="February"]<-0
tempdf$Group.1[tempdf$Group.1=="March"]<-0
tempdf$Group.1[tempdf$Group.1=="April"]<-0
tempdf$Group.1[tempdf$Group.1=="May"]<-0
tempdf$Group.1[tempdf$Group.1=="June"]<-0


tempdf$Group.1[tempdf$Group.1=="July"]<-1
tempdf$Group.1[tempdf$Group.1=="August"]<-1
tempdf$Group.1[tempdf$Group.1=="September"]<-1
tempdf$Group.1[tempdf$Group.1=="October"]<-1
tempdf$Group.1[tempdf$Group.1=="November"]<-1
```

```
tempdf$Group.1[tempdf$Group.1=="December"]<-1

tempo2 <- aggregate(tempdf$x, by=list(tempdf$Group.1), FUN=sum)

tempo2$Group.1[tempo2$Group.1==0]<-"Semester 1"
tempo2$Group.1[tempo2$Group.1==1]<-"Semester 2"

ws = unlist(subset(tempo2, select=c(x)))
cat = unlist(subset(tempo2, select=c(Group.1)))
barplot(ws,names.arg=cat,xlab="Semesters", ylab="Weekly Sales", main="For Year 2011 - Sem wise")

#Visualization Semester Wise for Year 2012

tempo = subset(walmart, format(Data, format = "%Y")=="2012", select=c(Weekly_Sales,Data))
tempo = tempo[with(tempo, order(Data)),]
tempo2 <- aggregate(tempo$Weekly_Sales, by=list(months(tempo$Data)), FUN=sum)
subset(tempo2, select=c(x))
subset(tempo2, select=c(Group.1))

tempdf = tempo2
tempdf

tempdf$Group.1[tempdf$Group.1=="January"]<-0
tempdf$Group.1[tempdf$Group.1=="February"]<-0
tempdf$Group.1[tempdf$Group.1=="March"]<-0
tempdf$Group.1[tempdf$Group.1=="April"]<-0
tempdf$Group.1[tempdf$Group.1=="May"]<-0
tempdf$Group.1[tempdf$Group.1=="June"]<-0


tempdf$Group.1[tempdf$Group.1=="July"]<-1
tempdf$Group.1[tempdf$Group.1=="August"]<-1
tempdf$Group.1[tempdf$Group.1=="September"]<-1
tempdf$Group.1[tempdf$Group.1=="October"]<-1
tempdf$Group.1[tempdf$Group.1=="November"]<-1
tempdf$Group.1[tempdf$Group.1=="December"]<-1

tempo2 <- aggregate(tempdf$x, by=list(tempdf$Group.1), FUN=sum)

tempo2$Group.1[tempo2$Group.1==0]<-"Semester 1"
tempo2$Group.1[tempo2$Group.1==1]<-"Semester 2"

ws = unlist(subset(tempo2, select=c(x)))
cat = unlist(subset(tempo2, select=c(Group.1)))
barplot(ws,names.arg=cat,xlab="Semesters", ylab="Weekly Sales", main="For Year 2012 - Sem wise")

#Linear regression
df<- read.csv("C:/Users/canis/Downloads/1572585064_walmart_store_sales/Walmart_Store_sales.csv")
```

```
head(df);
lapply(df,class)
install.packages("datetime")
library(datetime)
df[is.na(df)] <- 0;
df$Data<-as.Date(df$Date,format='%d-%m-%Y')
df$Date=as.Date(df$Date,format='%d-%m-%Y')
df_c<-df[1:100,]
df = df[with(df, order(Data)),]
date_rs<-vector(mode="list",length=nrow(df))

for(row in 1:nrow(df)){
  date <- df[row, "Data"]
  x <- as.Date("2010-02-05")
  diff<-julian(date,x)
  date_rs[row]=diff/7+1
}
df$date_rs<-date_rs


#Ho:Temperature, CPI, Fuel Price, Unemployment and date are not influencing weekly sales.

#input<-
df[1:nrow(df),c("Store"==1,"Weekly_Sales","Temperature","Fuel_Price","CPI","Unemployment")]
input = subset(subset(df, Store==1),
select=c('Weekly_Sales','Temperature','Fuel_Price','CPI','Unemployment'))

cor(input)
plot(input)

#Method 1:VIF METHOD
#Check multicollinearity

vifstep(input[,-1], th=10)

model<-lm(Weekly_Sales~Temperature+Fuel_Price+CPI+Unemployment, data = input)
print(model)
summary(model)

names(model)
model$fitted.values#predicted values for my dataset
input$Weekly_Sales
model$residuals

plot(model)


y<-coef(model)[1]
```

```
xtemp<-coef(model)[2]
xfuel<-coef(model)[3]
xcpi<-coef(model)[4]
xunemp<-coef(model)[5]

#weekly_sales = -2727200 - 2426temp -31637fuel + 17872cpi + 90632umep

#Method 2: Significance Method

model1<-lm(Weekly_Sales~Temperature+Fuel_Price+CPI+Unemployment, data = input)
print(model1)
summary(model1)

names(model1)
model1$fitted.values #predicted values for my dataset
input$Weekly_Sales
model1$residuals

plot(model1)

vifstep(input[,c(-3,-4)], th=10)


#removed Fuel_Price and Unemployment due to high p values.

model2<- lm(Weekly_Sales~Temperature+CPI, data = input)
summary(model2)

names(model2)
model2$fitted.values #predicted values for my dataset
input$Weekly_Sales
model2$residuals

plot(model2)

y1<-coef(model2)[1]
xtemp1<-coef(model2)[2]
xcpi1<-coef(model2)[3]

#Weekly_Sales=-233190-2769temp1+9156cpi1

vifstep(input[,c(-3,-4)], th=10)


#H0:there is no correlation amongst residuals,
#i.e. they are independent
#in case of regression, we need high p value so that we cannot reject the null
```

```
##Method 3 :step

model3<-step(model)

#Akaike information criterion
#AIC : balance accuracy and complexity


summary(model3)

#WeeklySales=-2048494.5 -2587.7temp+14730.4CPI+78679.8Unemp
names(model3)

input$Weekly_Sales
model3$fitted.values
model3$residuals

plot(model3)

y2<-coef(model3)[1]
xtemp2<-coef(model3)[2]
xcpi2<-coef(model3)[3]
xumep2<-coef(model3)[4]

#weeklysales=-2048494 -2588temp2+14730cpi2+78680unemp2

#based on p value higher than 0.05 we eliminate Unemployment from the model. ('refer to model 2')

#dwtest for all models
install.packages("lmtest")
library(zoo)
library(lmtest)
dwtest(model1)
dwtest(model2)
dwtest(model3)


#predict(model2, newdata=input)
predict(model2)
model2$fitted.values
a=as.data.frame(predict(model2))

View(a)
names(a)

#MAPE - mean absolute percentage Error
model2$residuals
```

```
model2$residuals/input$Weekly_Sales #PE

abs(model2$residuals/input$Weekly_Sales) #APE

mean(abs(model2$residuals/input$Weekly_Sales)) #MAPE
mean(abs(model3$residuals/input$Weekly_Sales))
mean(abs(model1$residuals/input$Weekly_Sales))
mean(abs(model$residuals/input$Weekly_Sales))

#result=y+xtemp*x1+xfuel*x2+xcpi*x3+xunemp*x4+xdate*x5

dev.off()
```

Source Code Execution

```
> #Preparing Libraries
> install.packages("pacman")
Installing package into 'C:/Users/canis/Documents/R/win-library/4.0'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/pacman_0.5.1.zip'
Content type 'application/zip' length 390259 bytes (381 KB)
downloaded 381 KB

package 'pacman' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\canis\AppData\Local\Temp\RtmpoRGk2Y\downloaded_packages
> install.packages (c("vcd","plyr","forecast")
+ library(vcd)
Error: unexpected symbol in:
"install.packages (c("vcd","plyr","forecast")
library"
> install.packages("datetime")
Installing package into 'C:/Users/canis/Documents/R/win-library/4.0'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/datetime_0.1.4.zip'
Content type 'application/zip' length 62065 bytes (60 KB)
downloaded 60 KB

package 'datetime' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\canis\AppData\Local\Temp\RtmpoRGk2Y\downloaded_packages
> library(datetime)
Warning message:
package 'datetime' was built under R version 4.0.4
> install.packages("dplyr")
Installing package into 'C:/Users/canis/Documents/R/win-library/4.0'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/dplyr_1.0.4.zip'
Content type 'application/zip' length 1327782 bytes (1.3 MB)
downloaded 1.3 MB

package 'dplyr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\canis\AppData\Local\Temp\RtmpoRGk2Y\downloaded_packages
> library(dplyr)

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union

Warning message:
package 'dplyr' was built under R version 4.0.4
```

```
> install.packages("data.table")
Installing package into 'C:/Users/canis/Documents/R/win-library/4.0'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/data.table_1.14.0.zip'
Content type 'application/zip' length 2603296 bytes (2.5 MB)
downloaded 2.5 MB

package 'data.table' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\canis\AppData\Local\Temp\RtmpoRGk2Y\downloaded_packages
> library(data.table)
data.table 1.14.0 using 4 threads (see ?getDTthreads).  Latest news: r-datatable.com

Attaching package: 'data.table'

The following objects are masked from 'package:dplyr':

    between, first, last

Warning message:
package 'data.table' was built under R version 4.0.4
> install.packages("ggplot2")
Installing package into 'C:/Users/canis/Documents/R/win-library/4.0'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/ggplot2_3.3.3.zip'
Content type 'application/zip' length 4070114 bytes (3.9 MB)
downloaded 3.9 MB

package 'ggplot2' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\canis\AppData\Local\Temp\RtmpoRGk2Y\downloaded_packages
> library(ggplot2)
Warning message:
package 'ggplot2' was built under R version 4.0.4
> install.packages("EnvStats")
Installing package into 'C:/Users/canis/Documents/R/win-library/4.0'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/EnvStats_2.4.0.zip'
Content type 'application/zip' length 6060293 bytes (5.8 MB)
downloaded 5.8 MB

package 'EnvStats' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\canis\AppData\Local\Temp\RtmpoRGk2Y\downloaded_packages
> library(EnvStats)

Attaching package: 'EnvStats'

The following objects are masked from 'package:stats':

    predict, predict.lm
```

```
The following object is masked from ‘package:base’:

    print.default

Warning message:
package ‘EnvStats’ was built under R version 4.0.4
> install.packages(c("raster"))
Installing package into ‘C:/Users/canis/Documents/R/win-library/4.0’
(as ‘lib’ is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/raster_3.4-5.zip'
Content type 'application/zip' length 3563112 bytes (3.4 MB)
downloaded 3.4 MB

package ‘raster’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\canis\AppData\Local\Temp\RtmpoRGk2Y\downloaded_packages
> library(sp)
> library(raster)

