# Marketing Mix Mode

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Created/Updated by | Create/Update date | Reviewed by | Comments |
| 1 | Samtha Reddy | 11/26/2018 |  | Created for Indian tech |

# Files used:

## Excel files:

1. RawData\_CPSMediaBudget.csv
2. CPS\_InquiryData.csv
3. inquiry\_rawdf\_usstate.csv # us inquiries file, with fixed missing zip codes
4. inquiry\_rawdf\_usstate\_updated
5. inquiry\_rawdf\_usstate\_updated2
6. Inquirydf\_media\_dma1

## R code files:

1. Basic Stats Initial Analysis2
2. MM Model v4(DMA)

# w/o DMA considerations, not removing any international students

## Steps

1. After summing the media costs for each medium per week per year- June 2017-May 2018, June 2018-May 2019, we get below stats:

NA’s mean there was not cost of that medium for some weeks

> summary(MediaCostdf\_yrwk$DigitalCost)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

3960 12349 13261 13215 16576 16576 2

> summary(MediaCostdf\_yrwk$TVCost)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

4400 52800 71750 63903 75445 94188 35

> summary(MediaCostdf\_yrwk$RadioCost)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

10095 29658 45279 49184 68639 77002 35

> summary(MediaCostdf\_yrwk$BillboardCost)

Min. 1st Qu. Median Mean 3rd Qu. Max.

1094 1138 1368 1659 1423 10814

> summary(MediaCostdf\_yrwk$OpenHouseCost)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

2499 2499 2499 2499 2499 2499 62

> summary(MediaCostdf\_yrwk$OutdoorCost)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

1929 1929 1929 2067 2291 2412 50

> summary(MediaCostdf\_yrwk$TotalCost)

Min. 1st Qu. Median Mean 3rd Qu. Max.

5055 15323 19260 66234 124238 187941

Inquiry – from 2nd week 2016 till 26th week, 2018

> mean(Inquirydf\_yrwk\_all$Inquiries)

[1] 88.68462

> summary(Inquirydf\_yrwk\_all$Inquiries)

Min. 1st Qu. Median Mean 3rd Qu. Max.

14.00 66.25 84.00 88.68 106.75 327.00

But the final dataset which we try to analyze

#NOTE- we decide to DROP 25th row #OUTLIEAR

> summary(df\_yrwk\_all$Inquiries)

Min. 1st Qu. Median Mean 3rd Qu. Max.

14.00 70.50 86.50 92.56 117.00 188.00

> summary(df\_yrwk\_all$DigitalCost)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0 12853 13261 13072 16576 16576

> summary(df\_yrwk\_all$TVCost)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0 0 0 25815 52800 94188

> summary(df\_yrwk\_all$RadioCost)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0 0 0 19666 35746 76715

> summary(df\_yrwk\_all$BillboardCost)

Min. 1st Qu. Median Mean 3rd Qu. Max.

1094 1094 1368 1589 1805 5692

> summary(df\_yrwk\_all$OpenHouseCost)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0 0 0 0 0 0

> summary(df\_yrwk\_all$OutdoorCost)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00 0.00 0.00 38.59 0.00 1929.40

> summary(df\_yrwk\_all$TotalCost)

Min. 1st Qu. Median Mean 3rd Qu. Max.

5055 14533 18163 60181 103484 185292

1. Based on the final dataset we see **for the given timeframe** we are analyzing we can **drop Open House and Outdoor Cost.**

Analyzing **correlation** between Inquiry and each type of cost:

> cor(as.numeric(df\_yrwk\_all$Inquiries), df\_yrwk\_all$DigitalCost, method ="pearson")

[1] 0.2232691

> cor(as.numeric(df\_yrwk\_all$Inquiries), df\_yrwk\_all$DigitalCost, method ="pearson")

[1] 0.2232691

> cor(as.numeric(df\_yrwk\_all$Inquiries), df\_yrwk\_all$TVCost, method ="pearson")

[1] -0.07808704

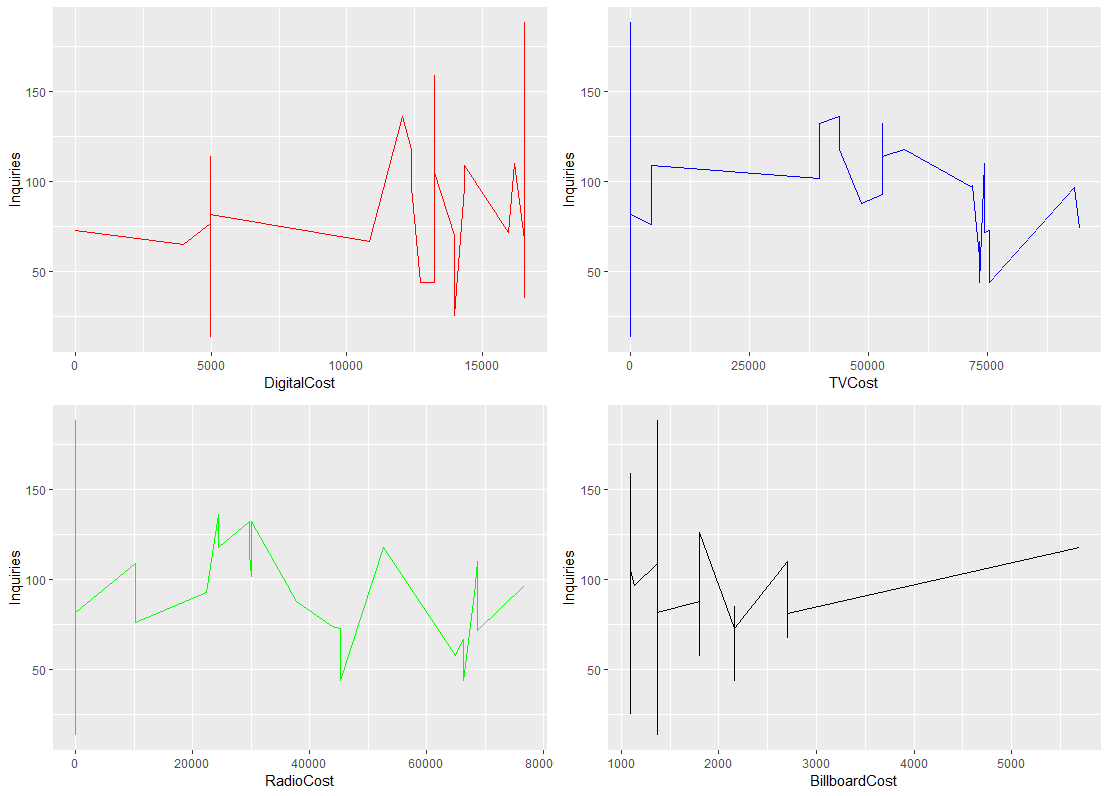
> cor(as.numeric(df\_yrwk\_all$Inquiries), df\_yrwk\_all$RadioCost, method ="pearson")

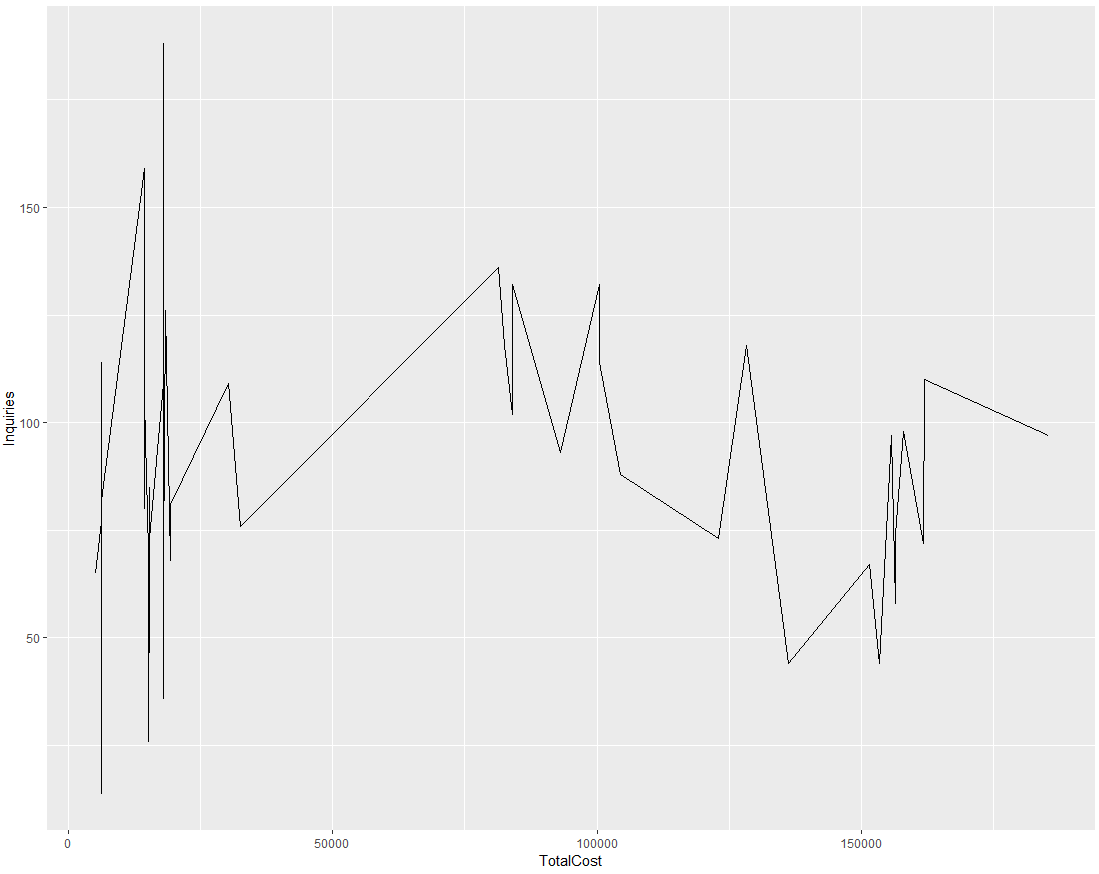
[1] -0.09031698

> cor(as.numeric(df\_yrwk\_all$Inquiries), df\_yrwk\_all$BillboardCost, method ="pearson")

[1] -0.06032666

# all methods-pearson, kendall, spearman don't show any strong relationship





1. Divide data into test(20%data, 8records) and train(80%, 42records)
2. Using nlsLM on train data and all the cost we get the rate
3. Using the rate we created adstock values of each cost
4. Then using lm method on train data we create the lm model
5. Recreate test data by calculating the adstock values of cost in test data
6. Run lm model created in step 4 on modified test data.
7. Calculate rmse value
8. Repeat the steps 1 to 7 a 100 times(cross validation), and take final model which has least rmse value

## R code

####################################################################################

#set path where all files are kept

setwd("C:/Users/Samtha Reddy/Dropbox (Personal)/Copy of Work/R Work/Marketing Mix Model")

#function to check if packages installed or not and if not install it.

#Then loads all the list of packages, use require to

#check if package installed then only load it or else give a warning .

check.packages <- function(pkg){

new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]

if (length(new.pkg))

install.packages(new.pkg, dependencies = TRUE)

sapply(pkg, require, character.only = TRUE)

}

#packages to be installed and loaded

packages<-c("sqldf", "gsubfn", "proto", "RSQLite", "dplyr", "caret","lattice","ggplot2","minpack.lm","gridExtra")

check.packages(packages)

#load budget file

media\_rawdf <- read.csv("RawData\_CPSMediaBudget.csv", header = TRUE) #2806 media records, 20 variables

#load inquiry file

inquiry\_rawdf <- read.csv("CPS\_InquiryData.csv",header= TRUE, na.strings = c("", "NA")) #11529 inquiries, 23 variables

# rename the colnames in dataframe with "." as SQL command do not work with "." if in col names

names(media\_rawdf)[names(media\_rawdf) == "Cost.Digital"] <- "Cost\_Digital"

names(media\_rawdf)[names(media\_rawdf) == "Cost.TV"] <- "Cost\_TV"

names(media\_rawdf)[names(media\_rawdf) == "Cost.Radio"] <- "Cost\_Radio"

names(media\_rawdf)[names(media\_rawdf) == "Cost.Billiboard"] <- "Cost\_Billboard"

names(media\_rawdf)[names(media\_rawdf) == "Cost.Open.House"] <- "Cost\_OpenHouse"

names(media\_rawdf)[names(media\_rawdf) == "Cost.outdoor"] <- "Cost\_Outdoor"

#replacing na with 0

#media\_rawdf$Cost\_Digital[is.na(media\_rawdf$Cost\_Digital)] <- 0

#media\_rawdf$Cost\_TV[is.na(media\_rawdf$Cost\_TV)] <- 0

#media\_rawdf$Cost\_Radio[is.na(media\_rawdf$Cost\_Radio)] <- 0

#media\_rawdf$Cost\_Billboard[is.na(media\_rawdf$Cost\_Billboard)] <- 0

#media\_rawdf$Cost\_OpenHouse[is.na(media\_rawdf$Cost\_OpenHouse)] <- 0

#media\_rawdf$Cost\_Outdoor[is.na(media\_rawdf$Cost\_Outdoor)] <- 0

# ------------------Without DMA considerations, not removing any international students ----------------------------------#

#grouping all the cost of all mediums by respective week for each year

MediaCostdf\_yrwk <- sqldf('select Year , WeekNum, sum(Cost) as TotalCost,

sum(Cost\_Digital) as DigitalCost,

sum(Cost\_TV) as TVCost,

sum(Cost\_Radio) as RadioCost,

sum(Cost\_Billboard) as BillboardCost,

sum(Cost\_OpenHouse) as OpenHouseCost,

sum(Cost\_Outdoor) as OutdoorCost

from media\_rawdf group by Year,WeekNum')

summary(MediaCostdf\_yrwk$DigitalCost)

summary(MediaCostdf\_yrwk$TVCost)

summary(MediaCostdf\_yrwk$RadioCost)

summary(MediaCostdf\_yrwk$BillboardCost)

summary(MediaCostdf\_yrwk$OpenHouseCost)

summary(MediaCostdf\_yrwk$OutdoorCost)

summary(MediaCostdf\_yrwk$TotalCost)

#fill all NA's to zero

MediaCostdf\_yrwk[is.na(MediaCostdf\_yrwk)] <- 0

#grouping all the inquiries of all mediums by respective week for each year

Inquirydf\_yrwk\_all<-sqldf('select Year, WeekNum,count(\*) as Inquiries from inquiry\_rawdf group by Year,WeekNum')

mean(Inquirydf\_yrwk\_all$Inquiries)

summary(Inquirydf\_yrwk\_all$Inquiries)

#combining inquiries and medium cost for the respective weeks in a single dataframe , in that process we loose some data from cost as inquiry only till june 2018 but budget is till sep 2018

df\_yrwk\_all <-sqldf("SELECT m.Year, m.WeekNum , m.TotalCost,m.DigitalCost, m.TVCost,m.RadioCost,m.BillboardCost,

m.OpenHouseCost, m.OutdoorCost, i.Inquiries

FROM Inquirydf\_yrwk\_all i JOIN MediaCostdf\_yrwk m

ON i.Year = m.Year

AND i.WeekNum = m.WeekNum") # 51 obs, 10 variables

summary(df\_yrwk\_all$Inquiries)

#NOTE- we decide to DROP 25th row #OUTLIEAR

df\_yrwk\_all <- df\_yrwk\_all[-1,]

summary(df\_yrwk\_all$DigitalCost)

summary(df\_yrwk\_all$TVCost)

summary(df\_yrwk\_all$RadioCost)

summary(df\_yrwk\_all$BillboardCost)

summary(df\_yrwk\_all$OpenHouseCost)

summary(df\_yrwk\_all$OutdoorCost)

summary(df\_yrwk\_all$TotalCost)

df\_yrwk\_all[is.na(df\_yrwk\_all)] <- 0

#----------general visualizations to understand relations ---

cor(as.numeric(df\_yrwk\_all$Inquiries), df\_yrwk\_all$DigitalCost, method ="pearson")

cor(as.numeric(df\_yrwk\_all$Inquiries), df\_yrwk\_all$TVCost, method ="pearson")

cor(as.numeric(df\_yrwk\_all$Inquiries), df\_yrwk\_all$RadioCost, method ="pearson")

cor(as.numeric(df\_yrwk\_all$Inquiries), df\_yrwk\_all$BillboardCost, method ="pearson")

# all methods - pearson, kendall, spearman - don't show any strong relationship

#df\_yrwk\_Digital <-sqldf("SELECT m.Year, m.WeekNum , m.TotalCost,m.DigitalCost, i.Inquiries FROM Inquirydf\_yrwk\_all i JOIN MediaCostdf\_yrwk m ON i.Year = m.Year AND i.WeekNum = m.WeekNum AND m.DigitalCost <> 0 ") # 49 obs, 10 variables

#df\_yrwk\_TV <-sqldf("SELECT m.Year, m.WeekNum , m.TotalCost,m.TVCost, i.Inquiries FROM Inquirydf\_yrwk\_all i JOIN MediaCostdf\_yrwk m ON i.Year = m.Year AND i.WeekNum = m.WeekNum AND m.TVCost <> 0 ") # 22 obs, 5 variables

#df\_yrwk\_Radio <-sqldf("SELECT m.Year, m.WeekNum , m.TotalCost,m.RadioCost, i.Inquiries FROM Inquirydf\_yrwk\_all i JOIN MediaCostdf\_yrwk m ON i.Year = m.Year AND i.WeekNum = m.WeekNum AND m.RadioCost <> 0 ") # 22 obs, 5 variables

#df\_yrwk\_Billboard <-sqldf("SELECT m.Year, m.WeekNum , m.TotalCost,m.BillboardCost, i.Inquiries FROM Inquirydf\_yrwk\_all i JOIN MediaCostdf\_yrwk m ON i.Year = m.Year AND i.WeekNum = m.WeekNum AND m.BillboardCost <> 0 ")

P1 <- ggplot(df\_yrwk\_all, aes(x = DigitalCost, y = Inquiries)) + geom\_line(color="red")

P2 <- ggplot(df\_yrwk\_all, aes(x = TVCost, y = Inquiries)) + geom\_line(color="blue")

P3 <- ggplot(df\_yrwk\_all, aes(x = RadioCost, y = Inquiries)) + geom\_line(color="green")

P4 <- ggplot(df\_yrwk\_all, aes(x = BillboardCost, y = Inquiries)) + geom\_line(color="black")

grid.arrange(P1,P2,P3,P4,ncol = 2,nrow=2)

ggplot(df\_yrwk\_all, aes(x = TotalCost, y = Inquiries)) + geom\_line(color="black")

# --------------calculating adstock rate ----------------------------------#

adstock<-function(x,rater=0){

adstock\_val<-as.numeric(stats::filter(x=x,filter=rater,method="recursive"))

#print(adstock\_val)

return(adstock\_val)

}

#----------------running the model----------------------------------------------#

score\_v6 = list()

scoretrain\_v6 = list()

number\_v6 <- 0

bestscore\_v6 <-100

bestrate\_v6 <- 0

for( i in 1:100)

{

#create indexes for train data

#create train data

train.index\_v6 <- createDataPartition(df\_yrwk\_all$Inquiries, p = 0.8, list = FALSE)

train.data\_v6 <- df\_yrwk\_all[train.index\_v6, ]

#create test data

test.data\_v6<- df\_yrwk\_all[-train.index\_v6, ]

#build model

model\_v6\_nls <- nlsLM(data= train.data\_v6, Inquiries~ b0 + b1 \* adstock(DigitalCost,rater) +

b2 \* adstock(TVCost, rater) +

b3 \* adstock(RadioCost, rater) +

b4 \* adstock(BillboardCost, rater),

algorithm = "LM",

start = c(b0=1, b1= 1, b2= 1, b3=1, b4=1, rater=0),

lower = c(b0=-Inf, b1=-Inf, b2=-Inf,b3=-Inf, b4=-Inf,rater=0),

upper = c(b0= Inf, b1= Inf, b2= Inf, b3=Inf, b4=Inf, rater=1))

newrate <- summary(model\_v6\_nls)$coefficients[6,1]

#summary(model\_v6\_nls)

train.data\_v6$DigitalCostAd <- adstock(train.data\_v6$DigitalCost,newrate)

train.data\_v6$TVCostAd <-adstock(train.data\_v6$TVCost,newrate)

train.data\_v6$RadioCostAd <- adstock(train.data\_v6$RadioCost, newrate)

train.data\_v6$BillboardCostAd <- adstock(train.data\_v6$BillboardCost, newrate)

train.data\_v6[is.na(train.data\_v6)] <- 0

modFit\_v6\_lm<-lm(Inquiries~DigitalCostAd +

TVCostAd +

RadioCostAd +

BillboardCostAd ,

data =train.data\_v6)

#predict(modFit\_v5\_lm, new\_data= train.data\_v5, interval = c("confidence"), level = .95)

#print(summary(model\_v5))

#e build the test data

test.data\_v6$DigitalCostAd <- adstock(test.data\_v6$DigitalCost,newrate)

test.data\_v6$TVCostAd <- adstock(test.data\_v6$TVCost,newrate)

test.data\_v6$RadioCostAd <- adstock(test.data\_v6$RadioCost,newrate)

test.data\_v6$BillboardCostAd <- adstock(test.data\_v6$BillboardCost,newrate)

test.data\_v6[is.na(test.data\_v6)] <- 0

# Make predictions

#summary(modFitv4\_DMA.08)

predictionsv6 <- modFit\_v6\_lm %>% predict(test.data\_v6)

# Model performance

# (a) Prediction error, RMSE

score\_v6[i]=RMSE(predictionsv6, test.data\_v6$Inquiries)

#predictionsv5\_train <- model %>% predict(train.data\_v4\_DMA)

#scoretrain[i]<- RMSE(predictionsv4\_DMA.10\_train , train.data\_v4\_DMA$Inquiries)

if(as.numeric(score\_v6[i]) < as.numeric(bestscore\_v6))

{

bestscore\_v6 = score\_v6[i]

bestmodel <- modFit\_v6\_lm

number\_v6 <- i

bestrate\_v6<- newrate

predictionsv6\_train <- modFit\_v6\_lm %>% predict(train.data\_v6)

scoretrain\_v6[i] <- RMSE(predictionsv6\_train , train.data\_v6$Inquiries)

maev6\_test<- abs(mean((predictionsv6-test.data\_v6$Inquiries))) #Mean absolute error

}

rm(train.data\_v6)

rm(test.data\_v6)

}

print(score\_v6[number\_v6])

print(scoretrain\_v6[number\_v6])

print(number\_v6)

print(bestscore\_v6)

print(bestrate\_v6)

summary(bestmodel)

#final model

#based on best model and bestrate we got we see only digital cost is the best related to inquiries and recreate the model with Adstock digital cost

df\_yrwk\_all\_new <- df\_yrwk\_all

df\_yrwk\_all\_new$DigitalCostAd <- adstock(df\_yrwk\_all\_new$DigitalCost,bestrate\_v6)

df\_yrwk\_all\_new$TVCostAd <-adstock(df\_yrwk\_all\_new$TVCost,bestrate\_v6)

df\_yrwk\_all\_new$RadioCostAd <- adstock(df\_yrwk\_all\_new$RadioCost, bestrate\_v6)

df\_yrwk\_all\_new$BillboardCostAd <- adstock(df\_yrwk\_all\_new$BillboardCost, bestrate\_v6)

df\_yrwk\_all\_new[is.na(df\_yrwk\_all\_new)] <- 0

predictionsv6\_full <- predict(bestmodel,df\_yrwk\_all\_new)

RMSE(predictionsv6\_full , df\_yrwk\_all\_new$Inquiries)

mae\_v6\_full <- abs(mean(predictionsv6\_full - df\_yrwk\_all\_new$Inquiries))

# using lm method adstock on digital only

model\_final\_v6.1<- lm(Inquiries~DigitalCostAd, data=df\_yrwk\_all\_new)

summary(model\_final\_v6.1)

confint(model\_final\_v6.1)

predictions\_full\_v6.1 <-predict(model\_final\_v6.1, df\_yrwk\_all\_new )

RMSE(predictions\_full\_v6.1 , df\_yrwk\_all\_new$Inquiries)

mea\_full\_v6.1 <- abs(mean(predictions\_full\_v6.1 - df\_yrwk\_all\_new$Inquiries))

##based on best model and bestrate we got we drop radio cost, and drop billboardCost as it is not signifcant as high p value and recreate the model with Adstock digital cost

# using lm method adstock on digital, TV only

model\_final\_v6.2<- lm(Inquiries~DigitalCostAd + TVCostAd , data=df\_yrwk\_all\_new)

summary(model\_final\_v6.2)

confint(model\_final\_v6.2)

predictions\_full\_v6.2 <-predict(model\_final\_v6.2, df\_yrwk\_all\_new )

RMSE(predictions\_full\_v6.2 , df\_yrwk\_all\_new$Inquiries)

mea\_full\_v6.2 <- abs(mean(predictions\_full\_v6.2 - df\_yrwk\_all\_new$Inquiries))