Attaching package: ‘raster’

The following object is masked from ‘package:EnvStats’:

    cv

The following object is masked from ‘package:data.table’:

    shift

The following object is masked from ‘package:dplyr’:

    select

Warning message:
package ‘raster’ was built under R version 4.0.4
> library(usdm)
> #Reading Dataset
> walmart <- read.csv("C:/Users/canis/Downloads/1572585064_walmart_store_sales/Walmart_Store_sales.csv")
> X <- aggregate(walmart$Weekly_Sales, by=list(Category=walmart$Store), FUN=sum)
> walmart$Data <- as.Date(walmart$Date, format='%d-%m-%Y')
> maxi = max(X)
>
> temporary = unlist(subset(X, select=c(x)))
> storeNumber = 1
> storeWithMaxSales = 1
> maxSales = 0
> while(storeNumber<=45){
+   x=temporary[storeNumber]
+   if(x==maxi){
+     storeWithMaxSales= storeNumber
+     break
+   }
+   storeNumber = storeNumber + 1
+ }
```

```
+   storeNumber = storeNumber + 1
+ }
> #Solution of Q1 ---------------------------------
> sprintf("Store with max sales is %i",storeNumber)
[1] "Store with max sales is 20"
> ggplot(walmart, aes(x=Store  ,  y=Weekly_Sales)) +
+   geom_bar(stat="identity")
> storeNumber = 1
> #Standard deviation Store wise
> walmart = walmart[with(walmart, order(Data)),]
>
> storeWithMaxSD = 1
> maxSD = 0
> while(storeNumber<=45){
+   s<-subset(walmart, Store == storeNumber)
+   p<-subset(s,select=c(Weekly_Sales))
+   p = unlist(p)
+   x = sd(p)
+   print(x)
+   if(x>maxSD){
+     maxSD = x
+     storeWithMaxSD = storeNumber
+   }
+   storeNumber = storeNumber + 1
+ }
[1] 155980.8
[1] 237683.7
[1] 46319.63
[1] 266201.4
[1] 37737.97
[1] 212525.9
[1] 112585.5
[1] 106280.8
[1] 69028.67
[1] 302262.1
[1] 165833.9
[1] 139166.9
[1] 265507
[1] 317569.9
[1] 120538.7
[1] 85769.68
[1] 112162.9
[1] 176641.5
[1] 191722.6
[1] 275900.6
[1] 128752.8
[1] 161251.4
[1] 249788
[1] 167745.7
[1] 112976.8
[1] 110431.3
[1] 239930.1
[1] 181759
[1] 99120.14
[1] 22809.67
[1] 125855.9
```

```
[1] 138017.3
[1] 24132.93
[1] 104630.2
[1] 211243.5
[1] 60725.17
[1] 21837.46
[1] 42768.17
[1] 217466.5
[1] 119002.1
[1] 187907.2
[1] 50262.93
[1] 40598.41
[1] 24762.83
[1] 130168.5
>
>
> #Store with Max Standard Deviation
> #Solution of Q2 ---------------------------------
> sprintf("Store Number %i with SD %f", storeWithMaxSD, maxSD)
[1] "Store Number 14 with SD 317569.949476"
>
```

```
> Q3onDateBasis2012 = subset(walmart, Data >= '2012-07-01' & Data <= '2012-09-30')
> Q3onDateBasis2011 = subset(walmart, Data >= '2011-07-01' & Data <= '2011-09-30')
> Q2onDateBasis2012 = subset(walmart, Data >= '2012-04-01' & Data <= '2012-06-30')
>
> print(length(onDateBasis2012$Weekly_Sales))
[1] 405
>
> Q3SalesSum2012 = aggregate(Q3onDateBasis2012$Weekly_Sales, by=list(Q3onDateBasis2012$Store), FUN=sum)
> Q3SalesSum2011 = aggregate(Q3onDateBasis2011$Weekly_Sales, by=list(Q3onDateBasis2011$Store), FUN=sum)
> Q2SalesSum2012 = aggregate(Q2onDateBasis2012$Weekly_Sales, by=list(Q2onDateBasis2012$Store), FUN=sum)
>
>
> names(Q3SalesSum2012)[names(Q3SalesSum2012) == "Group.1"] <- "Store"
> names(Q3SalesSum2012)[names(Q3SalesSum2012) == "x"] <- "Sales"
>
> names(Q3SalesSum2011)[names(Q3SalesSum2011) == "Group.1"] <- "Store"
> names(Q3SalesSum2011)[names(Q3SalesSum2011) == "x"] <- "Sales"
>
>
> names(Q2SalesSum2012)[names(Q2SalesSum2012) == "Group.1"] <- "Store"
> names(Q2SalesSum2012)[names(Q2SalesSum2012) == "x"] <- "Sales"
>
> Q3SalesSum2012 = Q3SalesSum2012[with(Q3SalesSum2012, order(Store)),]
> Q3SalesSum2011 = Q3SalesSum2011[with(Q3SalesSum2011, order(Store)),]
> Q2SalesSum2012 = Q2SalesSum2012[with(Q2SalesSum2012, order(Store)),]
>
> resultlist <- vector(mode = "list", length = 46)
>
> storeNumber = 1
> while(storeNumber<=45){
+   temp = ((Q3SalesSum2012$Sales[storeNumber] - Q3SalesSum2011$Sales[storeNumber])/(Q3SalesSum2011$Sales[storeNumber]))*100
+   print(temp)
+   resultlist[storeNumber] = temp
+   storeNumber = storeNumber + 1
+ }
[1] 18.36798
[1] 19.186
[1] 24.93697
[1] 16.81586
[1] 22.09876
[1] 17.65615
[1] 13.60747
[1] 18.75509
[1] 17.91852
[1] 12.62342
[1] 16.63416
[1] 17.1501
[1] 15.61218
[1] -8.478813
[1] 11.66642
[1] 10.27474
```

```
[1] 18.51044
[1] 22.28124
[1] 12.67353
[1] 16.86429
[1] 6.235572
[1] 12.18716
[1] 17.02651
[1] 13.64242
[1] 17.65787
[1] 15.50716
[1] 7.773545
[1] 13.33421
[1] 13.49094
[1] 19.93003
[1] 10.76401
[1] 15.04512
[1] 19.72019
[1] 18.34933
[1] 20.28151
[1] -6.161256
[1] 13.89575
[1] 25.2114
[1] 23.99735
[1] 14.34895
[1] 22.28689
[1] 16.14728
[1] 14.42565
[1] 26.68068
[1] 11.03196
>
> resultlist
[[1]]
[1] 18.36798

[[2]]
[1] 19.186

[[3]]
[1] 24.93697

[[4]]
[1] 16.81586

[[5]]
[1] 22.09876

[[6]]
[1] 17.65615

[[7]]
[1] 13.60747
```

```
[[8]]
[1] 18.75509

[[9]]
[1] 17.91852

[[10]]
[1] 12.62342

[[11]]
[1] 16.63416

[[12]]
[1] 17.1501

[[13]]
[1] 15.61218

[[14]]
[1] -8.478813

[[15]]
[1] 11.66642

[[16]]
[1] 10.27474

[[17]]
[1] 18.51044

[[18]]
[1] 22.28124

[[19]]
[1] 12.67353

[[20]]
[1] 16.86429

[[21]]
[1] 6.235572

[[22]]
[1] 12.18716

[[23]]
[1] 17.02651
```

```
[[24]]
[1] 13.64242

[[25]]
[1] 17.65787

[[26]]
[1] 15.50716

[[27]]
[1] 7.773545

[[28]]
[1] 13.33421

[[29]]
[1] 13.49094

[[30]]
[1] 19.93003

[[31]]
[1] 10.76401

[[32]]
[1] 15.04512

[[33]]
[1] 19.72019

[[34]]
[1] 18.34933

[[35]]
[1] 20.28151

[[36]]
[1] -6.161256

[[37]]
[1] 13.89575

[[38]]
[1] 25.2114

[[39]]
[1] 23.99735

[[40]]
[1] 14.34895
```

```
[[41]]
[1] 22.28689

[[42]]
[1] 16.14728

[[43]]
[1] 14.42565

[[44]]
[1] 26.68068

[[45]]
[1] 11.03196

[[46]]
NULL

> highestScore = max(unlist(resultlist))
> index = 0
>
> for(x in resultlist){
+   if(x==highestScore){
+     index = index+1
+     break
+   }
+   index=index+1
+ }
> #Solution of Q3: comparing 2011 Q3 and 2012 Q3---------------------------------
> sprintf("Store Number %i has highest Quaterly Growth Rate of %f", index, highestScore)
[1] "Store Number 44 has highest Quaterly Growth Rate of 26.680678"
>
>
> storeNumber = 1
> while(storeNumber<=45){
+   temp = ((Q3SalesSum2012$Sales[storeNumber] - Q2SalesSum2012$Sales[storeNumber])/(Q2SalesSum2012$Sales[storeNumber]))*100
+   print(temp)
+   resultlist[storeNumber] = temp
+   storeNumber = storeNumber + 1
+ }
[1] 9.124088
[1] 8.934926
[1] 8.916682
[1] 10.70152
[1] 6.523068
[1] 7.261511
[1] 22.21068
[1] 12.38289
[1] 7.685592
[1] 9.312615
```

```
[1] 11.32969
[1] 7.108835
[1] 9.38
[1] -3.934138
[1] 6.232995
[1] 19.54933
[1] 8.018571
[1] 10.44146
[1] 12.2343
[1] 12.07182
[1] 8.924469
[1] 7.954657
[1] 14.41799
[1] 12.1731
[1] 10.46928
[1] 16.35947
[1] 8.255861
[1] 9.622808
[1] 4.662999
[1] 9.601719
[1] 9.649143
[1] 12.75627
[1] 7.301542
[1] 8.650341
[1] 15.14907
[1] 3.699963
[1] 12.64987
[1] 12.14503
[1] 16.33781
[1] 13.40836
[1] 15.23008
[1] 9.170403
[1] 9.00995
[1] 14.41535
[1] 3.273038
>
> resultlist
[[1]]
[1] 9.124088