#========================================

par(mfrow = c(2, 2))

plot\_bestmodel <- plot(bestmodel)

plot\_model\_final1 <- plot(model\_final1)

plot\_model\_final2 <- plot(model\_final2)

grid.arrange(P4,P5,P6,P7,ncol = 2,nrow=2)

#varaiance data # The variance is a numerical measure of how the data values is dispersed around the mean.

sd(df\_yrwk\_all$DigitalCost)

sd(df\_yrwk\_all\_new$DigitalCostAd)

sd(df\_yrwk\_all$TVCost)

sd(df\_yrwk\_all\_new$TVCostAd)

sd(df\_yrwk\_all$BillboardCost)

sd(df\_yrwk\_all\_new$DigitalCostAd)

sd(df\_yrwk\_all$Inquiries)

sd(predictionsv5\_full1)

sd(predictionsv5\_full2)

##################using nls without adstock############################

model\_v7\_nls <- nlsLM(data= df\_yrwk\_all, Inquiries~ b0 + b1 \* DigitalCost +

b2 \* TVCost +

b3 \* RadioCost +

b4 \* BillboardCost,

algorithm = "LM",

start = c(b0=1, b1= 1, b2= 1, b3=1, b4=1),

lower = c(b0=-Inf, b1=-Inf, b2=-Inf,b3=-Inf, b4=-Inf),

upper = c(b0= Inf, b1= Inf, b2= Inf, b3=Inf, b4=Inf))

summary(model\_v7\_nls)

predictionsv7\_full <- predict(model\_v7\_nls,df\_yrwk\_all)

RMSE(predictionsv7\_full , df\_yrwk\_all$Inquiries)

mae\_v7\_full <- abs(mean(predictionsv7\_full - df\_yrwk\_all$Inquiries))

model\_v8\_nls <- nlsLM(data= df\_yrwk\_all, Inquiries~ b0 + b1 \* DigitalCost ,

algorithm = "LM",

start = c(b0=1, b1= 1 ),

lower = c(b0=-Inf, b1=-Inf),

upper = c(b0= Inf, b1= Inf))

summary(model\_v8\_nls)

predictionsv8\_full <- predict(model\_v8\_nls,df\_yrwk\_all)

RMSE(predictionsv8\_full , df\_yrwk\_all$Inquiries)

mae\_v8\_full <- abs(mean(predictionsv8\_full - df\_yrwk\_all$Inquiries))

model\_v9\_lm <- lm(data= df\_yrwk\_all, Inquiries~ DigitalCost + TVCost + RadioCost + BillboardCost)

summary(model\_v9\_lm)

predictionsv9\_full <- predict(model\_v9\_lm,df\_yrwk\_all)

RMSE(predictionsv9\_full , df\_yrwk\_all$Inquiries)

mae\_v9\_full <- abs(mean(predictionsv9\_full - df\_yrwk\_all$Inquiries))

model\_v10\_lm <- lm(data= df\_yrwk\_all, Inquiries~ DigitalCost )

summary(model\_v10\_lm)

predictionsv10\_full <- predict(model\_v10\_lm,df\_yrwk\_all)

RMSE(predictionsv10\_full , df\_yrwk\_all$Inquiries)

mae\_v10\_full <- abs(mean(predictionsv10\_full - df\_yrwk\_all$Inquiries))

#####################################################################################

## R results

>   
> print(score\_v6[number\_v6])

[[1]]

[1] 20.80795

> print(scoretrain\_v6[number\_v6])

[[1]]

[1] 31.23976

> print(number\_v6)

[1] 1

> print(bestscore\_v6)

[[1]]

[1] 20.80795

> print(bestrate\_v6)

[1] 0.7322605

>

> summary(bestmodel)

Call:

lm(formula = Inquiries ~ DigitalCostAd + TVCostAd + RadioCostAd +

BillboardCostAd, data = train.data\_v6)

Residuals:

Min 1Q Median 3Q Max

-55.291 -18.257 -3.756 21.734 79.701

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 13.1028891 29.4457720 0.445 0.658924

DigitalCostAd 0.0018262 0.0004786 3.816 0.000499 \*\*\*

TVCostAd 0.0009032 0.0005934 1.522 0.136446

RadioCostAd -0.0010429 0.0007020 -1.486 0.145870

BillboardCostAd -0.0023630 0.0045444 -0.520 0.606166

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 33.28 on 37 degrees of freedom

Multiple R-squared: 0.2903, Adjusted R-squared: 0.2135

F-statistic: 3.783 on 4 and 37 DF, p-value: 0.01121

> df\_yrwk\_all\_new <- df\_yrwk\_all

> df\_yrwk\_all\_new$DigitalCostAd <- adstock(df\_yrwk\_all\_new$DigitalCost,bestrate\_v6)

> df\_yrwk\_all\_new$TVCostAd <-adstock(df\_yrwk\_all\_new$TVCost,bestrate\_v6)

> df\_yrwk\_all\_new$RadioCostAd <- adstock(df\_yrwk\_all\_new$RadioCost, bestrate\_v6)

> df\_yrwk\_all\_new$BillboardCostAd <- adstock(df\_yrwk\_all\_new$BillboardCost, bestrate\_v6)

> df\_yrwk\_all\_new[is.na(df\_yrwk\_all\_new)] <- 0

>

> predictionsv6\_full <- predict(bestmodel,df\_yrwk\_all\_new)

> RMSE(predictionsv6\_full , df\_yrwk\_all\_new$Inquiries)

[1] 30.16815

> mae\_v6\_full <- abs(mean(predictionsv6\_full - df\_yrwk\_all\_new$Inquiries))

> mae\_v6\_full

[1] 3.110652

> # using lm method adstock used in model6 on digital only

>

> model\_final\_v6.1<- lm(Inquiries~DigitalCostAd, data=df\_yrwk\_all\_new)

> summary(model\_final\_v6.1)

Call:

lm(formula = Inquiries ~ DigitalCostAd, data = df\_yrwk\_all\_new)

Residuals:

Min 1Q Median 3Q Max

-60.961 -19.748 0.644 18.193 79.487

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.961e+01 1.661e+01 1.783 0.08097 .

DigitalCostAd 1.348e-03 3.429e-04 3.932 0.00027 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 31.2 on 48 degrees of freedom

Multiple R-squared: 0.2436, Adjusted R-squared: 0.2279

F-statistic: 15.46 on 1 and 48 DF, p-value: 0.0002703

> confint(model\_final\_v6.1)

2.5 % 97.5 %

(Intercept) -3.7864597756 62.999763737

DigitalCostAd 0.0006587684 0.002037651

> predictions\_full\_v6.1 <-predict(model\_final\_v6.1, df\_yrwk\_all\_new )

>

> RMSE(predictions\_full\_v6.1 , df\_yrwk\_all\_new$Inquiries)

[1] 30.57325

> mea\_full\_v6.1 <- abs(mean(predictions\_full\_v6.1 - df\_yrwk\_all\_new$Inquiries))

> mea\_full\_v6.1

[1] 9.522088e-15

> model\_final\_v6.2<- lm(Inquiries~DigitalCostAd + TVCostAd , data=df\_yrwk\_all\_new)

> summary(model\_final\_v6.2)

Call:

lm(formula = Inquiries ~ DigitalCostAd + TVCostAd, data = df\_yrwk\_all\_new)

Residuals:

Min 1Q Median 3Q Max

-61.387 -19.922 1.364 17.806 78.786

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.841e+01 1.734e+01 1.639 0.107936

DigitalCostAd 1.317e-03 3.650e-04 3.608 0.000746 \*\*\*

TVCostAd 3.002e-05 1.108e-04 0.271 0.787558

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 31.51 on 47 degrees of freedom

Multiple R-squared: 0.2448, Adjusted R-squared: 0.2127

F-statistic: 7.617 on 2 and 47 DF, p-value: 0.001363

> confint(model\_final\_v6.2)

2.5 % 97.5 %

(Intercept) -6.4664911853 63.294694452

DigitalCostAd 0.0005825743 0.002051223

TVCostAd -0.0001928173 0.000252860

> predictions\_full\_v6.2 <-predict(model\_final\_v6.2, df\_yrwk\_all\_new )

>

> RMSE(predictions\_full\_v6.2 , df\_yrwk\_all\_new$Inquiries)

[1] 30.54938

> mea\_full\_v6.2 <- abs(mean(predictions\_full\_v6.2 - df\_yrwk\_all\_new$Inquiries))

> mea\_full\_v6.2

[1] 1.179447e-14

########### with nls , without adstock, all#####################

> model\_v7\_nls <- nlsLM(data= df\_yrwk\_all, Inquiries~ b0 + b1 \* DigitalCost +

+ b2 \* TVCost +

+ b3 \* RadioCost +

+ b4 \* BillboardCost,

+ algorithm = "LM",

+ start = c(b0=1, b1= 1, b2= 1, b3=1, b4=1),

+ lower = c(b0=-Inf, b1=-Inf, b2=-Inf,b3=-Inf, b4=-Inf),

+ upper = c(b0= Inf, b1= Inf, b2= Inf, b3=Inf, b4=Inf))

> summary(model\_v7\_nls)

Formula: Inquiries ~ b0 + b1 \* DigitalCost + b2 \* TVCost + b3 \* RadioCost +

b4 \* BillboardCost

Parameters:

Estimate Std. Error t value Pr(>|t|)

b0 72.7356489 19.0031426 3.828 0.000397 \*\*\*

b1 0.0020105 0.0012041 1.670 0.101921

b2 0.0001086 0.0005235 0.207 0.836606

b3 -0.0002791 0.0006673 -0.418 0.677746

b4 -0.0023732 0.0069517 -0.341 0.734409

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 35.77 on 45 degrees of freedom

Number of iterations to convergence: 2

Achieved convergence tolerance: 1.49e-08

> predictionsv7\_full <- predict(model\_v7\_nls,df\_yrwk\_all)

> RMSE(predictionsv7\_full , df\_yrwk\_all$Inquiries)

[1] 33.93652

> mae\_v7\_full <- abs(mean(predictionsv7\_full - df\_yrwk\_all$Inquiries))

> mae\_v7\_full

[1] 3.407601e-08

########### with nls , without adstock, only digital, ##################

> model\_v8\_nls <- nlsLM(data= df\_yrwk\_all, Inquiries~ b0 + b1 \* DigitalCost ,

+

+ algorithm = "LM",

+ start = c(b0=1, b1= 1 ),

+ lower = c(b0=-Inf, b1=-Inf),

+ upper = c(b0= Inf, b1= Inf))

> summary(model\_v8\_nls)

Formula: Inquiries ~ b0 + b1 \* DigitalCost

Parameters:

Estimate Std. Error t value Pr(>|t|)

b0 68.394973 16.010764 4.272 9.12e-05 \*\*\*

b1 0.001849 0.001165 1.587 0.119

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 34.97 on 48 degrees of freedom

Number of iterations to convergence: 2

Achieved convergence tolerance: 1.49e-08

> predictionsv8\_full <- predict(model\_v8\_nls,df\_yrwk\_all)

> RMSE(predictionsv8\_full , df\_yrwk\_all$Inquiries)

[1] 34.26608

> mae\_v8\_full <- abs(mean(predictionsv8\_full - df\_yrwk\_all$Inquiries))

> mae\_v8\_full

[1] 2.951084e-08

> predictionsv8\_full <- predict(model\_v8\_nls,df\_yrwk\_all)

> RMSE(predictionsv8\_full , df\_yrwk\_all$Inquiries)

[1] 34.26608

> mae\_v8\_full <- abs(mean(predictionsv8\_full - df\_yrwk\_all$Inquiries))

> mae\_v8\_full

[1] 2.951084e-08

|  |
| --- |
| #########simple lm method####################################################  model\_v9\_lm <- lm(data= df\_yrwk\_all, Inquiries~ DigitalCost + TVCost + RadioCost + BillboardCost)  >  >  > summary(model\_v9\_lm)  Call:  lm(formula = Inquiries ~ DigitalCost + TVCost + RadioCost + BillboardCost,  data = df\_yrwk\_all)  Residuals:  Min 1Q Median 3Q Max  -72.231 -23.649 3.557 22.960 85.184  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 72.7356498 19.0031446 3.828 0.000397 \*\*\*  DigitalCost 0.0020105 0.0012041 1.670 0.101921  TVCost 0.0001086 0.0005235 0.207 0.836606  RadioCost -0.0002791 0.0006673 -0.418 0.677746  BillboardCost -0.0023732 0.0069517 -0.341 0.734409  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 35.77 on 45 degrees of freedom  Multiple R-squared: 0.06804, Adjusted R-squared: -0.0148  F-statistic: 0.8213 on 4 and 45 DF, p-value: 0.5185  >  > predictionsv9\_full <- predict(model\_v9\_lm,df\_yrwk\_all)  > RMSE(predictionsv9\_full , df\_yrwk\_all$Inquiries)  [1] 33.93652  > mae\_v9\_full <- abs(mean(predictionsv9\_full - df\_yrwk\_all$Inquiries))  > mae\_v9\_full  [1] 3.083865e-14  > model\_v10\_lm <- lm(data= df\_yrwk\_all, Inquiries~ DigitalCost )  >  >  > summary(model\_v10\_lm)  Call:  lm(formula = Inquiries ~ DigitalCost, data = df\_yrwk\_all)  Residuals:  Min 1Q Median 3Q Max  -68.226 -23.929 0.774 23.559 88.962  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 68.394973 16.010764 4.272 9.12e-05 \*\*\*  DigitalCost 0.001849 0.001165 1.587 0.119  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 34.97 on 48 degrees of freedom  Multiple R-squared: 0.04985, Adjusted R-squared: 0.03005  F-statistic: 2.518 on 1 and 48 DF, p-value: 0.1191  >  > predictionsv10\_full <- predict(model\_v10\_lm,df\_yrwk\_all)  > RMSE(predictionsv10\_full , df\_yrwk\_all$Inquiries)  [1] 34.26608  > mae\_v10\_full <- abs(mean(predictionsv10\_full - df\_yrwk\_all$Inquiries))  > mae\_v10\_full  [1] 2.216701e-14 |
|  |

**Key points to note:**

There is less variability in the predicted values than the actual values.

> sd(df\_yrwk\_all$DigitalCost)

[1] 4288.842

> sd(df\_yrwk\_all\_new$DigitalCostAd)

[1] 14376.17

> sd(df\_yrwk\_all$TVCost)

[1] 33242.29

> sd(df\_yrwk\_all\_new$TVCostAd)

[1] 43679.36

> sd(df\_yrwk\_all$Inquiries)

[1] 35.51037

> sd(predictionsv5\_full2)

[1] 21.86384

> sd(predictionsv5\_full1)

[1] 20.44221

##############################################################################

# Removing international inquiries & mapping inquiry to DMA.

Since mapping to DMA, no of inquiries coming for that week we get 9 possible values so we increase our number points and possibility to get better fit model

## Steps

|  |  |  |  |
| --- | --- | --- | --- |
|  | No of Records | no of variables | Timeline |
| inquiry\_rawdf | 11529 | 23 | 2nd week 2016 - 26th week 2018 |
| media\_rawdf | 2806 | 20 | 25th week 2017 -39th week 2018 |

1. 3935 international records dropped from 11529 inquiry records.
2. fix all missing zipcodes in inquiry file and reload it back. Remove the one "Canada" inquiry as well. Now we have #7593 inquiry records left.
3. We then map DMA to the zip code of each inquiry records. (We use publicly available DMA – zipcode. We find 60 inquires not getting mapped to any DMA. We download the file again fix those inquiries zipcode(The issue zipcode entries were incorrect we had search those 60 inquires in

google to get exact zip code). So, the one which could not map to any correct US state zipcode we dropped so 7592 entries left.

1. After re mapping the DMA to zip code of the inquiry we still had 5 records that could not be mapped to any DMA.
2. We create a DMA – DMA City related to Indiana tech table:

dma\_formedia<-data.frame("DMA\_City" = c('Louisville','Cincinnati','Evansville','Fort Wayne','Indianapolis','Lafayette','Munster','South Bend','Chicago','NW IN','Detroit'), "DMA\_CODE" = c('529','515','649','509','527','582','602','588','602','602','505'))

1. Mapped DMA location media file to the respective DMA code – though 11 cities but only 9 unique DMA code
2. Now mapping inquiries (7592 records) to above key 9 DMA locations, 508 inquiries could not be mapped to key 9 DMAs.
3. Downloaded and manually changed to map 9 key DMA locations where MEDIA\_DMA\_CODE = NA, after removing 2016 data + 2017 and its all weeknum less than = 25, so left with 3062 records.

We manually updated the DMA for them by considering nearest media DMA locations like:

WV, VA- Cincinnati

For TN - Louisville

MO- Evansville

IA- Chicago

1. Remapped the inquiries dataset from step 8 to key 9 DMA but still 82 records could not be mapped as those came from some non-neighboring cities of INDIANA state. So final inquiry dataset left (3062- 82 = 2980)
2. grouping all the inquiries of all mediums by respective week for each year, and each DMA – 397 observations[# check :sum(df\_yrwk\_dma$Inquiries) # has to match count of Inquirydf\_media\_dma\_updated3 i.e 2980]
3. grouping all the media cost by respective week for each year, and each DMA – 471 observations
4. Grouping the media and inquiry file from time frame 26th week 2017 till 26th week 2018, our final dataset is 397 observations.

## R Code

#set path where all files are kept

setwd("C:/Users/Samtha Reddy/Dropbox (Personal)/Copy of Work/R Work/Marketing Mix Model")

#load budget file

media\_rawdf <- read.csv("RawData\_CPSMediaBudget.csv", header = TRUE) #2806 media records

#load inquiry file

inquiry\_rawdf <- read.csv("CPS\_InquiryData.csv",header= TRUE, na.strings = c("", "NA")) #11529 inquiries

dmazip\_df <- read.csv("DMA-ZIP.csv", header= TRUE)

#rename the colnames in new df to match the train.data dataframe else predict() will not work

names(dmazip\_df)[names(dmazip\_df) == "DMA.CODE"] <- "DMA\_CODE"

names(dmazip\_df)[names(dmazip\_df) == "DMA.NAME"] <- "DMA\_NAME"

#function to check if packages installed or not and if not install it.

#Then loads all the list of packages, use require to

#check if package installed then only load it or else give a warning .

check.packages <- function(pkg){

new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]

if (length(new.pkg))

install.packages(new.pkg, dependencies = TRUE)

sapply(pkg, require, character.only = TRUE)

}

#packages to be installed and loaded

packages<-c("sqldf", "gsubfn", "proto", "RSQLite", "dplyr", "caret","lattice","ggplot2","minpack.lm","gridExtra")

check.packages(packages)

# rename the colnames in dataframe with "." as SQL command do not work with "." if in col names

names(media\_rawdf)[names(media\_rawdf) == "Cost.Digital"] <- "Cost\_Digital"

names(media\_rawdf)[names(media\_rawdf) == "Cost.TV"] <- "Cost\_TV"

names(media\_rawdf)[names(media\_rawdf) == "Cost.Radio"] <- "Cost\_Radio"

names(media\_rawdf)[names(media\_rawdf) == "Cost.Billiboard"] <- "Cost\_Billboard"

names(media\_rawdf)[names(media\_rawdf) == "Cost.Open.House"] <- "Cost\_OpenHouse"

names(media\_rawdf)[names(media\_rawdf) == "Cost.outdoor"] <- "Cost\_Outdoor"

# removing international students by removing records where State column = ""

inquiry\_rawdf\_nonstate\_index <- which(is.na(inquiry\_rawdf$state))

length(inquiry\_rawdf\_nonstate\_index) #3935 international records

inquiry\_rawdf\_usstate <- inquiry\_rawdf[-inquiry\_rawdf\_nonstate\_index, ] #7594 us state inquiries

#remove spaces in location column in media file

media\_rawdf$Location <- trimws(media\_rawdf$Location)

#chnaging name of some locations in media file

media\_rawdf$Location[media\_rawdf$Location=="Fort Wayne/Huntington/ Kendallville"] <-"Fort Wayne"

media\_rawdf$Location[media\_rawdf$Location=="Cincy/Northern KY"] <-"Cincinnati"

unique(media\_rawdf$Location)# 11 key media locations

#------------------------------------------------------------------------------------

# using subset function - trying to find DMA code for all key 13 locations

#newdata <- unqiue(subset(dmazip\_df, DMA\_NAME in ("LOUISVILLE","CINCINNATI"), select=c(DMA\_NAME, DMA\_CODE)))

#unique(dmazip\_df$DMA\_CODE[dmazip\_df$DMA\_NAME=="LOUISVILLE"])

#unique(dmazip\_df$DMA\_CODE[dmazip\_df$DMA\_NAME=="CINCINNATI"])

#unique(dmazip\_df$DMA\_CODE[dmazip\_df$DMA\_NAME=="EVANSVILLE"])

#unique(dmazip\_df$DMA\_CODE[dmazip\_df$DMA\_NAME=="FORT WAYNE"])

#unique(dmazip\_df$DMA\_CODE[dmazip\_df$DMA\_NAME=="INDIANAPOLIS"])

#unique(dmazip\_df$DMA\_CODE[dmazip\_df$DMA\_NAME=="KOKOMO"])

#unique(dmazip\_df$DMA\_CODE[dmazip\_df$DMA\_NAME=="NW IN"])

#----------------------------------------------------------------------------

#fixing inquiries which have no zipcode

inquiry\_rawdf\_usstate$city <- (trimws(inquiry\_rawdf\_usstate$city))

inquiry\_rawdf\_usstate$state <- (trimws(inquiry\_rawdf\_usstate$state))

#download us inquiries file once and fix it and reload hence to be done only once

#write.csv(inquiry\_rawdf\_usstate,"inquiry\_rawdf\_usstate1.csv")

# fix all missing zipcodes and reload it back , make sure to remove first extra column from csv before loading it.remove the one "Canada" inquiry

inquiry\_rawdf\_usstate\_updated<- read.csv("inquiry\_rawdf\_usstate.csv",header= TRUE, na.strings = c("", "NA")) #7593 records

#check

which(is.na(inquiry\_rawdf\_usstate\_updated$zip))# this should return zero

#DMA Mapping

#create unquie dma-zip dataframe for loolup purposes

lookup\_DMA <-unique(dmazip\_df)

#adding DMA to inquiry based on DMA-zipcode file

inquiry\_rawdf\_usstate\_updated <- sqldf("select i.\*,l.DMA\_CODE from inquiry\_rawdf\_usstate\_updated i left join lookup\_DMA l ON i.zip = l.ZIPCODE ")

str(inquiry\_rawdf\_usstate\_updated )

missing\_dma<-subset(inquiry\_rawdf\_usstate\_updated, is.na(DMA\_CODE),select=c(city, state, zip)) # inquiries dont have dma code mapped out, it seems issue with their zipcode- so download fix their zipcodes

#do this only once

#write.csv(inquiry\_rawdf\_usstate\_updated, "inquiry\_rawdf\_usstate\_updated1.csv")

inquiry\_rawdf\_usstate\_updated2<- read.csv("inquiry\_rawdf\_usstate\_updated.csv",header= TRUE, na.strings = c("", "NA"))

str(inquiry\_rawdf\_usstate\_updated2)

#str(inquiry\_rawdf\_usstate\_updated2)

trimws(inquiry\_rawdf\_usstate\_updated2$zip)

inquiry\_rawdf\_usstate\_updated2<-inquiry\_rawdf\_usstate\_updated2[,1:24]# drop DMA\_Code column #7592 records

#Do re adding DMA to inquiry based on DMA-zipcode file after fixing zipcodes

inquiry\_rawdf\_usstate\_updated2 <- sqldf("select i.\*,l.DMA\_CODE from inquiry\_rawdf\_usstate\_updated2 i left join lookup\_DMA l ON i.zip = l.ZIPCODE ")#7592 records

missing\_dma2<-subset(inquiry\_rawdf\_usstate\_updated2, is.na(DMA\_CODE),select=c(city, state, zip, DMA\_CODE)) # still 5 records no DMA mapped

#newdata <- unqiue(subset(dmazip\_df, DMA\_NAME in ("LOUISVILLE","CINCINNATI"), select=c(DMA\_NAME, DMA\_CODE)))

# using subset function - trying to find DMA code for all key 13 locations

#newdata <- unqiue(subset(dmazip\_df, DMA\_NAME in ("LOUISVILLE","CINCINNATI"), select=c(DMA\_NAME, DMA\_CODE)))

dma\_formedia <- data.frame("DMA\_City" = c('Louisville','Cincinnati','Evansville','Fort Wayne','Indianapolis','Lafayette','Munster','South Bend', 'Chicago','NW IN','Detroit'), "DMA\_CODE" = c('529','515','649','509','527','582','602','588', '602','602','505'))

media\_rawdf<- sqldf("select i.\*,l.DMA\_Code from media\_rawdf i left join dma\_formedia l ON i.location = l.DMA\_City ")

#media\_rawdf<-media\_rawdf[,1:20]

which(is.na(media\_rawdf$DMA\_CODE))# should return zero

#write.csv(inquiry\_rawdf\_usstate\_updated2,"inquiry\_rawdf\_usstate\_updated2.csv")