[[2]]
[1] 8.934926

[[3]]
[1] 8.916682

[[4]]
[1] 10.70152

[[5]]
[1] 6.523068
```

```
[[6]]
[1] 7.261511

[[7]]
[1] 22.21068

[[8]]
[1] 12.38289

[[9]]
[1] 7.685592

[[10]]
[1] 9.312615

[[11]]
[1] 11.32969

[[12]]
[1] 7.108835

[[13]]
[1] 9.38

[[14]]
[1] -3.934138

[[15]]
[1] 6.232995

[[16]]
[1] 19.54933

[[17]]
[1] 8.018571

[[18]]
[1] 10.44146

[[19]]
[1] 12.2343

[[20]]
[1] 12.07182

[[21]]
[1] 8.924469

[[22]]
[1] 7.954657
```

```
[[23]]
[1] 14.41799

[[24]]
[1] 12.1731

[[25]]
[1] 10.46928

[[26]]
[1] 16.35947

[[27]]
[1] 8.255861

[[28]]
[1] 9.622808

[[29]]
[1] 4.662999

[[30]]
[1] 9.601719

[[31]]
[1] 9.649143

[[32]]
[1] 12.75627

[[33]]
[1] 7.301542

[[34]]
[1] 8.650341

[[35]]
[1] 15.14907

[[36]]
[1] 3.699963

[[37]]
[1] 12.64987

[[38]]
[1] 12.14503
```

```
[[39]]
[1] 16.33781

[[40]]
[1] 13.40836

[[41]]
[1] 15.23008

[[42]]
[1] 9.170403

[[43]]
[1] 9.00995

[[44]]
[1] 14.41535

[[45]]
[1] 3.273038

[[46]]
NULL

> highestScore = max(unlist(resultlist))
> index = 0
>
> for(x in resultlist){
+   if(x==highestScore){
+     index = index+1
+     break
+   }
+   index=index+1
+ }
> #Solution of Q3: comparing Q2 and Q3 2012---------------------------------
> sprintf("Store Number %i has highest Quarterly Growth Rate of %f", index, highestScore)
[1] "Store Number 7 has highest Quarterly Growth Rate of 22.210682"
>
```

```
> #Solution of Q4 ---------------------------------
> SalesGreater = subset(HolidayDateFrame, Weekly_Sales>meanNonHoliday)
> holidayWithHigherSales = subset(SalesGreater, select=c(Data))
>
> print(holidayWithHigherSales)
           Data
43   2010-11-26
186  2010-11-26
472  2010-11-26
758  2010-11-26
1044 2010-11-26
1330 2010-11-26
1473 2010-11-26
1616 2010-11-26
1759 2010-11-26
1902 2010-11-26
2045 2010-11-26
2474 2010-11-26
2617 2010-11-26
2760 2010-11-26
2903 2010-11-26
3046 2010-11-26
3189 2010-11-26
3332 2010-11-26
3475 2010-11-26
3618 2010-11-26
3761 2010-11-26
3904 2010-11-26
4333 2010-11-26
4476 2010-11-26
4762 2010-11-26
4905 2010-11-26
5477 2010-11-26
5620 2010-11-26
5763 2010-11-26
6335 2010-11-26
48   2010-12-31
191  2010-12-31
477  2010-12-31
763  2010-12-31
1335 2010-12-31
1478 2010-12-31
1764 2010-12-31
1907 2010-12-31
2622 2010-12-31
2765 2010-12-31
3194 2010-12-31
3337 2010-12-31
3766 2010-12-31
3909 2010-12-31
4338 2010-12-31
5482 2010-12-31
95   2011-11-25
238  2011-11-25
```

```
524  2011-11-25
810  2011-11-25
1096 2011-11-25
1382 2011-11-25
1525 2011-11-25
1668 2011-11-25
1811 2011-11-25
1954 2011-11-25
2097 2011-11-25
2383 2011-11-25
2526 2011-11-25
2669 2011-11-25
2812 2011-11-25
2955 2011-11-25
3098 2011-11-25
3241 2011-11-25
3384 2011-11-25
3527 2011-11-25
3670 2011-11-25
3813 2011-11-25
3956 2011-11-25
4385 2011-11-25
4528 2011-11-25
4814 2011-11-25
4957 2011-11-25
5529 2011-11-25
5672 2011-11-25
5815 2011-11-25
6387 2011-11-25
100  2011-12-30
243  2011-12-30
529  2011-12-30
815  2011-12-30
1387 2011-12-30
1530 2011-12-30
1673 2011-12-30
1816 2011-12-30
1959 2011-12-30
2674 2011-12-30
2817 2011-12-30
3246 2011-12-30
3389 2011-12-30
3818 2011-12-30
3961 2011-12-30
4390 2011-12-30
4533 2011-12-30
5534 2011-12-30
5820 2011-12-30
```

```
> ggplot(SalesGreater, aes(x=Date  ,  y=Weekly_Sales)) +
+   geom_bar(stat="identity")
>
```

```
> #Solution of Q5 ---------------------------------
>
> #Visualization of monthly sales of year 2010
>
> tempo = subset(walmart, format(Data, format = "%Y")=="2010", select=c(Weekly_Sales,Data))
> tempo = tempo[with(tempo, order(Data)),]
> tempo2 <- aggregate(tempo$Weekly_Sales, by=list(Category=months(tempo$Data)), FUN=sum)
> subset(tempo2, select=c(x))
          x
1  133623246
2  139435112
3  183184734
4   92245565
5  135763112
6   94231083
7   89122936
8  138207691
9  110946587
10 129818974
11  84439241
> names(tempo2)
[1] "Category" "x"
> subset(tempo2, select=c(Category))
    Category
1      April
2     August
3   December
4   February
5       July
6       June
7      March
8        May
9   November
10   October
11 September
>
> sort.month <- function(x, dataframe = NULL, abbreviated = FALSE){
+   y <- data.frame(m1 = month.name, m2 = month.abb, n = 1:12)
+   z <- if(abbreviated) match(x, y[, 'm2']) else match(x, y[, 'm1'])
+   x <- if(is.null(dataframe)) x else dataframe
+   h <- data.frame(z, x)
+   h[order(z), ][, -1]
+ }
>
>
> tempo2=sort.month(tempo2$Category, tempo2)
>
> ws = unlist(subset(tempo2, select=c(x)))
> cat = unlist(subset(tempo2, select=c('Category')))
> barplot(ws,names.arg=cat,xlab="Months", ylab="Weekly Sales", main="For Year 2010")
```

```
> #visualization of monthly sales of year 2011
>
> tempo = subset(walmart, format(Data, format = "%Y")=="2011", select=c(Weekly_Sales,Data))
> tempo = tempo[with(tempo, order(Data)),]
> tempo2 <- aggregate(tempo$Weekly_Sales, by=list(Category=months(tempo$Data)), FUN=sum)
> subset(tempo2, select=c(x))
           x
1  137180053
2   94334296
3  183126398
4   92842024
5  120928179
6  134473616
7   93331656
8   87748525
9  134786200
10 113032586
11 136049595
12 128707888
> subset(tempo2, select=c('Category'))
    Category
1      April
2     August
3   December
4   February
5    January
6       July
7       June
8      March
9        May
10  November
11   October
12 September
>
> sort.month <- function(x, dataframe = NULL, abbreviated = FALSE){
+   y <- data.frame(m1 = month.name, m2 = month.abb, n = 1:12)
+   z <- if(abbreviated) match(x, y[, 'm2']) else match(x, y[, 'm1'])
+   x <- if(is.null(dataframe)) x else dataframe
+   h <- data.frame(z, x)
+   h[order(z), ][, -1]
+ }
>
>
> tempo2=sort.month(tempo2$Category, tempo2)
>
> ws = unlist(subset(tempo2, select=c(x)))
> cat = unlist(subset(tempo2, select=c('Category')))
> barplot(ws,names.arg=cat,xlab="Months", ylab="Weekly Sales", main="For Year 2011")
>
```

```
> #Visualization of monthly sales of year 2012
>
> tempo = subset(walmart, format(Data, format = "%Y")=="2012", select=c(Weekly_Sales,Data))
> tempo = tempo[with(tempo, order(Data)),]
> tempo2 <- aggregate(tempo$Weekly_Sales, by=list(Category=months(tempo$Data)), FUN=sum)
> subset(tempo2, select=c(x))
           x
1  135418590
2  141961415
3   95968564
4  123939050
5  136256431
6  142677508
7  137168161
8   94716403
9   90666527
10 132315485
> names(tempo2)
[1] "Category" "x"
> subset(tempo2, select=c(Category))
    Category
1      April
2     August
3   February
4    January
5       July
6       June
7      March
8        May
9    October
10 September
>
> sort.month <- function(x, dataframe = NULL, abbreviated = FALSE){
+   y <- data.frame(m1 = month.name, m2 = month.abb, n = 1:12)
+   z <- if(abbreviated) match(x, y[, 'm2']) else match(x, y[, 'm1'])
+   x <- if(is.null(dataframe)) x else dataframe
+   h <- data.frame(z, x)
+   h[order(z), ][, -1]
+ }
>
>
> tempo2=sort.month(tempo2$Category, tempo2)
>
> ws = unlist(subset(tempo2, select=c(x)))
>
>
> cat = unlist(subset(tempo2, select=c('Category')))
> barplot(ws,names.arg=cat,xlab="Months", ylab="Weekly Sales", main="For Year 2012")
>
```

```
> #Visualization Semester Wise for Year 2010
>
> tempo = subset(walmart, format(Data, format = "%Y")=="2010", select=c(Weekly_Sales,Data))
> tempo = tempo[with(tempo, order(Data)),]
> tempo2 <- aggregate(tempo$Weekly_Sales, by=list(months(tempo$Data)), FUN=sum)
> subset(tempo2, select=c(x))
          x
1  133623246
2  139435112
3  183184734
4   92245565
5  135763112
6   94231083
7   89122936
8  138207691
9  110946587
10 129818974
11  84439241
> subset(tempo2, select=c(Group.1))
     Group.1
1      April
2     August
3   December
4   February
5       July
6       June
7      March
8        May
9   November
10   October
11 September
>
> tempdf = tempo2
> tempdf
     Group.1         x
1      April 133623246
2     August 139435112
3   December 183184734
4   February  92245565
5       July 135763112
6       June  94231083
7      March  89122936
8        May 138207691
9   November 110946587
10   October 129818974
11 September  84439241
>
>
> tempdf$Group.1[tempdf$Group.1=="February"]<-0
> tempdf$Group.1[tempdf$Group.1=="March"]<-0
> tempdf$Group.1[tempdf$Group.1=="April"]<-0
> tempdf$Group.1[tempdf$Group.1=="May"]<-0
> tempdf$Group.1[tempdf$Group.1=="June"]<-0
>


>
> tempdf$Group.1[tempdf$Group.1=="July"]<-1
> tempdf$Group.1[tempdf$Group.1=="August"]<-1
> tempdf$Group.1[tempdf$Group.1=="September"]<-1
> tempdf$Group.1[tempdf$Group.1=="October"]<-1
> tempdf$Group.1[tempdf$Group.1=="November"]<-1
> tempdf$Group.1[tempdf$Group.1=="December"]<-1
>
> tempo2 <- aggregate(tempdf$x, by=list(tempdf$Group.1), FUN=sum)
>
> tempo2$Group.1[tempo2$Group.1==0]<-"Semester 1"
> tempo2$Group.1[tempo2$Group.1==1]<-"Semester 2"
>
> ws = unlist(subset(tempo2, select=c(x)))
> cat = unlist(subset(tempo2, select=c(Group.1)))
> barplot(ws,names.arg=cat,xlab="Semesters", ylab="Weekly Sales", main="For Year 2010 - Sem wise")
>
```