# now map inquiries to above key 9 DMA locations

dma\_formedia\_lookup <- as.data.frame(unique(dma\_formedia$DMA\_CODE))

names(dma\_formedia\_lookup)[names(dma\_formedia\_lookup) == "unique(dma\_formedia$DMA\_CODE)"] <- "MEDIA\_DMA\_CODE"

Inquirydf\_media\_dma<-sqldf("SELECT i.\*, m.MEDIA\_DMA\_CODE

FROM inquiry\_rawdf\_usstate\_updated2 i left JOIN dma\_formedia\_lookup m

ON i.DMA\_CODE = m.MEDIA\_DMA\_CODE") #HAS TO BE 7592 records ONLY

length(which(is.na(Inquirydf\_media\_dma$MEDIA\_DMA\_CODE)))# 508 inquiries could not be mapped to key 9 DMAs

#-- downloaded this file and manually changed to map 12 key DMA locations where MEDIA\_DMA\_CODE = NA, and removed 2016 data + 2017 and its all weeknum less than = 25

#write.csv(Inquirydf\_media\_dma,"Inquirydf\_media\_dma1.csv")

#now reload the fixed file

Inquirydf\_media\_dma\_updated<- read.csv("Inquirydf\_media\_dma1.csv",header= TRUE, na.strings = c("", "NA")) #still 82 inquiries could not be mapped to key dma , total rec=3062

#now re do the mapping and create a new column to MEDIA\_DMA\_CODE2 which has only inquiry dma's mapped to key dma

str(Inquirydf\_media\_dma\_updated)

Inquirydf\_media\_dma\_updated2<-sqldf("SELECT i.\*, m.MEDIA\_DMA\_CODE as MEDIA\_DMA\_CODE2

FROM Inquirydf\_media\_dma\_updated i left JOIN dma\_formedia\_lookup m

ON i.DMA\_CODE = m.MEDIA\_DMA\_CODE")

str(Inquirydf\_media\_dma\_updated2)

media\_dma\_mismatch<-which(is.na(Inquirydf\_media\_dma\_updated2$MEDIA\_DMA\_CODE2))

length(media\_dma\_mismatch) #82

Inquirydf\_media\_dma\_updated3 <- Inquirydf\_media\_dma\_updated2[-media\_dma\_mismatch,] #2980

#dropping on hold records from old digital sheet

#write.csv(media\_rawdf,"media\_rawdf.csv") --download and remove the on hold records- 764, left - 2806

#media\_rawdf\_updated <- read.csv("media\_rawdf.csv", header = TRUE)

##################################################################################################################################

#grouping all the cost of all mediums by respective week for each year

MediaCostdf\_yrwk <- sqldf('select Year , WeekNum, sum(Cost) as TotalCost,

sum(Cost\_Digital) as DigitalCost,

sum(Cost\_TV) as TVCost,

sum(Cost\_Radio) as RadioCost,

sum(Cost\_Billboard) as BillboardCost,

sum(Cost\_OpenHouse) as OpenHouseCost,

sum(Cost\_Outdoor) as OutdoorCost

from media\_rawdf group by Year,WeekNum')

#grouping all the inquiries of all mediums by respective week for each year

#Inquirydf\_yrwk<-sqldf('select Year, WeekNum,count(\*) as Inquiries from Inquirydf\_media\_dma\_updated2 group by Year,WeekNum') #non matching dma's not removed here as we are mapping to Inquirydf\_media\_dma\_updated2 instead of Inquirydf\_media\_dma\_updated3

Inquirydf\_yrwk<-sqldf('select Year, WeekNum,count(\*) as Inquiries from Inquirydf\_media\_dma\_updated2 group by Year,WeekNum') #non matching dma's not removed here as we are mapping to Inquirydf\_media\_dma\_updated2 instead of Inquirydf\_media\_dma\_updated3

#combining inquiries and medium cost for the respective weeks in a single dataframe , in that process we loose some data from cost as inquiry only till june 2018 but budget is till sep 2018

df\_yrwk <-sqldf("SELECT m.Year, m.WeekNum , m.TotalCost,m.DigitalCost, m.TVCost,m.RadioCost,m.BillboardCost,

m.OpenHouseCost, m.OutdoorCost, i.Inquiries

FROM Inquirydf\_yrwk i JOIN MediaCostdf\_yrwk m

ON i.Year = m.Year

AND i.WeekNum = m.WeekNum")

df\_yrwk[is.na(df\_yrwk)] <- 0

boxplot(df\_yrwk$Inquiries)

#Reallocating budget data to respective week based on

#logic budget for week = 50%of budget of that week +

#25% budget of previous week + 25% budget of previous 2 weeks

df\_yrwk$Estimated\_DigitalCost <- df\_yrwk$DigitalCost\*(0.75)+ lag(df\_yrwk$DigitalCost, default = 0 )\*(.25)

df\_yrwk$Estimated\_BillboardCost<- df\_yrwk$BillboardCost\*(0.5) + lag(df\_yrwk$BillboardCost, default = 0 )\*(.25) + lag(df\_yrwk$BillboardCost,default = 0, 2 )\*(.25)

df\_yrwk$Estimated\_TVCost <- df\_yrwk$TVCost\*(0.5) +lag(df\_yrwk$TVCost,default = 0 )\*(.25)+ lag(df\_yrwk$TVCost,default = 0,2 )\*(.25)

df\_yrwk$Estimated\_RadioCost <- df\_yrwk$RadioCost\*(0.5) + lag(df\_yrwk$RadioCost , default = 0)\*(.25) + lag(df\_yrwk$RadioCost, default = 0,2 )\*(.25)

df\_yrwk$Estimated\_OpenHouseCost<- df\_yrwk$OpenHouseCost\*(0.5) + lag(df\_yrwk$OpenHouseCost , default = 0)\*(.25) + lag(df\_yrwk$OpenHouseCost, default = 0,2 )\*(.25)

df\_yrwk$Estimated\_OutdoorCost<- df\_yrwk$OutdoorCost\*(0.5) + lag(df\_yrwk$OutdoorCost , default = 0)\*(.25) + lag(df\_yrwk$OutdoorCost, default = 0,2 )\*(.25)

df\_yrwk[is.na(df\_yrwk)] <- 0

#Reallocating inquiries data to respective week based on logic Inquiry for week = 50%of inquiry of that week + 25% inquiry of previous week + 25% inquiry of previous 2 weeks

df\_yrwk$Estimated\_Inquiries <- df\_yrwk$Inquiries\*(0.5) + lag(df\_yrwk$Inquiries,default = 0 )\*(.25)+ lag(df\_yrwk$Inquiries, 2 , default = 0)\*(.25)

head(df\_yrwk)

#tail(df\_yrwk)

str(df\_yrwk)

##############################################################################################################################

###############################################DMA mapped####################################################

MediaCostdf\_yrwkdma <- sqldf('select Year , WeekNum,DMA\_CODE, sum(Cost) as TotalCost,

sum(Cost\_Digital) as DigitalCost,

sum(Cost\_TV) as TVCost,

sum(Cost\_Radio) as RadioCost,

sum(Cost\_Billboard) as BillboardCost,

sum(Cost\_OpenHouse) as OpenHouseCost,

sum(Cost\_Outdoor) as OutdoorCost

from media\_rawdf group by Year,WeekNum,DMA\_CODE') # 471 records

#grouping all the inquiries of all mediums by respective week for each year, and each DMA

Inquirydf\_yrwkdma<-sqldf('select Year, WeekNum,MEDIA\_DMA\_CODE2,count(\*) as Inquiries from Inquirydf\_media\_dma\_updated3 group by Year,WeekNum, MEDIA\_DMA\_CODE2') # 397 record

df\_yrwk\_dma <-sqldf("SELECT i.Year, i.WeekNum , m.TotalCost,m.DigitalCost, m.TVCost,m.RadioCost,m.BillboardCost,

m.OpenHouseCost, m.OutdoorCost, i.Inquiries,i.MEDIA\_DMA\_CODE2 as inquirydma, m.DMA\_CODE as mediadma

FROM Inquirydf\_yrwkdma i left JOIN MediaCostdf\_yrwkdma m

ON i.Year = m.Year

AND i.WeekNum = m.WeekNum

AND i.MEDIA\_DMA\_CODE2 = m.DMA\_CODE")# 397 recds

#head(df\_yrwk\_dma)

#tail(df\_yrwk\_dma)

#str(df\_yrwk\_dma)

df\_yrwk\_dma[is.na(df\_yrwk\_dma)] <- 0

# check

sum(df\_yrwk\_dma$Inquiries) # has to match count of Inquirydf\_media\_dma\_updated3 i.e 2980

#P4 <- ggplot(df\_yrwk\_dma, aes(x = DigitalCost, y = Inquiries)) + geom\_line(color="red")

#P5 <- ggplot(df\_yrwk\_dma, aes(x = TVCost, y = Inquiries)) + geom\_line(color="blue")

#P6 <- ggplot(df\_yrwk\_dma, aes(x = RadioCost, y = Inquiries)) + geom\_line(color="green")

#P7 <- ggplot(df\_yrwk\_dma, aes(x = BillboardCost, y = Inquiries)) + geom\_line(color="black")

#grid.arrange(P4,P5,P6,P7,ncol = 2,nrow=2)

#ggplot(df\_yrwk\_all, aes(x = TotalCost, y = Inquiries)) + geom\_line(color="black")

#Reallocating budget data to respective week based on

#logic budget for week = 50%of budget of that week +

#25% budget of previous week + 25% budget of previous 2 weeks on data mapped to key DMA

df\_yrwk\_dma$Estimated\_DigitalCost <- df\_yrwk\_dma$DigitalCost\*(0.75)+ lag(df\_yrwk\_dma$DigitalCost, default = 0 )\*(.25)

df\_yrwk\_dma$Estimated\_BillboardCost<- df\_yrwk\_dma$BillboardCost\*(0.5) + lag(df\_yrwk\_dma$BillboardCost, default = 0 )\*(.25) + lag(df\_yrwk\_dma$BillboardCost,default = 0, 2 )\*(.25)

df\_yrwk\_dma$Estimated\_TVCost <- df\_yrwk\_dma$TVCost\*(0.5) +lag(df\_yrwk\_dma$TVCost,default = 0 )\*(.25)+ lag(df\_yrwk\_dma$TVCost,default = 0,2 )\*(.25)

df\_yrwk\_dma$Estimated\_RadioCost <- df\_yrwk\_dma$RadioCost\*(0.5) + lag(df\_yrwk\_dma$RadioCost , default = 0)\*(.25) + lag(df\_yrwk\_dma$RadioCost, default = 0,2 )\*(.25)

df\_yrwk\_dma$Estimated\_OpenHouseCost<- df\_yrwk\_dma$OpenHouseCost\*(0.5) + lag(df\_yrwk\_dma$OpenHouseCost , default = 0)\*(.25) + lag(df\_yrwk\_dma$OpenHouseCost, default = 0,2 )\*(.25)

df\_yrwk\_dma$Estimated\_OutdoorCost<- df\_yrwk\_dma$OutdoorCost\*(0.5) + lag(df\_yrwk\_dma$OutdoorCost , default = 0)\*(.25) + lag(df\_yrwk\_dma$OutdoorCost, default = 0,2 )\*(.25)

#Reallocating inquiries data to respective week based on logic Inquiry for week = 50%of inquiry of that week + 25% inquiry of previous week + 25% inquiry of previous 2 weeks

df\_yrwk\_dma$Estimated\_Inquiries <- df\_yrwk\_dma$Inquiries\*(0.5) + lag(df\_yrwk\_dma$Inquiries,default = 0 )\*(.25)+ lag(df\_yrwk\_dma$Inquiries, 2 , default = 0)\*(.25)

#----------------training the model with DMA----------------

score\_v4\_DMA.01 = list()

scoretrain\_v4\_DMA.01 = list()

number\_v4\_DMA.01 <- 0

bestscore\_v4\_DMA.01 <-100

score\_v4\_DMA.02 = list()

scoretrain\_v4\_DMA.02 = list()

number\_v4\_DMA.02 <- 0

bestscore\_v4\_DMA.02 <-100

score\_v4\_DMA.03 = list()

scoretrain\_v4\_DMA.03 = list()

number\_v4\_DMA.03 <- 0

bestscore\_v4\_DMA.03 <-100

score\_v4\_DMA.04 = list()

scoretrain\_v4\_DMA.04 = list()

number\_v4\_DMA.04 <- 0

bestscore\_v4\_DMA.04 <-100

score\_v4\_DMA.05 = list()

scoretrain\_v4\_DMA.05 = list()

number\_v4\_DMA.05 <- 0

bestscore\_v4\_DMA.05 <-100

for(i in 1:100)

{

#create indexes for train data

train.index\_v4\_DMA <- createDataPartition(df\_yrwk\_dma$Inquiries, p = 0.8, list = FALSE)

#create train data

train.data\_v4\_DMA <- df\_yrwk\_dma[train.index\_v4\_DMA, ]

#create test data

test.data\_v4\_DMA<- df\_yrwk\_dma[-train.index\_v4\_DMA, ]

#model without adstock and without take changes in media budget/Inquiries

modFitv4\_DMA.01<- lm(Inquiries~DigitalCost + TVCost +RadioCost+ BillboardCost , data =train.data\_v4\_DMA)

# Make predictions

predictionsv4\_DMA.01 <- modFitv4\_DMA.01 %>% predict(test.data\_v4\_DMA)

# Model performance # (a) Prediction error, RMSE

score\_v4\_DMA.01[i] <- RMSE(predictionsv4\_DMA.01, test.data\_v4\_DMA$Inquiries)

if(as.numeric(score\_v4\_DMA.01[i]) < as.numeric(bestscore\_v4\_DMA.01 ))

{

bestscore\_v4\_DMA.01 = score\_v4\_DMA.01[i]

bestmodel\_v4\_DMA.01 <- modFitv4\_DMA.01

number\_v4\_DMA.01 <- i

predictionsv4\_DMA.01\_train <- modFitv4\_DMA.01 %>% predict(train.data\_v4\_DMA)

scoretrain\_v4\_DMA.01[i]<- RMSE(predictionsv4\_DMA.01\_train , train.data\_v4\_DMA$Inquiries)

maev4\_DMA.01 <- abs(mean((predictionsv4\_DMA.01- test.data\_v4\_DMA$Inquiries))) #Mean absolute error of test data

}

modFitv4\_DMA.02 <-lm(Inquiries~Estimated\_DigitalCost + Estimated\_BillboardCost + Estimated\_TVCost + Estimated\_RadioCost , data =train.data\_v4\_DMA)

predictionsv4\_DMA.02 <- modFitv4\_DMA.02 %>% predict(test.data\_v4\_DMA)

# Model performance # (a) Prediction error, RMSE

score\_v4\_DMA.02[i] <- RMSE(predictionsv4\_DMA.02, test.data\_v4\_DMA$Inquiries)

if(as.numeric(score\_v4\_DMA.02[i]) < as.numeric(bestscore\_v4\_DMA.02 ))

{

bestscore\_v4\_DMA.02 = score\_v4\_DMA.02[i]

bestmodel\_v4\_DMA.02 <- modFitv4\_DMA.02

number\_v4\_DMA.02 <- i

predictionsv4\_DMA.02\_train <- modFitv4\_DMA.02 %>% predict(train.data\_v4\_DMA)

scoretrain\_v4\_DMA.02[i]<- RMSE(predictionsv4\_DMA.02\_train , train.data\_v4\_DMA$Inquiries)

maev4\_DMA.02 <- abs(mean((predictionsv4\_DMA.02-test.data\_v4\_DMA$Inquiries))) #Mean absolute error

}

modFitv4\_DMA.03 <-lm(Estimated\_Inquiries~DigitalCost + BillboardCost + TVCost +RadioCost, data =train.data\_v4\_DMA)

predictionsv4\_DMA.03 <- modFitv4\_DMA.03 %>% predict(test.data\_v4\_DMA)

# Model performance # (a) Prediction error, RMSE

score\_v4\_DMA.03[i] <- RMSE(predictionsv4\_DMA.03, test.data\_v4\_DMA$Inquiries)

if(as.numeric(score\_v4\_DMA.03[i]) < as.numeric(bestscore\_v4\_DMA.03))

{

bestscore\_v4\_DMA.03 = score\_v4\_DMA.03[i]

bestmodel\_v4\_DMA.03 <- modFitv4\_DMA.03

number\_v4\_DMA.03 <- i

predictionsv4\_DMA.03\_train <- modFitv4\_DMA.03 %>% predict(train.data\_v4\_DMA)

scoretrain\_v4\_DMA.03[i]<- RMSE(predictionsv4\_DMA.03\_train , train.data\_v4\_DMA$Inquiries)

maev4\_DMA.03 <- abs(mean((predictionsv4\_DMA.03 - test.data\_v4\_DMA$Inquiries))) #Mean absolute error

}

modFitv4\_DMA.04 <-lm(Estimated\_Inquiries~Estimated\_DigitalCost + Estimated\_BillboardCost + Estimated\_TVCost + Estimated\_RadioCost , data =train.data\_v4\_DMA)

predictionsv4\_DMA.04 <- modFitv4\_DMA.04 %>% predict(test.data\_v4\_DMA)

# Model performance # (a) Prediction error, RMSE

score\_v4\_DMA.04[i] <- RMSE(predictionsv4\_DMA.04, test.data\_v4\_DMA$Inquiries)

if(as.numeric(score\_v4\_DMA.04[i]) < as.numeric(bestscore\_v4\_DMA.04))

{

bestscore\_v4\_DMA.04 = score\_v4\_DMA.04[i]

bestmodel\_v4\_DMA.04 <- modFitv4\_DMA.04

number\_v4\_DMA.04 <- i

predictionsv4\_DMA.04\_train <- modFitv4\_DMA.04 %>% predict(train.data\_v4\_DMA)

scoretrain\_v4\_DMA.04[i]<- RMSE(predictionsv4\_DMA.04\_train , train.data\_v4\_DMA$Inquiries)

maev4\_DMA.04 <- abs(mean((predictionsv4\_DMA.04- test.data\_v4\_DMA$Inquiries))) #Mean absolute error

}

modFitv4\_DMA.05 <- lm(Inquiries~DigitalCost + TVCost +RadioCost+ BillboardCost+ factor(inquirydma),data =train.data\_v4\_DMA)

predictionsv4\_DMA.05 <- predict(modFitv4\_DMA.05 ,test.data\_v4\_DMA)

# Model performance # (a) Prediction error, RMSE

score\_v4\_DMA.05[i] <- RMSE(predictionsv4\_DMA.05, test.data\_v4\_DMA$Inquiries)

if(as.numeric(score\_v4\_DMA.05[i]) < as.numeric(bestscore\_v4\_DMA.05))

{

bestscore\_v4\_DMA.05 = score\_v4\_DMA.05[i]

bestmodel\_v4\_DMA.05 <- modFitv4\_DMA.05

number\_v4\_DMA.05 <- i

predictionsv4\_DMA.05\_train <- modFitv4\_DMA.05 %>% predict(train.data\_v4\_DMA)

scoretrain\_v4\_DMA.05[i]<- RMSE(predictionsv4\_DMA.05\_train , train.data\_v4\_DMA$Inquiries)

maev4\_DMA.05 <- abs(mean((predictionsv4\_DMA.05- test.data\_v4\_DMA$Inquiries))) #Mean absolute error

}

rm(train.data\_v4\_DMA)

rm(test.data\_v4\_DMA)

}

#################################### Analysing each of the 4 models ##################################################################################

############################################################### modFitv4\_DMA.01 ##############################################################################

print(bestscore\_v4\_DMA.01)

print(number\_v4\_DMA.01)

summary(bestmodel\_v4\_DMA.01)

scoretrain\_v4\_DMA.01[number\_v4\_DMA.01]

predictions\_full\_01 <- predict(bestmodel\_v4\_DMA.01, df\_yrwk\_dma)

RMSE(predictions\_full\_01, df\_yrwk\_dma$Inquiries)

mean((df\_yrwk\_dma$Inquiries-predictions\_full\_01)) #Mean absolute error

############################################################### modFitv4\_DMA.02 ##############################################################################

print(bestscore\_v4\_DMA.02)

print(number\_v4\_DMA.02)

summary(bestmodel\_v4\_DMA.02)

scoretrain\_v4\_DMA.02[number\_v4\_DMA.02]

predictions\_full\_02 <- predict(bestmodel\_v4\_DMA.02, df\_yrwk\_dma)

RMSE(predictions\_full\_02, df\_yrwk\_dma$Inquiries)

mean((df\_yrwk\_dma$Inquiries-predictions\_full\_02)) #Mean absolute error

############################################################### modFitv4\_DMA.03 ##############################################################################

print(bestscore\_v4\_DMA.03)

print(number\_v4\_DMA.03)

summary(bestmodel\_v4\_DMA.03)

scoretrain\_v4\_DMA.03[number\_v4\_DMA.03]

predictions\_full\_03 <- predict(bestmodel\_v4\_DMA.03, df\_yrwk\_dma)

RMSE(predictions\_full\_03, df\_yrwk\_dma$Inquiries)

mean((df\_yrwk\_dma$Inquiries-predictions\_full\_03)) #Mean absolute error

############################################################### modFitv4\_DMA.04 ##############################################################################

print(bestscore\_v4\_DMA.04)

print(number\_v4\_DMA.04)

summary(bestmodel\_v4\_DMA.04)

scoretrain\_v4\_DMA.04[number\_v4\_DMA.04]

predictions\_full\_04 <- predict(bestmodel\_v4\_DMA.04, df\_yrwk\_dma)

RMSE(predictions\_full\_04, df\_yrwk\_dma$Inquiries)

mean((df\_yrwk\_dma$Inquiries-predictions\_full\_04)) #Mean absolute error

mean((df\_yrwk\_dma$Estimated\_Inquiries-predictions\_full\_04))

##############################################################################################################################################

print(bestscore\_v4\_DMA.05)

print(number\_v4\_DMA.05)

summary(bestmodel\_v4\_DMA.05)

scoretrain\_v4\_DMA.04[number\_v4\_DMA.05]

predictions\_full\_05 <- predict(bestmodel\_v4\_DMA.05, df\_yrwk\_dma)

RMSE(predictions\_full\_05, df\_yrwk\_dma$Inquiries)

mean((df\_yrwk\_dma$Inquiries-predictions\_full\_05)) #Mean absolute error

#######################################################################################################################################

#using the best model to predict entire dataset

predictions\_full <- predict(bestmodel, df\_yrwk\_dma)

error\_per <- sqrt(mean((predictions\_full-df\_yrwk\_dma$Inquiries)^2))

mae <- mean((df\_yrwk\_dma$Inquiries-predictions\_full)) #Mean absolute error

mse <- mean((df\_yrwk\_dma$Inquiries-predictions\_full)^2)

mse\_baseline <- mean((df\_yrwk\_dma$Inquiries-mean(predictions\_full))^2)

RMSE(predictions\_full,df\_yrwk\_dma$Inquiries)

r2 <- 1- (mse/mse\_baseline) # R² is the ratio between how good our model is vs how good is the naive mean model.The maximum value of R² is 1 but minimum can be minus infinity.

boxplot(df\_yrwk\_dma$Inquiries)

summary(df\_yrwk\_dma$Inquiries)

####visualization can ignore as done for understanding #######################################

comparison\_tbl <- data.frame( "Inquiries"=df\_yrwk\_dma$Inquiries, "predicted" =predictions\_full\_01, "residual"=(predictions\_full-df\_yrwk\_dma$Inquiries),

DigitalCost=df\_yrwk\_dma$DigitalCost,TVCost=df\_yrwk\_dma$TVCost, RadioCost=df\_yrwk\_dma$RadioCost, BillboardCost=df\_yrwk\_dma$BillboardCost)

# plotting the observerd/actual inquiries

ggplot(comparison\_tbl, aes(x = DigitalCost, y = Inquiries)) + # Set up canvas with outcome variable on y-axis

geom\_point() # Plot the actual points

ggplot(comparison\_tbl, aes(x = DigitalCost, y = Inquiries)) + # Set up canvas with outcome variable on y-axis

geom\_point() +

geom\_point(aes(y = predicted), shape = 2) # Add the predicted values

#connect actual data points with their corresponding predicted value using geom\_segment():

ggplot(comparison\_tbl, aes(x = DigitalCost, y = Inquiries)) +

geom\_smooth(method = "lm", se = FALSE, color = "green") +

geom\_segment(aes(xend = DigitalCost, yend = predicted), alpha = .2) +# alpha to fade lines

geom\_point() +

geom\_point(aes(y = predicted), shape = 2)+

theme\_bw() # Add theme for cleaner look

#residuals

ggplot(comparison\_tbl, aes(x = DigitalCost, y = Inquiries)) +

geom\_smooth(method = "lm", se = FALSE, color = "green") +

geom\_segment(aes(xend = DigitalCost, yend = predicted), alpha = .2) +# alpha to fade lines

geom\_point(aes(color = abs(residual))) + # Alpha mapped to abs(residuals)

scale\_color\_gradient2(low = "blue", mid = "orange", high = "red") +

guides(color = FALSE) + # color legend removed

geom\_point(aes(y = predicted), shape = 2) +

theme\_bw()

ggplot(comparison\_tbl, aes(x = DigitalCost, y = Inquiries)) +

geom\_segment(aes(xend = DigitalCost, yend = predicted)) +

geom\_point() +

geom\_point(aes(y = predicted), shape = 2)

ggplot(comparison\_tbl, aes(x = DigitalCost, y = Inquiries)) +

#geom\_segment(aes(xend = DigitalCost, yend = predicted), alpha = .2) + # Lines to connect points

geom\_point() + # Points of actual values

geom\_point(aes(y = predicted), shape = 2, color ="red") + # Points of predicted values

theme\_bw()

ggplot(comparison\_tbl, aes(x = DigitalCost, y = Inquiries)) +

geom\_segment(aes(xend = DigitalCost, yend = predicted), alpha = .2) + # Lines to connect points

geom\_point(aes(color = residual)) +

scale\_color\_gradient2(low = "blue", mid = "dark green", high = "red") +

guides(color = FALSE) +

#geom\_point() + # Points of actual values

geom\_point(aes(y = predicted), shape = 2, color ="black") + # Points of predicted values

theme\_bw()