```
> #Visualization Semester Wise for Year 2011
>
> tempo = subset(walmart, format(Data, format = "%Y")=="2011", select=c(Weekly_Sales,Data))
> tempo = tempo[with(tempo, order(Data)),]
> tempo2 <- aggregate(tempo$Weekly_Sales, by=list(months(tempo$Data)), FUN=sum)
> subset(tempo2, select=c(x))
           x
1  137180053
2   94334296
3  183126398
4   92842024
5  120928179
6  134473616
7   93331656
8   87748525
9  134786200
10 113032586
11 136049595
12 128707888
> subset(tempo2, select=c(Group.1))
     Group.1
1      April
2     August
3   December
4   February
5    January
6       July
7       June
8      March
9        May
10  November
11   October
12 September
>
> tempdf = tempo2
> tempdf
     Group.1         x
1      April 137180053
2     August  94334296
3   December 183126398
4   February  92842024
5    January 120928179
6       July 134473616
7       June  93331656
8      March  87748525
9        May 134786200
10  November 113032586
11   October 136049595
12 September 128707888
`

> tempdf$Group.1[tempdf$Group.1=="January"]<-0
> tempdf$Group.1[tempdf$Group.1=="February"]<-0
> tempdf$Group.1[tempdf$Group.1=="March"]<-0
> tempdf$Group.1[tempdf$Group.1=="April"]<-0
> tempdf$Group.1[tempdf$Group.1=="May"]<-0
> tempdf$Group.1[tempdf$Group.1=="June"]<-0
>
>
> tempdf$Group.1[tempdf$Group.1=="July"]<-1
> tempdf$Group.1[tempdf$Group.1=="August"]<-1
> tempdf$Group.1[tempdf$Group.1=="September"]<-1
> tempdf$Group.1[tempdf$Group.1=="October"]<-1
> tempdf$Group.1[tempdf$Group.1=="November"]<-1
> tempdf$Group.1[tempdf$Group.1=="December"]<-1
>
> tempo2 <- aggregate(tempdf$x, by=list(tempdf$Group.1), FUN=sum)
>
> tempo2$Group.1[tempo2$Group.1==0]<-"Semester 1"
> tempo2$Group.1[tempo2$Group.1==1]<-"Semester 2"
>
> ws = unlist(subset(tempo2, select=c(x)))
> cat = unlist(subset(tempo2, select=c(Group.1)))
> barplot(ws,names.arg=cat,xlab="Semesters", ylab="Weekly Sales", main="For Year 2011 - Sem wise")
```

```
> #Visualization Semester Wise for Year 2012
>
> tempo = subset(walmart, format(Data, format = "%Y")=="2012", select=c(Weekly_Sales,Data))
> tempo = tempo[with(tempo, order(Data)),]
> tempo2 <- aggregate(tempo$Weekly_Sales, by=list(months(tempo$Data)), FUN=sum)
> subset(tempo2, select=c(x))
           x
1  135418590
2  141961415
3   95968564
4  123939050
5  136256431
6  142677508
7  137168161
8   94716403
9   90666527
10 132315485
> subset(tempo2, select=c(Group.1))
     Group.1
1      April
2     August
3   February
4    January
5       July
6       June
7      March
8        May
9    October
10 September
>
> tempdf = tempo2
> tempdf
     Group.1         x
1      April 135418590
2     August 141961415
3   February  95968564
4    January 123939050
5       July 136256431
6       June 142677508
7      March 137168161
8        May  94716403
9    October  90666527
10 September 132315485
>
> tempdf$Group.1[tempdf$Group.1=="January"]<-0
> tempdf$Group.1[tempdf$Group.1=="February"]<-0
> tempdf$Group.1[tempdf$Group.1=="March"]<-0
> tempdf$Group.1[tempdf$Group.1=="April"]<-0
> tempdf$Group.1[tempdf$Group.1=="May"]<-0
> tempdf$Group.1[tempdf$Group.1=="June"]<-0
>
>
>
> tempdf$Group.1[tempdf$Group.1=="July"]<-1
> tempdf$Group.1[tempdf$Group.1=="August"]<-1
> tempdf$Group.1[tempdf$Group.1=="September"]<-1
> tempdf$Group.1[tempdf$Group.1=="October"]<-1
> tempdf$Group.1[tempdf$Group.1=="November"]<-1
> tempdf$Group.1[tempdf$Group.1=="December"]<-1
>
> tempo2 <- aggregate(tempdf$x, by=list(tempdf$Group.1), FUN=sum)
>
> tempo2$Group.1[tempo2$Group.1==0]<-"Semester 1"
> tempo2$Group.1[tempo2$Group.1==1]<-"Semester 2"
>
> ws = unlist(subset(tempo2, select=c(x)))
> cat = unlist(subset(tempo2, select=c(Group.1)))
> barplot(ws,names.arg=cat,xlab="Semesters", ylab="Weekly Sales", main="For Year 2012 - Sem wise")
>
```

```
> df<- read.csv("C:/Users/canis/Downloads/1572585064_walmart_store_sales/Walmart_Store_sales.csv")
> head(df);
  Store       Date Weekly_Sales Holiday_Flag Temperature Fuel_Price CPI Unemployment
1     1   5/2/2010      1643691            0          42        2.6 211          8.1
2     1  12/2/2010      1641957            1          39        2.5 211          8.1
3     1 19-02-2010      1611968            0          40        2.5 211          8.1
4     1 26-02-2010      1409728            0          47        2.6 211          8.1
5     1   5/3/2010      1554807            0          46        2.6 211          8.1
6     1  12/3/2010      1439542            0          58        2.7 211          8.1
> lapply(df,class)
$Store
[1] "integer"

$Date
[1] "character"

$Weekly_Sales
[1] "numeric"

$Holiday_Flag
[1] "integer"

$Temperature
[1] "numeric"

$Fuel_Price
[1] "numeric"

$CPI
[1] "numeric"

$Unemployment
[1] "numeric"

> install.packages("datetime")
Error in install.packages : Updating loaded packages
> library(datetime)
> df[is.na(df)] <- 0;
> df$Data<-as.Date(df$Date,format='%d-%m-%Y')
> df$Date=as.Date(df$Date,format='%d-%m-%Y')
> df_c<-df[1:100,]
> df = df[with(df, order(Data)),]
> date_rs<-vector(mode="list",length=nrow(df))
>
> for(row in 1:nrow(df)){
+   date <- df[row, "Data"]
+   x <- as.Date("2010-02-05")

> #Ho:Temperature, CPI, Fuel Price, Unemployment and date are not influencing weekly sales.
>
> #input<-df[1:nrow(df),c("Store"==1,"Weekly_Sales","Temperature","Fuel_Price","CPI","Unemployment")]
> input = subset(subset(df, Store==1), select=c('Weekly_Sales','Temperature','Fuel_Price','CPI','Unemployment'))
>
> cor(input)
             Weekly_Sales Temperature Fuel_Price   CPI Unemployment
Weekly_Sales        1.000       -0.22       0.12  0.23       -0.098
Temperature        -0.223        1.00       0.23  0.12       -0.181
Fuel_Price          0.125        0.23       1.00  0.76       -0.514
CPI                 0.225        0.12       0.76  1.00       -0.813
Unemployment       -0.098       -0.18      -0.51 -0.81        1.000
> plot(input)
>
> #Method 1:VIF METHOD
> #Check multicollinearity
>
> vifstep(input[,-1], th=10)
No variable from the 4 input variables has collinearity problem.

The linear correlation coefficients ranges between:
min correlation ( CPI ~ Temperature ):  0.12
max correlation ( Unemployment ~ CPI ):  -0.81

---------- VIFs of the remained variables --------
     Variables VIF
1  Temperature 1.1
2   Fuel_Price 2.7
3          CPI 5.7
4 Unemployment 3.3
>
> model<-lm(Weekly_Sales~Temperature+Fuel_Price+CPI+Unemployment, data = input)
> print(model)

Call:
lm(formula = Weekly_Sales ~ Temperature + Fuel_Price + CPI +
    Unemployment, data = input)

Coefficients:
 (Intercept)   Temperature    Fuel_Price           CPI  Unemployment
    -2727200         -2426        -31637         17872         90632
```

```
> summary(model)

Call:
lm(formula = Weekly_Sales ~ Temperature + Fuel_Price + CPI +
    Unemployment, data = input)

Residuals:
    Min      1Q  Median      3Q     Max
-316968  -85750  -15239   51482  844800

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2727200    1759519   -1.55   0.1234
Temperature     -2426        918   -2.64   0.0091 **
Fuel_Price     -31637      47552   -0.67   0.5070
CPI             17872       6807    2.63   0.0096 **
Unemployment    90632      58925    1.54   0.1263
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 148000 on 138 degrees of freedom
Multiple R-squared:  0.129,     Adjusted R-squared:  0.104
F-statistic: 5.11 on 4 and 138 DF,  p-value: 0.000714
```

```
> names(model)
 [1] "coefficients"  "residuals"     "effects"       "rank"          "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
> model$fitted.values#predicted values for my dataset
      3       4       7       8      11      12      13      15      16      17      20      21      24      25      26      28      29
1607224 1590025 1563845 1567529 1492574 1495691 1489043 1467888 1469744 1467156 1472065 1468944 1466619 1468992 1474076 1464728 1465818
     30      33      34      37      38      39      42      43      46      47      48      50      51      52      55      56      59
1470067 1477983 1480829 1519672 1513861 1515529 1557681 1524484 1549733 1543150 1550807 1573395 1557996 1565305 1550148 1541215 1542954
     60      63      64      65      67      68      69      72      73      76      77      78      81      82      85      86      87
1530174 1525406 1522211 1527449 1521880 1536950 1512182 1486772 1496662 1515430 1520840 1518381 1518206 1525133 1550000 1567895 1567983
     89      90      91      94      95      98      99     100     102     103     104     107     108     111     112     113     115
1592657 1612514 1607845 1630072 1641886 1677693 1691271 1702194 1647785 1635689 1636294 1657226 1630629 1617066 1613835 1609257 1588346
    116     117     120     121     124     125     126     128     129     130     133     134     135     137     138     139     142
1595357 1597172 1597701 1583206 1583669 1588207 1575079 1575002 1565408 1557089 1548322 1566604 1561555 1577392 1593211 1583454 1582985
    143       1       2       5       6       9      10      14      18      19      22      23      27      31      32      35      36
1583213 1596168 1608754 1588860 1560682 1511146 1497290 1473945 1473256 1480056 1472800 1472640 1463987 1480482 1486332 1509411 1529113
     40      41      44      45      49      53      54      57      58      61      62      66      70      71      74      75      79
1543681 1541186 1559818 1560156 1541447 1576364 1596197 1547358 1561698 1553568 1533670 1545419 1498010 1495195 1521136 1523511 1507715
     80      83      84      88      92      93      96      97     101     105     106     109     110     114     118     119     122
1512423 1526275 1555711 1592666 1640106 1634064 1675576 1692519 1646725 1630188 1651000 1623027 1630905 1583713 1580003 1587215 1583373
    123     127     131     132     136     140     141
1584115 1564400 1548554 1548843 1552611 1576485 1594054
> input$Weekly_Sales
  [1] 1611968 1409728 1472516 1404430 1466058 1391256 1425101 1494252 1399662 1432070 1503284 1422712 1448939 1385065 1371987 1508238
 [17] 1513080 1449143 1430379 1351791 1459409 1345454 1384209 1483784 1955624 1891035 2387950 1367320 1391014 1327405 1316899 1686843
 [33] 1456800 1576818 1541102 1559889 1564820 1455091 1604776 1428218 1466047 1532115 1438830 1455120 1396927 1352220 1530761 1464693
 [49] 1514260 1380020 1394562 1493526 1502563 1445249 1539484 2033321 1881177 2270189 1497463 1459601 1394394 1319326 1819870 1539388
 [65] 1677473 1511068 1649605 1621032 1521578 1468928 1595902 1555445 1630607 1527846 1540421 1527014 1497955 1439124 1597868 1494122
 [81] 1582083 1517429 1506126 1437059 1508069 1493660 1643691 1641957 1554807 1439542 1594968 1545419 1603955 1615525 1542561 1492418
 [97] 1546074 1605492 1540164 1507461 1453330 1508240 1551659 1494479 1548034 1682614 1444732 1606630 1649615 1636263 1553192 1495065
[113] 1614259 1629391 1635078 1588948 1488538 1534850 1624384 1525147 1550229 1540471 1630990 1697230 1594939 1584084 1799682 1550370
[129] 1636340 1802477 1688421 1675431 1899677 1684520 1611096 1624478 1697231 1769854 1631136 1592410 1661767 1670786 1573073
> model$residuals
      3       4       7       8      11      12      13      15      16      17      20      21      24      25      26      28      29
   4744 -180297  -91329 -163099  -26516 -104435  -63942   26364  -70082  -35086   31219  -46232  -17680  -83927 -102090   43510   47262
     30      33      34      37      38      39      42      43      46      47      48      50      51      52      55      56      59
 -20924  -47604 -129038  -60263 -168407 -131320  -73897  431140  341302  844800 -183487 -182381 -230591 -248406  136695  -84415   33864
     60      63      64      65      67      68      69      72      73      76      77      78      81      82      85      86      87
  10929   34483   42609  -72358   82895 -108732  -46135   45343  -57832  -60310 -123914 -166161   12556  -60440  -35740 -187875 -173421
     89      90      91      94      95      98      99     100     102     103     104     107     108     111     112     113     115
 -99131 -109951 -162596  -90588  391435  203483  578918 -204731 -188184 -241295 -316968  162644  -91241   60407 -102767   40348   32686
    116     117     120     121     124     125     126     128     129     130     133     134     135     137     138     139     142
 -73779 -128244   -1799  -27761   46938  -60361  -34657  -47988  -67453 -117965   49546  -72482   20529  -59963  -87085 -146395  -74917
    143       1       2       5       6       9      10      14      18      19      22      23      27      31      32      35      36
 -89553   47522   33203  -34054 -121140   83822   48128  130010  142269   62506   19619   73435  141505   59682   21129  -56082  -20873
     40      41      44      45      49      53      54      57      58      61      62      66      70      71      74      75      79
   7978  -46707  -11784  122458  -96714   30265   53418   88905   -8506  -58503   80589   83972  137069   93753  -32598   11338  116669
     80      83      84      88      92      93      96      97     101     105     106     109     110     114     118     119     122
  12724   23955  -15239   38323   57124  -39125  -91492  107163  -96355    6152  151477   65394   44526  315964  104517   23881   41104
    123     127     131     132     136     140     141
 113116  205454   82582   43567  109157   94301  -20981
>
```

```
> plot(model)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
> y<-coef(model)[1]
>
> xtemp<-coef(model)[2]
> xfuel<-coef(model)[3]
> xcpi<-coef(model)[4]
> xunemp<-coef(model)[5]
>
> #weekly_sales = -2727200 - 2426temp -31637fuel + 17872cpi + 90632umep
>
> #Method 2: Significance Method
>
> model1<-lm(Weekly_Sales~Temperature+Fuel_Price+CPI+Unemployment, data = input)
> print(model1)

Call:
lm(formula = Weekly_Sales ~ Temperature + Fuel_Price + CPI +
    Unemployment, data = input)

Coefficients:
 (Intercept)   Temperature    Fuel_Price           CPI  Unemployment
    -2727200         -2426        -31637         17872         90632

> summary(model1)

Call:
lm(formula = Weekly_Sales ~ Temperature + Fuel_Price + CPI +
    Unemployment, data = input)

Residuals:
    Min      1Q  Median      3Q     Max
-316968  -85750  -15239   51482  844800

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2727200    1759519   -1.55   0.1234
Temperature     -2426        918   -2.64   0.0091 **
Fuel_Price     -31637      47552   -0.67   0.5070
CPI             17872       6807    2.63   0.0096 **
Unemployment    90632      58925    1.54   0.1263
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 148000 on 138 degrees of freedom
Multiple R-squared:  0.129,     Adjusted R-squared:  0.104
F-statistic: 5.11 on 4 and 138 DF,  p-value: 0.000714
```

```
> names(model1)
 [1] "coefficients"  "residuals"     "effects"       "rank"          "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
> model1$fitted.values #predicted values for my dataset
      3       4       7       8      11      12      13      15      16      17      20      21      24      25      26      28      29
1607224 1590025 1563845 1567529 1492574 1495691 1489043 1467888 1469744 1467156 1472065 1468944 1466619 1468992 1474076 1464728 1465818
     30      33      34      37      38      39      42      43      46      47      48      50      51      52      55      56      59
1470067 1477983 1480829 1519672 1513861 1515529 1557681 1524484 1549733 1543150 1550807 1573395 1557996 1565305 1550148 1541215 1542954
     60      63      64      65      67      68      69      72      73      76      77      78      81      82      85      86      87
1530174 1525406 1522211 1527449 1521880 1536950 1512182 1486772 1496662 1515430 1520840 1518381 1518206 1525133 1550000 1567895 1567983
     89      90      91      94      95      98      99     100     102     103     104     107     108     111     112     113     115
1592657 1612514 1607845 1630072 1641886 1677693 1691271 1702194 1647785 1635689 1636294 1657226 1630629 1617066 1613835 1609257 1588346
    116     117     120     121     124     125     126     128     129     130     133     134     135     137     138     139     142
1595357 1597172 1597701 1583206 1583669 1588207 1575079 1575002 1565408 1557089 1548322 1566604 1561555 1577392 1593211 1583454 1582985
    143       1       2       5       6       9      10      14      18      19      22      23      27      31      32      35      36
1583213 1596168 1608754 1588860 1560682 1511146 1497290 1473945 1473256 1480056 1472800 1472640 1463987 1480482 1486332 1509411 1529113
     40      41      44      45      49      53      54      57      58      61      62      66      70      71      74      75      79
1543681 1541186 1559818 1560156 1541447 1576364 1596197 1547358 1561698 1553568 1533670 1545419 1498010 1495195 1521136 1523511 1507715
     80      83      84      88      92      93      96      97     101     105     106     109     110     114     118     119     122
1512423 1526275 1555711 1592666 1640106 1634064 1675576 1692519 1646725 1630188 1651000 1623027 1630905 1583713 1580003 1587215 1583373
    123     127     131     132     136     140     141
1584115 1564400 1548554 1548843 1552611 1576485 1594054
> input$Weekly_Sales
  [1] 1611968 1409728 1472516 1404430 1466058 1391256 1425101 1494252 1399662 1432070 1503284 1422712 1448939 1385065 1371987 1508238
 [17] 1513080 1449143 1430379 1351791 1459409 1345454 1384209 1483784 1955624 1891035 2387950 1367320 1391014 1327405 1316899 1686843
 [33] 1456800 1576818 1541102 1559889 1564820 1455091 1604776 1428218 1466047 1532115 1438830 1455120 1396927 1352220 1530761 1464693
 [49] 1514260 1380020 1394562 1493526 1502563 1445249 1539484 2033321 1881177 2270189 1497463 1459601 1394394 1319326 1819870 1539388
 [65] 1677473 1511068 1649605 1621032 1521578 1468928 1595902 1555445 1630607 1527846 1540421 1527014 1497955 1439124 1597868 1494122
 [81] 1582083 1517429 1506126 1437059 1508069 1493660 1643691 1641957 1554807 1439542 1594968 1545419 1603955 1615525 1542561 1492418
 [97] 1546074 1605492 1540164 1507461 1453330 1508240 1551659 1494479 1548034 1682614 1444732 1606630 1649615 1636263 1553192 1495065
[113] 1614259 1629391 1635078 1588948 1488538 1534850 1624384 1525147 1550229 1540471 1630990 1697230 1594939 1584084 1799682 1550370
[129] 1636340 1802477 1688421 1675431 1899677 1684520 1611096 1624478 1697231 1769854 1631136 1592410 1661767 1670786 1573073
> model1$residuals
      3       4       7       8      11      12      13      15      16      17      20      21      24      25      26      28      29
   4744 -180297  -91329 -163099  -26516 -104435  -63942   26364  -70082  -35086   31219  -46232  -17680  -83927 -102090   43510   47262
     30      33      34      37      38      39      42      43      46      47      48      50      51      52      55      56      59
 -20924  -47604 -129038  -60263 -168407 -131320  -73897  431140  341302  844800 -183487 -182381 -230591 -248406  136695  -84415   33864
     60      63      64      65      67      68      69      72      73      76      77      78      81      82      85      86      87
  10929   34483   42609  -72358   82895 -108732  -46135   45343  -57832  -60310 -123914 -166161   12556  -60440  -35740 -187875 -173421
     89      90      91      94      95      98      99     100     102     103     104     107     108     111     112     113     115
 -99131 -109951 -162596  -90588  391435  203483  578918 -204731 -188184 -241295 -316968  162644  -91241   60407 -102767   40348   32686
    116     117     120     121     124     125     126     128     129     130     133     134     135     137     138     139     142
 -73779 -128244   -1799  -27761   46938  -60361  -34657  -47988  -67453 -117965   49546  -72482   20529  -59963  -87085 -146395  -74917
    143       1       2       5       6       9      10      14      18      19      22      23      27      31      32      35      36
 -89553   47522   33203  -34054 -121140   83822   48128  130010  142269   62506   19619   73435  141505   59682   21129  -56082  -20873
     40      41      44      45      49      53      54      57      58      61      62      66      70      71      74      75      79
   7978  -46707  -11784  122458  -96714   30265   53418   88905   -8506  -58503   80589   83972  137069   93753  -32598   11338  116669
     80      83      84      88      92      93      96      97     101     105     106     109     110     114     118     119     122
  12724   23955  -15239   38323   57124  -39125  -91492  107163  -96355    6152  151477   65394   44526  315964  104517   23881   41104
    123     127     131     132     136     140     141
 113116  205454   82582   43567  109157   94301  -20981

> plot(model1)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
> vifstep(input[,c(-3,-4)], th=10)
No variable from the 3 input variables has collinearity problem.