#----------------running the model with adstock, and nlsm----------------------------------------------#

score\_v11\_DMA.Ad = list()

scoretrain\_v11\_DMA.Ad = list()

number\_v11\_DMA.Ad <- 0

bestscore\_v11\_DMA.Ad <-100

bestrate\_v11\_DMA.Ad<- 0

score\_v11\_DMA.Ad\_nls = list()

scoretrain\_v11\_DMA.Ad\_nls = list()

number\_v11\_DMA.Ad\_nls <- 0

bestscore\_v11\_DMA.Ad\_nls <-100

score\_v11\_DMA\_nls = list()

scoretrain\_v11\_DMA\_nls = list()

number\_v11\_DMA\_nls <- 0

bestscore\_v11\_DMA\_nls <-100

for(i in 1:100)

{

#create indexes for train data

train.index\_v11\_DMA.Ad <- createDataPartition(df\_yrwk\_dma$Inquiries, p = 0.8, list = FALSE)

#create train data

train.data\_v11\_DMA.Ad <- df\_yrwk\_dma[train.index\_v11\_DMA.Ad, ]

#create test data

test.data\_v11\_DMA.Ad <- df\_yrwk\_dma[-train.index\_v11\_DMA.Ad, ]

#build model

train.data\_v11\_DMA.Ad[is.na(train.data\_v11\_DMA.Ad)] <- 0

model\_v11\_DMA.Ad <- nlsLM(data= train.data\_v11\_DMA.Ad, Inquiries~ b0 + b1 \* adstock(DigitalCost,rater) +

b2 \* adstock(TVCost, rater) +

b3 \* adstock(RadioCost, rater) +

b4 \* adstock(BillboardCost, rater),

algorithm = "LM",

start = c(b0=1, b1= 1, b2= 1, b3=1, b4=1, rater=0),

lower = c(b0=-Inf, b1=-Inf, b2=-Inf,b3=-Inf, b4=-Inf,rater=0),

upper = c(b0= Inf, b1= Inf, b2= Inf, b3=Inf, b4=Inf, rater=1))

newrate <- summary(model\_v11\_DMA.Ad)$coefficients[6,1]

#summary(model\_v11\_DMA.Ad)

train.data\_v11\_DMA.Ad$DigitalCostAd <- adstock(train.data\_v11\_DMA.Ad $DigitalCost,newrate)

train.data\_v11\_DMA.Ad$TVCostAd <-adstock(train.data\_v11\_DMA.Ad$TVCost,newrate)

train.data\_v11\_DMA.Ad$RadioCostAd <- adstock(train.data\_v11\_DMA.Ad$RadioCost, newrate)

train.data\_v11\_DMA.Ad$BillboardCostAd <- adstock(train.data\_v11\_DMA.Ad$BillboardCost, newrate)

train.data\_v11\_DMA.Ad[is.na(train.data\_v11\_DMA.Ad)] <- 0

#lm with adstock and DMA

modFit\_v11\_DMA.Ad <-lm(Inquiries~DigitalCostAd +

TVCostAd +

RadioCostAd +

BillboardCostAd ,

data = train.data\_v11\_DMA.Ad)

#nls with adstock and DMA, just apply the model\_v11\_DMA.Ad on test data

modFit\_v11\_DMA.Ad\_nls <- nlsLM(data= train.data\_v11\_DMA.Ad, Inquiries~ b0 + b1 \* DigitalCostAd +

b2 \* TVCostAd +

b3 \* RadioCostAd +

b4 \* BillboardCostAd,

algorithm = "LM",

start = c(b0=1, b1= 1, b2= 1, b3=1, b4=1),

lower = c(b0=-Inf, b1=-Inf, b2=-Inf,b3=-Inf, b4=-Inf),

upper = c(b0= Inf, b1= Inf, b2= Inf, b3=Inf, b4=Inf))

# pure nls with no adstock but DMA

modFit\_v11\_DMA\_nls <- nlsLM(data= train.data\_v11\_DMA.Ad, Inquiries~ b0 + b1 \* DigitalCost +

b2 \* TVCost +

b3 \* RadioCost +

b4 \* BillboardCost,

algorithm = "LM",

start = c(b0=1, b1= 1, b2= 1, b3=1, b4=1),

lower = c(b0=-Inf, b1=-Inf, b2=-Inf,b3=-Inf, b4=-Inf),

upper = c(b0= Inf, b1= Inf, b2= Inf, b3=Inf, b4=Inf))

#predict(modFit\_v5\_lm, new\_data= train.data\_v5, interval = c("confidence"), level = .95)

#print(summary(model\_v5))

#e build the test data

test.data\_v11\_DMA.Ad$DigitalCostAd <- adstock(test.data\_v11\_DMA.Ad$DigitalCost,newrate)

test.data\_v11\_DMA.Ad$TVCostAd <- adstock(test.data\_v11\_DMA.Ad$TVCost,newrate)

test.data\_v11\_DMA.Ad$RadioCostAd <- adstock(test.data\_v11\_DMA.Ad$RadioCost,newrate)

test.data\_v11\_DMA.Ad$BillboardCostAd <- adstock(test.data\_v11\_DMA.Ad$BillboardCost,newrate)

test.data\_v11\_DMA.Ad[is.na(test.data\_v11\_DMA.Ad)] <- 0

# Make predictions for adstock, DMA,using lm method

predictions\_v11\_DMA.Ad <- modFit\_v11\_DMA.Ad %>% predict(test.data\_v11\_DMA.Ad)

# Model performance

# (a) Prediction error, RMSE

score\_v11\_DMA.Ad[i]=RMSE( predictions\_v11\_DMA.Ad, test.data\_v11\_DMA.Ad$Inquiries)

if(as.numeric(score\_v11\_DMA.Ad [i]) < as.numeric(bestscore\_v11\_DMA.Ad))

{

bestscore\_v11\_DMA.Ad = score\_v11\_DMA.Ad[i]

bestmodel\_v11\_DMA.Ad <- modFit\_v11\_DMA.Ad

number\_v11\_DMA.Ad<- i

bestrate\_v11\_DMA.Ad<- newrate

predictions\_v11\_DMA.Ad\_train <-modFit\_v11\_DMA.Ad%>% predict(train.data\_v11\_DMA.Ad)

scoretrain\_v11\_DMA.Ad[i] <- RMSE(predictions\_v11\_DMA.Ad\_train, train.data\_v11\_DMA.Ad$Inquiries)

mae\_v11\_DMA.Ad\_test<- abs(mean((predictions\_v11\_DMA.Ad-test.data\_v11\_DMA.Ad$Inquiries))) #Mean absolute error

}

# Make predictions for adstock, DMA,using nlslm method

#summary(modFitv4\_DMA.08)

predictions\_v11\_DMA.Ad\_nls <- modFit\_v11\_DMA.Ad\_nls %>% predict(test.data\_v11\_DMA.Ad)

# Model performance

# (a) Prediction error, RMSE

score\_v11\_DMA.Ad\_nls[i]=RMSE(predictions\_v11\_DMA.Ad\_nls, test.data\_v11\_DMA.Ad$Inquiries)

if(as.numeric(score\_v11\_DMA.Ad\_nls[i]) < as.numeric(bestscore\_v11\_DMA.Ad\_nls))

{

bestscore\_v11\_DMA.Ad\_nls = score\_v11\_DMA.Ad\_nls[i]

bestmodel\_v11\_DMA.Ad\_nls <- modFit\_v11\_DMA.Ad\_nls

number\_v11\_DMA.Ad\_nls<- i

predictions\_v11\_DMA.Ad\_nls\_train <-modFit\_v11\_DMA.Ad\_nls%>% predict(train.data\_v11\_DMA.Ad)

scoretrain\_v11\_DMA.Ad\_nls[i] <- RMSE(predictions\_v11\_DMA.Ad\_nls\_train, train.data\_v11\_DMA.Ad$Inquiries)

mae\_v11\_DMA.Ad\_nls\_test<- abs(mean((predictions\_v11\_DMA.Ad\_nls-test.data\_v11\_DMA.Ad$Inquiries))) #Mean absolute error

}

# Make predictions no adstock, but DMA and method =nls

#summary(modFitv4\_DMA.08)

predictions\_v11\_DMA\_nls <- modFit\_v11\_DMA\_nls%>% predict(test.data\_v11\_DMA.Ad)

# Model performance

# (a) Prediction error, RMSE

score\_v11\_DMA\_nls[i]=RMSE(predictions\_v11\_DMA\_nls, test.data\_v11\_DMA.Ad$Inquiries)

#predictionsv5\_train <- model %>% predict(train.data\_v4\_DMA)

#scoretrain[i]<- RMSE(predictionsv4\_DMA.10\_train , train.data\_v4\_DMA$Inquiries)

if(as.numeric(score\_v11\_DMA\_nls[i]) < as.numeric(bestscore\_v11\_DMA\_nls))

{

bestscore\_v11\_DMA\_nls = score\_v11\_DMA\_nls[i]

bestmodel\_v11\_DMA\_nls<- modFit\_v11\_DMA\_nls

number\_v11\_DMA\_nls<- i

predictions\_v11\_DMA\_nls\_train <- modFit\_v11\_DMA\_nls%>% predict(train.data\_v11\_DMA.Ad)

scoretrain\_v11\_DMA\_nls[i] <- RMSE(predictions\_v11\_DMA\_nls\_train, train.data\_v11\_DMA.Ad$Inquiries)

mae\_v11\_DMA\_nls\_test<- abs(mean((predictions\_v11\_DMA\_nls-test.data\_v11\_DMA.Ad$Inquiries))) #Mean absolute error

}

rm(train.data\_v11\_DMA.Ad)

rm(test.data\_v11\_DMA.Ad)

}

# Make predictions for adstock, DMA,using lm method

summary(bestmodel\_v11\_DMA.Ad)

print(number\_v11\_DMA.Ad)

print(score\_v11\_DMA.Ad[number\_v11\_DMA.Ad])

scoretrain\_v11\_DMA.Ad[number\_v11\_DMA.Ad]

bestscore\_v11\_DMA.Ad

bestrate\_v11\_DMA.Ad

#based on best model and bestrate we got we see only digital cost is the best related to inquiries and recreate the model with Adstock digital cost

df\_yrwk\_dma\_new <- df\_yrwk\_dma

df\_yrwk\_dma\_new$DigitalCostAd <- adstock(df\_yrwk\_dma\_new$DigitalCost,bestrate\_v11\_DMA.Ad)

df\_yrwk\_dma\_new$TVCostAd <-adstock(df\_yrwk\_dma\_new$TVCost,bestrate\_v11\_DMA.Ad)

df\_yrwk\_dma\_new$RadioCostAd <- adstock(df\_yrwk\_dma\_new$RadioCost, bestrate\_v11\_DMA.Ad)

df\_yrwk\_dma\_new$BillboardCostAd <- adstock(df\_yrwk\_dma\_new$BillboardCost, bestrate\_v11\_DMA.Ad)

df\_yrwk\_dma\_new[is.na(df\_yrwk\_dma\_new)] <- 0

predictions\_full\_v11\_DMA.Ad <-predict(bestmodel\_v11\_DMA.Ad, df\_yrwk\_dma\_new )

RMSE(predictions\_full\_v11\_DMA.Ad, df\_yrwk\_dma\_new$Inquiries)

mea\_full\_v6.1 <- abs(mean(predictions\_full\_v11\_DMA.Ad - df\_yrwk\_dma\_new$Inquiries))