The linear correlation coefficients ranges between:
min correlation ( Unemployment ~ Weekly_Sales ):  -0.09795539
max correlation ( Temperature ~ Weekly_Sales ):  -0.2227006

---------- VIFs of the remained variables --------
     Variables      VIF
1 Weekly_Sales 1.074504
2  Temperature 1.100114
3 Unemployment 1.055682
>
```

```
> #removed Fuel_Price and Unemployment due to high p values.
>
> model2<- lm(weekly_Sales~Temperature+CPI, data = input)
> summary(model2)

Call:
lm(formula = Weekly_Sales ~ Temperature + CPI, data = input)

Residuals:
    Min      1Q  Median      3Q     Max
-312205  -85704   -9198   57222  830489

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -233190     616327  -0.378  0.70574
Temperature    -2769        877  -3.157  0.00195 **
CPI             9156       2872   3.187  0.00177 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 147900 on 140 degrees of freedom
Multiple R-squared:  0.1139,    Adjusted R-squared:  0.1012
F-statistic: 8.998 on 2 and 140 DF,  p-value: 0.0002107
```

```
> model2$fitted.values #predicted values for my dataset
      3       4       7       8      11      12      13      15      16      17      20      21      24      25      26      28      29
1590733 1572461 1549497 1556354 1510335 1513979 1506409 1485525 1483489 1474974 1469912 1468222 1469335 1469987 1475430 1463610 1464249
     30      33      34      37      38      39      42      43      46      47      48      50      51      52      55      56      59
1467878 1476079 1480003 1520085 1513100 1514144 1564445 1526852 1564358 1557461 1568258 1604817 1584280 1588247 1560406 1547700 1555661
     60      63      64      65      67      68      69      72      73      76      77      78      81      82      85      86      87
1537869 1540605 1537381 1541577 1534659 1554731 1524686 1495994 1504682 1493420 1501650 1499275 1492366 1498433 1523458 1537986 1530280
     89      90      91      94      95      98      99     100     102     103     104     107     108     111     112     113     115
1558195 1581900 1575453 1592393 1600498 1630578 1642372 1653446 1645691 1631093 1631531 1659461 1628361 1612886 1610274 1606306 1603629
    116     117     120     121     124     125     126     128     129     130     133     134     135     137     138     139     142
1610518 1609710 1602269 1583212 1577476 1580504 1562904 1585130 1576071 1569947 1564772 1585903 1579290 1597106 1613056 1597691 1624212
    143       1       2       5       6       9      10      14      18      19      22      23      27      31      32      35      36
1621087 1582378 1594235 1573100 1542119 1524586 1512837 1491723 1476843 1480123 1476665 1476801 1461933 1478651 1485299 1505746 1528470
     40      41      44      45      49      53      54      57      58      61      62      66      70      71      74      75      79
1544760 1542787 1567784 1574631 1568699 1595953 1615620 1559527 1578830 1569945 1548111 1563661 1507959 1505492 1500079 1500156 1486489
     80      83      84      88      92      93      96      97     101     105     106     109     110     114     118     119     122
1489514 1499379 1532719 1562086 1609015 1599051 1633855 1649906 1642729 1626043 1650514 1620022 1628047 1599180 1587166 1592588 1581211
    123     127     131     132     136     140     141
1580263 1572439 1560473 1563486 1570905 1620370 1637595
> input$weekly_Sales
  [1] 1611968 1409728 1472516 1404430 1466058 1391256 1425101 1494252 1399662 1432070 1503284 1422712 1448939 1385065 1371987 1508238
 [17] 1513080 1449143 1430379 1351791 1459409 1345454 1384209 1483784 1955624 1891035 2387950 1367320 1391014 1327405 1316899 1686843
 [33] 1456800 1576818 1541102 1559889 1564820 1455091 1604776 1428218 1466047 1532115 1438830 1455120 1396927 1352220 1530761 1464693
 [49] 1514260 1380020 1394562 1493526 1502563 1445249 1539484 2033321 1881177 2270189 1497463 1459601 1394394 1319326 1819870 1539388
 [65] 1677473 1511068 1649605 1621032 1521578 1468928 1595902 1555445 1630607 1527846 1540421 1527014 1497955 1439124 1597868 1494122
 [81] 1582083 1517429 1506126 1437059 1508069 1493660 1643691 1641957 1554807 1439542 1594968 1545419 1603955 1615525 1542561 1492418
 [97] 1546074 1605492 1540164 1507461 1453330 1508240 1551659 1494479 1548034 1682614 1444732 1606630 1649615 1636263 1553192 1495065
[113] 1614259 1629391 1635078 1588948 1488538 1534850 1624384 1525147 1550229 1540471 1630990 1697230 1594939 1584084 1799682 1550370
[129] 1636340 1802477 1688421 1675431 1899677 1684520 1611096 1624478 1697231 1769854 1631136 1592410 1661767 1670786 1573073
> model2$residuals
         3          4          7          8         11         12         13         15         16         17         20
  21234.821 -162733.645  -76980.753 -151924.070  -44276.305 -122722.467  -81307.997    8726.476  -83827.176  -42904.385   33371.654
        21         24         25         26         28         29         30         33         34         37         38
 -45509.979  -20396.117  -84922.179 -103443.247   44627.624   48831.385  -18735.524  -45700.334 -128212.060  -60675.867 -167645.774
        39         42         43         46         47         48         50         51         52         55         56
-129935.054  -80660.893  428771.932  326677.377  830488.849 -200938.153 -213802.922 -256874.478 -271348.013  126437.146  -90900.161
        59         60         63         64         65         67         68         69         72         73         76
  21157.340    3233.173   19284.376   27439.030  -86486.790   70116.494 -126512.376  -58639.176   36120.663  -65851.746  -38300.497
        77         78         81         82         85         86         87         89         90         91         94
-104723.435 -147055.521   38395.585  -33739.910   -9198.192 -157965.828 -135718.449  -64669.203  -79337.633 -130203.719  -52909.684
        95         98         99        100        102        103        104        107        108        111        112
 432822.514  250598.259  627816.706 -155983.532 -186089.335 -236699.393 -312205.279  160409.381  -88973.442   64586.596  -99206.175
       113        115        116        117        120        121        124        125        126        128        129
  43299.067   17402.665  -88940.382 -140781.668   -6366.750  -27767.409   53130.675  -52658.316  -22482.430  -58115.621  -78116.116
       130        133        134        135        137        138        139        142        143          1          2
-130823.372   33095.618  -91780.748    2793.550  -79676.971 -106930.437 -160632.088 -116143.727 -127427.449   61312.502   47722.378
         5          6          9         10         14         18         19         22         23         27         31
 -18293.753 -102577.663   70382.435   32581.975  112232.262  138682.062   62438.010   15753.361   69272.780  143558.465   61512.090
        32         35         36         40         41         44         45         49         53         54         57
  22161.854  -52416.291  -20230.504    6899.084  -48307.149  -19750.366  107982.810 -123967.159   10676.871   33995.361   76736.065
        58         61         62         66         70         71         74         75         79         80         83
 -25638.541  -74880.109   66148.539   65730.266  127119.151   83456.018  -11540.979   34694.025  137894.295   35633.302   50850.716
        84         88         92         93         96         97        101        105        106        109        110
   7751.942   68903.860   88214.351  -4112.284  -49770.957  149776.084  -92359.375   10296.504  151962.945   68398.808   47384.568
       114        118        119        122        123        127        131        132        136        140        141
 300496.498   97353.714   18508.052   43266.106  116967.803  197415.620   70662.821   28923.558   90862.829   50415.638  -64521.853
```

```
> plot(model2)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
> vifstep(input[,c(-3,-4)], th=10)
No variable from the 3 input variables has collinearity problem.

The linear correlation coefficients ranges between:
min correlation ( Unemployment ~ Weekly_Sales ):  -0.09795539
max correlation ( Temperature ~ Weekly_Sales ):  -0.2227006

---------- VIFs of the remained variables --------
     Variables      VIF
1 Weekly_Sales 1.074504
2  Temperature 1.100114
3 Unemployment 1.055682
> 
```

```
> y1<-coef(model1)[1]
> xtemp1<-coef(model1)[2]
> xcpi1<-coef(model1)[3]
> y1<-coef(model2)[1]
> xtemp1<-coef(model2)[2]
> xcpi1<-coef(model2)[3]
> 
```

```
> #H0:there is no correlation amongst residuals,
> #i.e. they are independent
> #in case of regression, we need high p value so that we cannot reject the null
>
> ##Method 3 :step
>
> model3<-step(model)
Start:  AIC=3409.07
Weekly_Sales ~ Temperature + Fuel_Price + CPI + Unemployment

               Df  Sum of Sq        RSS    AIC
- Fuel_Price    1 9.6512e+09 3.0185e+12 3407.5
<none>                       3.0088e+12 3409.1
- Unemployment  1 5.1580e+10 3.0604e+12 3409.5
- CPI           1 1.5030e+11 3.1591e+12 3414.0
- Temperature   1 1.5241e+11 3.1612e+12 3414.1

Step:  AIC=3407.53
Weekly_Sales ~ Temperature + CPI + Unemployment

               Df  Sum of Sq        RSS    AIC
<none>                       3.0185e+12 3407.5
- Unemployment  1 4.2856e+10 3.0613e+12 3407.5
- Temperature   1 1.8633e+11 3.2048e+12 3414.1
- CPI           1 1.9681e+11 3.2153e+12 3414.6
> 
```

```
> summary(model3)

Call:
lm(formula = Weekly_Sales ~ Temperature + CPI + Unemployment,
    data = input)

Residuals:
    Min      1Q  Median      3Q     Max
-311770  -85435  -10998   55936  841075

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2048494.5  1430740.6  -1.432  0.15445
Temperature     -2587.7      883.4  -2.929  0.00397 **
CPI             14730.4     4893.0   3.011  0.00310 **
Unemployment    78679.8    56007.2   1.405  0.16231
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 147400 on 139 degrees of freedom
Multiple R-squared:  0.1263,    Adjusted R-squared:  0.1074
F-statistic: 6.698 on 3 and 139 DF,  p-value: 0.0002957
```

```
> input$Weekly_Sales
  [1] 1611968 1409728 1472516 1404430 1466058 1391256 1425101 1494252 1399662 1432070 1503284 1422712 1448939 1385065 1371987 1508238
 [17] 1513080 1449143 1430379 1351791 1459409 1345454 1384209 1483784 1955624 1891035 2387950 1367320 1391014 1327405 1316899 1686843
 [33] 1456800 1576818 1541102 1559889 1564820 1455091 1604776 1428218 1466047 1532115 1438830 1455120 1396927 1352220 1530761 1464693
 [49] 1514260 1380020 1394562 1493526 1502563 1445249 1539484 2033321 1881177 2270189 1497463 1459601 1394394 1319326 1819870 1539388
 [65] 1677473 1511068 1649605 1621032 1521578 1468928 1595902 1555445 1630607 1527846 1540421 1527014 1497955 1439124 1597868 1494122
 [81] 1582083 1517429 1506126 1437059 1508069 1493660 1643691 1641957 1554807 1439542 1594968 1545419 1603955 1615525 1542561 1492418
 [97] 1546074 1605492 1540164 1507461 1453330 1508240 1551659 1494479 1548034 1682614 1444732 1606630 1649615 1636263 1553192 1495065
[113] 1614259 1629391 1635078 1588948 1488538 1534850 1624384 1525147 1550229 1540471 1630990 1697230 1594939 1584084 1799682 1550370
[129] 1636340 1802477 1688421 1675431 1899677 1684520 1611096 1624478 1697231 1769854 1631136 1592410 1661767 1670786 1573073
> model3$fitted.values
      3       4       7       8      11      12      13      15      16      17      20      21      24      25      26      28      29
1598340 1581452 1559347 1564536 1494813 1497913 1490532 1470693 1470517 1464285 1462994 1460703 1458620 1460062 1465980 1456598 1456972
     30      33      34      37      38      39      42      43      46      47      48      50      51      52      55      56      59
1460141 1467529 1471658 1514467 1508232 1509502 1556393 1520388 1553321 1546875 1556964 1583902 1566992 1572983 1553450 1543353 1555898
     60      63      64      65      67      68      69      72      73      76      77      78      81      82      85      86      87
1540735 1542843 1540867 1545827 1541436 1558774 1529274 1499337 1508031 1521204 1529273 1527431 1522121 1528186 1553474 1569118 1563983
     89      90      91      94      95      98      99     100     102     103     104     107     108     111     112     113     115
1586499 1609645 1604612 1623802 1632902 1665409 1677532 1688982 1643178 1630111 1631095 1659340 1631578 1620665 1618685 1615436 1597727
    116     117     120     121     124     125     126     128     129     130     133     134     135     137     138     139     142
1604498 1604075 1597890 1580094 1574843 1577922 1561723 1564503 1556090 1550420 1546183 1566756 1561400 1579758 1595898 1582772 1583942
    143       1       2       5       6       9      10      14      18      19      22      23      27      31      32      35      36
1581135 1589342 1601323 1582238 1553472 1510180 1497980 1476501 1467758 1472551 1466230 1465647 1454198 1469987 1475977 1500190 1521890
     40      41      44      45      49      53      54      57      58      61      62      66      70      71      74      75      79
1538408 1536857 1557770 1563297 1549822 1582467 1603131 1556182 1575998 1567456 1548515 1567503 1512221 1508495 1526334 1526980 1515859
     80      83      84      88      92      93      96      97     101     105     106     109     110     114     118     119     122
1519063 1529464 1561018 1588222 1636971 1628651 1665602 1682129 1639310 1626542 1649987 1625088 1633892 1593108 1583338 1588738 1578239
    123     127     131     132     136     140     141
1577367 1552393 1541618 1544488 1554387 1578843 1596174
```

```
> model3$residuals
          3           4           7           8          11          12          13          15          16          17
 13627.8653 -171724.3139  -86831.7098 -160106.4951  -28754.8034 -106656.4848  -65431.1771   23558.8039  -70854.6465  -32215.5360
         20          21          24          25          26          28          29          30          33          34
 40290.4563  -37991.0709   -9681.0712  -74996.4303  -93993.4135   51640.2281   56108.4059  -10998.4481  -37149.8850 -119866.4715
         37          38          39          42          43          46          47          48          50          51
-55057.4294 -162777.8919 -125292.4031  -72609.0734  435236.3130  337713.8253  841075.3033 -189644.1685 -192887.8979 -239586.2587
         52          55          56          59          60          63          64          65          67          68
-256083.4527  133392.6182  -86552.2486   20919.6630     366.9291   17046.1151   23952.3521  -90736.4111   63339.2472 -130555.6936
         69          72          73          76          77          78          81          82          85          86
 -63227.3126   32777.7092  -69200.7298  -66083.5742 -132345.7472 -175210.8596    8640.7311  -63492.4093  -39213.9770 -189097.9616
         87          89          90          91          94          95          98          99         100         102
-169421.5366  -92972.8737 -107082.5875 -159362.6430  -84317.8095  400419.0005  215767.9630  592657.4031 -191518.9398 -183577.1906
        103         104         107         108         111         112         113         115         116         117
-235717.4128 -311769.8385  160530.3291  -92190.6574   56807.5664 -107616.6472   34168.3901   23305.1208  -82919.7561 -135146.4584
        120         121         124         125         126         128         129         130         133         134
  -1988.4818  -24649.8949   55763.6836  -50076.5119  -21301.3715  -37489.3216  -58135.3697 -111296.1626   51684.7149  -72633.6415
        135         137         138         139         142         143           1           2           5           6
  20683.5799  -62328.6681  -89772.2439 -145712.7717  -75873.1623  -87475.7392   54349.1580   40634.5171  -27430.9052 -113929.9901
          9          10          14          18          19          22          23          27          31          32
  84788.1330   47438.8999  127454.3739  147766.5293   70010.4793   26187.6472   80426.7292  151293.3426   70176.5774   31483.8458
         35          36          40          41          44          45          49          53          54          57
 -46860.7982  -13649.8869   13251.1471  -42377.7666   -9736.1129  119317.0242 -105089.8691   24162.3188   46484.2501   80081.3742
         58          61          62          66          70          71          74          75          79          80
 -22806.7391  -72391.6838   65743.9238   61887.9073  122857.5087   80453.7747  -37795.8418    7869.7677  108524.8203    6084.0368
         83          84          88          92          93          96          97         101         105         106
  20765.4499  -20546.6771   42768.0407   60259.0236  -33711.6359  -81517.8038  117553.7390  -88940.3679    9797.7524  152490.0048
        109         110         114         118         119         122         123         127         131         132
  63332.4524   41539.6477  306568.4720  101181.5077   22358.0088   46238.7292  119864.3708  217460.9236   89517.3062   47922.2893
        136         140         141
 107379.8479   91942.7658  -23101.5082
> plot(model3)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
>
```

```
> y2<-coef(model3)[1]
> xtemp2<-coef(model3)[2]
> xcpi2<-coef(model3)[3]
> xumep2<-coef(model3)[4]
> #weeklysales=-2048494 -2588temp2+14730cpi2+78680unemp2
>
> #based on p value higher than 0.05 we eliminate Unemployment from the model. ('refer to model 2')
>
> #dwtest for all models
```

```
> library(lmtest)
Warning message:
package 'lmtest' was built under R version 4.0.4
> dwtest(model1)

        Durbin-Watson test

data:  model1
DW = 1.4067, p-value = 4.937e-05
alternative hypothesis: true autocorrelation is greater than 0

> dwtest(model2)

        Durbin-Watson test

data:  model2
DW = 1.3903, p-value = 6.363e-05
alternative hypothesis: true autocorrelation is greater than 0

> dwtest(model3)

        Durbin-Watson test

data:  model3
DW = 1.4031, p-value = 6.32e-05
alternative hypothesis: true autocorrelation is greater than 0

>
```

```
> #predict(model2, newdata=input)
> predict(model2)
      3       4       7       8      11      12      13      15      16      17      20      21      24      25      26      28      29
1590733 1572461 1549497 1556354 1510335 1513979 1506409 1485525 1483489 1474974 1469912 1468222 1469335 1469987 1475430 1463610 1464249
     30      33      34      37      38      39      42      43      46      47      48      50      51      52      55      56      59
1467878 1476079 1480003 1520085 1513100 1514144 1564445 1526852 1564358 1557461 1568258 1604817 1584280 1588247 1560406 1547700 1555661
     60      63      64      65      67      68      69      72      73      76      77      78      81      82      85      86      87
1537869 1540605 1537381 1541577 1534659 1554731 1524686 1495994 1504682 1493420 1501650 1499275 1492366 1498433 1523458 1537986 1530280
     89      90      91      94      95      98      99     100     102     103     104     107     108     111     112     113     115
1558195 1581900 1575453 1592393 1600498 1630578 1642372 1653446 1645691 1631093 1631531 1659461 1628361 1612886 1610274 1606306 1603629
    116     117     120     121     124     125     126     128     129     130     133     134     135     137     138     139     142
1610518 1609710 1602269 1583212 1577476 1580504 1562904 1585130 1576071 1569947 1564772 1585903 1579290 1597106 1613056 1597691 1624212
    143       1       2       5       6       9      10      14      18      19      22      23      27      31      32      35      36
1621087 1582378 1594235 1573100 1542119 1524586 1512837 1491723 1476843 1480123 1476665 1476801 1461933 1478651 1485299 1505746 1528470
     40      41      44      45      49      53      54      57      58      61      62      66      70      71      74      75      79
1544760 1542787 1567784 1574631 1568699 1595953 1615620 1559527 1578830 1569945 1548111 1563661 1507959 1505492 1500079 1500156 1486489
     80      83      84      88      92      93      96      97     101     105     106     109     110     114     118     119     122
1489514 1499379 1532719 1562086 1609015 1599051 1633855 1649906 1642729 1626043 1650514 1620022 1628047 1599180 1587166 1592588 1581211
    123     127     131     132     136     140     141
1580263 1572439 1560473 1563486 1570905 1620370 1637595
> model2$fitted.values
      3       4       7       8      11      12      13      15      16      17      20      21      24      25      26      28      29
1590733 1572461 1549497 1556354 1510335 1513979 1506409 1485525 1483489 1474974 1469912 1468222 1469335 1469987 1475430 1463610 1464249
     30      33      34      37      38      39      42      43      46      47      48      50      51      52      55      56      59
1467878 1476079 1480003 1520085 1513100 1514144 1564445 1526852 1564358 1557461 1568258 1604817 1584280 1588247 1560406 1547700 1555661
     60      63      64      65      67      68      69      72      73      76      77      78      81      82      85      86      87
1537869 1540605 1537381 1541577 1534659 1554731 1524686 1495994 1504682 1493420 1501650 1499275 1492366 1498433 1523458 1537986 1530280
     89      90      91      94      95      98      99     100     102     103     104     107     108     111     112     113     115
1558195 1581900 1575453 1592393 1600498 1630578 1642372 1653446 1645691 1631093 1631531 1659461 1628361 1612886 1610274 1606306 1603629
    116     117     120     121     124     125     126     128     129     130     133     134     135     137     138     139     142
1610518 1609710 1602269 1583212 1577476 1580504 1562904 1585130 1576071 1569947 1564772 1585903 1579290 1597106 1613056 1597691 1624212
    143       1       2       5       6       9      10      14      18      19      22      23      27      31      32      35      36
1621087 1582378 1594235 1573100 1542119 1524586 1512837 1491723 1476843 1480123 1476665 1476801 1461933 1478651 1485299 1505746 1528470
     40      41      44      45      49      53      54      57      58      61      62      66      70      71      74      75      79
1544760 1542787 1567784 1574631 1568699 1595953 1615620 1559527 1578830 1569945 1548111 1563661 1507959 1505492 1500079 1500156 1486489
     80      83      84      88      92      93      96      97     101     105     106     109     110     114     118     119     122
1489514 1499379 1532719 1562086 1609015 1599051 1633855 1649906 1642729 1626043 1650514 1620022 1628047 1599180 1587166 1592588 1581211
    123     127     131     132     136     140     141
1580263 1572439 1560473 1563486 1570905 1620370 1637595
> a=as.data.frame(predict(model2))
>
> view(a)
> names(a)
[1] "predict(model2)"
>

> #MAPE - mean absolute percentage Error
> model2$residuals
          3           4           7           8          11          12          13          15          16          17          20
  21234.821 -162733.645  -76980.753 -151924.070  -44276.305 -122722.467  -81307.997    8726.476  -83827.176  -42904.385   33371.654
         21          24          25          26          28          29          30          33          34          37          38
 -45509.979  -20396.117  -84922.179 -103443.247   44627.624   48831.385  -18735.524  -45700.334 -128212.060  -60675.867 -167645.774
         39          42          43          46          47          48          50          51          52          55          56
-129935.054  -80660.893  428771.932  326677.377  830488.849 -200938.153 -213802.922 -256874.478 -271348.013  126437.146  -90900.161
         59          60          63          64          65          67          68          69          72          73          76
  21157.340    3233.173   19284.376   27439.030  -86486.790   70116.494 -126512.376  -58639.176   36120.663  -65851.746  -38300.497
         77          78          81          82          85          86          87          89          90          91          94
-104723.435 -147055.521   38395.585  -33739.910   -9198.192 -157965.828 -135718.449  -64669.203  -79337.633 -130203.719  -52909.684
         95          98          99         100         102         103         104         107         108         111         112
 432822.514  250598.259  627816.706 -155983.532 -186089.335 -236699.393 -312205.279  160409.381  -88973.442   64586.596  -99206.175
        113         115         116         117         120         121         124         125         126         128         129
  43299.067   17402.665  -88940.382 -140781.668   -6366.750  -27767.409   53130.675  -52658.316  -22482.430  -58115.621  -78116.116
        130         133         134         135         137         138         139         142         143           1           2
-130823.372   33095.618  -91780.748    2793.550  -79676.971 -106930.437 -160632.088 -116143.727 -127427.449   61312.502   47722.378
          5           6           9          10          14          18          19          22          23          27          31
 -18293.753 -102577.663   70382.435   32581.975  112232.262  138682.062   62438.010   15753.361   69272.780  143558.465   61512.090
         32          35          36          40          41          44          45          49          53          54          57
  22161.854  -52416.291  -20230.504    6899.084  -48307.149  -19750.366  107982.810 -123967.159   10676.871   33995.361   76736.065
         58          61          62          66          70          71          74          75          79          80          83
 -25638.541  -74880.109   66148.539   65730.266  127119.151   83456.018  -11540.979   34694.025  137894.295   35633.302   50850.716
         84          88          92          93          96          97         101         105         106         109         110
   7751.942   68903.860   88214.351  -4112.284  -49770.957  149776.084  -92359.375   10296.504  151962.945   68398.808   47384.568
        114         118         119         122         123         127         131         132         136         140         141
 300496.498   97353.714   18508.052   43266.106  116967.803  197415.620   70662.821   28923.558   90862.829   50415.638  -64521.853
>
```

```
> model2$residuals/input$Weekly_Sales #PE
           3            4            7            8           11           12           13           15           16           17
 0.013173226 -0.115436235 -0.052278389 -0.108174903 -0.030200917 -0.088209831 -0.057054211  0.005840031 -0.059891011 -0.029959699
          20           21           24           25           26           28           29           30           33           34
 0.022199167 -0.031988197 -0.014076589 -0.061312766 -0.075396689  0.029589250  0.032272827 -0.012928693 -0.031949815 -0.094846065
          37           38           39           42           43           46           47           48           50           51
-0.041575640 -0.124601639 -0.093869519 -0.054361608  0.219250688  0.172750578  0.347783153 -0.146957663 -0.153702931 -0.193516219
          52           55           56           59           60           63           64           65           67           68
-0.206050691  0.074954909 -0.062397133  0.013417743  0.002097961  0.012362659  0.017534945 -0.059437388  0.043692399 -0.088580562
          69           72           73           76           77           78           81           82           85           86
-0.039998164  0.023575689 -0.045767561 -0.026321195 -0.074967016 -0.108751197  0.025082671 -0.023035475 -0.006074382 -0.114466310
          87           89           90           91           94           95           98           99          100          102
-0.097319779 -0.043299686 -0.052801543 -0.090090850 -0.034368460  0.212864858  0.133213570  0.276548212 -0.104165219 -0.127493276
         103          104          107          108          111          112          113          115          116          117
-0.169750745 -0.236640054  0.088143318 -0.057797938  0.038502322 -0.065653015  0.026248148  0.010735549 -0.058452731 -0.095839709
         120          121          124          125          126          128          129          130          133          134
-0.003989437 -0.017851751  0.032583372 -0.034465726 -0.014594986 -0.038058341 -0.052148515 -0.090904883  0.020712360 -0.061427865
         135          137          138          139          142          143            1            2            5            6
 0.001765741 -0.052507879 -0.070997003 -0.111778333 -0.077014874 -0.085312234  0.037301723  0.029064321 -0.011765934 -0.071257172
           9           10           14           18           19           22           23           27           31           32
 0.044127796  0.021082946  0.069972196  0.085843356  0.040476847  0.010555595  0.044805599  0.089417128  0.039938674  0.014701447
          35           36           40           41           44           45           49           53           54           57
-0.036066350 -0.013413319  0.004446262 -0.032323729 -0.012758356  0.064175618 -0.085806319  0.006645509  0.020608059  0.046897134
          58           61           62           66           70           71           74           75           79           80
-0.016507005 -0.050084860  0.040977641  0.040340381  0.077744988  0.052522802 -0.007753231  0.022604185  0.084890221  0.023363846
          83           84           88           92           93           96           97          101          105          106
 0.032802062  0.005032189  0.042246649  0.051975497 -0.002578333 -0.031419394  0.083223621 -0.059572476  0.006292400  0.084307821
         109          110          114          118          119          122          123          127          131          132
 0.040510523  0.028282014  0.158182953  0.057793148  0.011487864  0.026633858  0.068916845  0.111543439  0.043321238  0.018163387
         136          140          141
 0.054678430  0.030174803 -0.041016444

> abs(model2$residuals/input$Weekly_Sales) #APE
           3            4            7            8           11           12           13           15           16           17           20
0.013173226 0.115436235 0.052278389 0.108174903 0.030200917 0.088209831 0.057054211 0.005840031 0.059891011 0.029959699 0.022199167
          21           24           25           26           28           29           30           33           34           37           38
0.031988197 0.014076589 0.061312766 0.075396689 0.029589250 0.032272827 0.012928693 0.031949815 0.094846065 0.041575640 0.124601639
          39           42           43           46           47           48           50           51           52           55           56
0.093869519 0.054361608 0.219250688 0.172750578 0.347783153 0.146957663 0.153702931 0.193516219 0.206050691 0.074954909 0.062397133
          59           60           63           64           65           67           68           69           72           73           76
0.013417743 0.002097961 0.012362659 0.017534945 0.059437388 0.043692399 0.088580562 0.039998164 0.023575689 0.045767561 0.026321195
          77           78           81           82           85           86           87           89           90           91           94
0.074967016 0.108751197 0.025082671 0.023035475 0.006074382 0.114466310 0.097319779 0.043299686 0.052801543 0.090090850 0.034368460
          95           98           99          100          102          103          104          107          108          111          112
0.212864858 0.133213570 0.276548212 0.104165219 0.127493276 0.169750745 0.236640054 0.088143318 0.057797938 0.038502322 0.065653015
         113          115          116          117          120          121          124          125          126          128          129
0.026248148 0.010735549 0.058452731 0.095839709 0.003989437 0.017851751 0.032583372 0.034465726 0.014594986 0.038058341 0.052148515
         130          133          134          135          137          138          139          142          143            1            2
0.090904883 0.020712360 0.061427865 0.001765741 0.052507879 0.070997003 0.111778333 0.077014874 0.085312234 0.037301723 0.029064321
           5            6            9           10           14           18           19           22           23           27           31
0.011765934 0.071257172 0.044127796 0.021082946 0.069972196 0.085843356 0.040476847 0.010555595 0.044805599 0.089417128 0.039938674
          32           35           36           40           41           44           45           49           53           54           57
0.014701447 0.036066350 0.013413319 0.004446262 0.032323729 0.012758356 0.064175618 0.085806319 0.006645509 0.020608059 0.046897134
          58           61           62           66           70           71           74           75           79           80           83
0.016507005 0.050084860 0.040977641 0.040340381 0.077744988 0.052522802 0.007753231 0.022604185 0.084890221 0.023363846 0.032802062
          84           88           92           93           96           97          101          105          106          109          110
0.005032189 0.042246649 0.051975497 0.002578333 0.031419394 0.083223621 0.059572476 0.006292400 0.084307821 0.040510523 0.028282014
         114          118          119          122          123          127          131          132          136          140          141
0.158182953 0.057793148 0.011487864 0.026633858 0.068916845 0.111543439 0.043321238 0.018163387 0.054678430 0.030174803 0.041016444
>
> mean(abs(model2$residuals/input$Weekly_Sales)) #MAPE
[1] 0.06160946
> mean(abs(model3$residuals/input$Weekly_Sales))
[1] 0.06152315
> mean(abs(model1$residuals/input$Weekly_Sales))
[1] 0.06196679
> mean(abs(model$residuals/input$Weekly_Sales))
[1] 0.06196679
>
```