# Make predictions for adstock, DMA,using nlslm method

summary(bestmodel\_v11\_DMA.Ad\_nls)

number\_v11\_DMA.Ad\_nls

score\_v11\_DMA.Ad\_nls[number\_v11\_DMA.Ad\_nls]

scoretrain\_v11\_DMA.Ad\_nls[number\_v11\_DMA.Ad\_nls]

bestscore\_v11\_DMA.Ad\_nls

#no adstock, but DMA and method =nls

summary(bestmodel\_v11\_DMA\_nls)

number\_v11\_DMA\_nls

score\_v11\_DMA\_nls[number\_v11\_DMA\_nls]

scoretrain\_v11\_DMA\_nls[number\_v11\_DMA\_nls]

bestscore\_v11\_DMA\_nls

## R results

######## Analyzing each of the 4 models ##################################################################################

>

>

>

> ######### modFitv4\_DMA.01 ####################################

>

> print(bestscore\_v4\_DMA.01)

[[1]]

[1] 4.249366

> print(number\_v4\_DMA.01)

[1] 23

> summary(bestmodel\_v4\_DMA.01)

Call:

lm(formula = Inquiries ~ DigitalCost + TVCost + RadioCost + BillboardCost,

data = train.data\_v4\_DMA)

Residuals:

Min 1Q Median 3Q Max

-9.313 -4.254 -1.476 2.180 69.389

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 5.967e+00 6.237e-01 9.568 < 2e-16 \*\*\*

DigitalCost 1.183e-03 2.743e-04 4.313 2.16e-05 \*\*\*

TVCost 2.770e-06 7.159e-05 0.039 0.969

RadioCost 3.194e-06 1.034e-04 0.031 0.975

BillboardCost -1.108e-03 8.102e-04 -1.367 0.172

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.376 on 314 degrees of freedom

Multiple R-squared: 0.08777, Adjusted R-squared: 0.07615

F-statistic: 7.553 on 4 and 314 DF, p-value: 8.051e-06

> scoretrain\_v4\_DMA.01[number\_v4\_DMA.01]

[[1]]

[1] 7.317622

>

>

> predictions\_full\_01 <- predict(bestmodel\_v4\_DMA.01, df\_yrwk\_dma)

> RMSE(predictions\_full\_01, df\_yrwk\_dma$Inquiries)

[1] 6.824563

> mean((df\_yrwk\_dma$Inquiries-predictions\_full\_01)) #Mean absolute error

[1] -0.1521071

> ########### modFitv4\_DMA.02 ##################################

>

> print(bestscore\_v4\_DMA.02)

[[1]]

[1] 4.415686

> print(number\_v4\_DMA.02)

[1] 23

> summary(bestmodel\_v4\_DMA.02)

Call:

lm(formula = Inquiries ~ Estimated\_DigitalCost + Estimated\_BillboardCost +

Estimated\_TVCost + Estimated\_RadioCost, data = train.data\_v4\_DMA)

Residuals:

Min 1Q Median 3Q Max

-8.086 -4.580 -1.452 2.438 70.216

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 6.302e+00 7.417e-01 8.497 7.93e-16 \*\*\*

Estimated\_DigitalCost 9.316e-04 3.328e-04 2.799 0.00544 \*\*

Estimated\_BillboardCost -2.334e-04 1.510e-03 -0.155 0.87727

Estimated\_TVCost 3.440e-06 1.182e-04 0.029 0.97680

Estimated\_RadioCost -4.755e-05 1.541e-04 -0.309 0.75789

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.611 on 314 degrees of freedom

Multiple R-squared: 0.02851, Adjusted R-squared: 0.01614

F-statistic: 2.304 on 4 and 314 DF, p-value: 0.05837

> scoretrain\_v4\_DMA.02[number\_v4\_DMA.02]

[[1]]

[1] 7.551573

>

> predictions\_full\_02 <- predict(bestmodel\_v4\_DMA.02, df\_yrwk\_dma)

> RMSE(predictions\_full\_02, df\_yrwk\_dma$Inquiries)

[1] 7.046488

> mean((df\_yrwk\_dma$Inquiries-predictions\_full\_02)) #Mean absolute error

[1] -0.12836

> ############ modFitv4\_DMA.03 #################################

>

> print(bestscore\_v4\_DMA.03)

[[1]]

[1] 4.312785

> print(number\_v4\_DMA.03)

[1] 23

> summary(bestmodel\_v4\_DMA.03)

Call:

lm(formula = Estimated\_Inquiries ~ DigitalCost + BillboardCost +

TVCost + RadioCost, data = train.data\_v4\_DMA)

Residuals:

Min 1Q Median 3Q Max

-6.458 -2.437 -0.525 1.576 34.306

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 6.949e+00 3.491e-01 19.908 <2e-16 \*\*\*

DigitalCost 4.451e-04 1.535e-04 2.899 0.004 \*\*

BillboardCost -3.135e-04 4.535e-04 -0.691 0.490

TVCost 9.357e-06 4.007e-05 0.234 0.816

RadioCost -4.827e-05 5.790e-05 -0.834 0.405

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.128 on 314 degrees of freedom

Multiple R-squared: 0.04244, Adjusted R-squared: 0.03024

F-statistic: 3.479 on 4 and 314 DF, p-value: 0.008463

> scoretrain\_v4\_DMA.03[number\_v4\_DMA.03]

[[1]]

[1] 7.460572

>

>

> predictions\_full\_03 <- predict(bestmodel\_v4\_DMA.03, df\_yrwk\_dma)

> RMSE(predictions\_full\_03, df\_yrwk\_dma$Inquiries)

[1] 6.955488

> mean((df\_yrwk\_dma$Inquiries-predictions\_full\_03)) #Mean absolute error

[1] -0.008409851

>

> ########## modFitv4\_DMA.04 ###################################

> print(bestscore\_v4\_DMA.04)

[[1]]

[1] 4.499786

> print(number\_v4\_DMA.04)

[1] 23

> summary(bestmodel\_v4\_DMA.04)

Call:

lm(formula = Estimated\_Inquiries ~ Estimated\_DigitalCost + Estimated\_BillboardCost +

Estimated\_TVCost + Estimated\_RadioCost, data = train.data\_v4\_DMA)

Residuals:

Min 1Q Median 3Q Max

-6.107 -2.695 -0.561 1.698 34.402

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 6.887e+00 4.063e-01 16.949 <2e-16 \*\*\*

Estimated\_DigitalCost 4.061e-04 1.823e-04 2.227 0.0266 \*

Estimated\_BillboardCost 8.214e-04 8.273e-04 0.993 0.3215

Estimated\_TVCost 1.999e-05 6.474e-05 0.309 0.7577

Estimated\_RadioCost -1.006e-04 8.443e-05 -1.191 0.2345

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.17 on 314 degrees of freedom

Multiple R-squared: 0.02318, Adjusted R-squared: 0.01074

F-statistic: 1.863 on 4 and 314 DF, p-value: 0.1167

> scoretrain\_v4\_DMA.04[number\_v4\_DMA.04]

[[1]]

[1] 7.59613

>

>

> predictions\_full\_04 <- predict(bestmodel\_v4\_DMA.04, df\_yrwk\_dma)

> RMSE(predictions\_full\_04, df\_yrwk\_dma$Inquiries)

[1] 7.095255

> mean((df\_yrwk\_dma$Inquiries-predictions\_full\_04)) #Mean absolute error

[1] 0.003787329

> mean((df\_yrwk\_dma$Estimated\_Inquiries-predictions\_full\_04))

[1] -0.0188827

> ##############################################################################################################################################

> print(bestscore\_v4\_DMA.05)

[[1]]

[1] 3.528162

> print(number\_v4\_DMA.05)

[1] 38

> summary(bestmodel\_v4\_DMA.05)

Call:

lm(formula = Inquiries ~ DigitalCost + TVCost + RadioCost + BillboardCost +

factor(inquirydma), data = train.data\_v4\_DMA)

Residuals:

Min 1Q Median 3Q Max

-11.878 -3.190 -0.828 1.810 64.489

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 4.190e+00 1.219e+00 3.438 0.000667 \*\*\*

DigitalCost 4.042e-04 3.996e-04 1.011 0.312642

TVCost -1.408e-06 6.998e-05 -0.020 0.983964

RadioCost -8.474e-05 1.001e-04 -0.846 0.398035

BillboardCost 6.050e-05 1.201e-03 0.050 0.959854

factor(inquirydma)509 6.897e+00 1.639e+00 4.208 3.38e-05 \*\*\*

factor(inquirydma)515 3.823e-01 2.294e+00 0.167 0.867756

factor(inquirydma)527 8.418e+00 1.691e+00 4.978 1.07e-06 \*\*\*

factor(inquirydma)529 -7.507e-02 1.852e+00 -0.041 0.967696

factor(inquirydma)582 -2.527e+00 2.087e+00 -1.211 0.226796

factor(inquirydma)588 1.865e+00 1.639e+00 1.138 0.256154

factor(inquirydma)602 6.470e+00 2.521e+00 2.567 0.010745 \*

factor(inquirydma)649 -2.003e+00 1.851e+00 -1.082 0.279998

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.561 on 306 degrees of freedom

Multiple R-squared: 0.2849, Adjusted R-squared: 0.2568

F-statistic: 10.16 on 12 and 306 DF, p-value: < 2.2e-16

> scoretrain\_v4\_DMA.04[number\_v4\_DMA.05]

[[1]]

NULL

>

>

> predictions\_full\_05 <- predict(bestmodel\_v4\_DMA.05, df\_yrwk\_dma)

> RMSE(predictions\_full\_05, df\_yrwk\_dma$Inquiries)

[1] 5.969103

> mean((df\_yrwk\_dma$Inquiries-predictions\_full\_05)) #Mean absolute error

[1] -0.00623188

> #######################################################################################################################################

|  |
| --- |
| > # Make predictions for adstock, DMA,using lm method  > summary(bestmodel\_v11\_DMA.Ad)  Call:  lm(formula = Inquiries ~ DigitalCostAd + TVCostAd + RadioCostAd +  BillboardCostAd, data = train.data\_v11\_DMA.Ad)  Residuals:  Min 1Q Median 3Q Max  -9.562 -4.087 -1.504 2.114 69.215  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) 6.087e+00 6.058e-01 10.048 < 2e-16 \*\*\*  DigitalCostAd 1.207e-03 2.548e-04 4.737 3.29e-06 \*\*\*  TVCostAd 2.028e-05 7.350e-05 0.276 0.7828  RadioCostAd -2.500e-05 1.039e-04 -0.241 0.8100  BillboardCostAd -1.481e-03 8.209e-04 -1.803 0.0723 .  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 7.328 on 314 degrees of freedom  Multiple R-squared: 0.1052, Adjusted R-squared: 0.09385  F-statistic: 9.234 on 4 and 314 DF, p-value: 4.582e-07  > print(number\_v11\_DMA.Ad)  [1] 67  > print(score\_v11\_DMA.Ad[number\_v11\_DMA.Ad])  [[1]]  [1] 4.427753  > scoretrain\_v11\_DMA.Ad[number\_v11\_DMA.Ad]  [[1]]  [1] 7.270819  > bestscore\_v11\_DMA.Ad  [[1]]  [1] 4.427753  > bestrate\_v11\_DMA.Ad  [1] 0  > # Make predictions for adstock, DMA,using nlslm method  > summary(bestmodel\_v11\_DMA.Ad\_nls)  Formula: Inquiries ~ b0 + b1 \* DigitalCostAd + b2 \* TVCostAd + b3 \* RadioCostAd +  b4 \* BillboardCostAd  Parameters:  Estimate Std. Error t value Pr(>|t|)  b0 5.973e+00 6.002e-01 9.952 < 2e-16 \*\*\*  b1 1.231e-03 2.624e-04 4.691 4.07e-06 \*\*\*  b2 7.190e-06 7.159e-05 0.100 0.920  b3 -2.613e-05 1.037e-04 -0.252 0.801  b4 -1.068e-03 8.155e-04 -1.309 0.191  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 7.319 on 314 degrees of freedom  Number of iterations to convergence: 2  Achieved convergence tolerance: 1.49e-08  > number\_v11\_DMA.Ad\_nls  [1] 77  > score\_v11\_DMA.Ad\_nls[number\_v11\_DMA.Ad\_nls]  [[1]]  [1] 4.675362  > scoretrain\_v11\_DMA.Ad\_nls[number\_v11\_DMA.Ad\_nls]  [[1]]  [1] 7.261547  > bestscore\_v11\_DMA.Ad\_nls  [[1]]  [1] 4.675362  >  >  > # #no adstock, but DMA and method =nls  > summary(bestmodel\_v11\_DMA\_nls)  Formula: Inquiries ~ b0 + b1 \* DigitalCost + b2 \* TVCost + b3 \* RadioCost +  b4 \* BillboardCost  Parameters:  Estimate Std. Error t value Pr(>|t|)  b0 6.207e+00 6.212e-01 9.991 < 2e-16 \*\*\*  b1 1.089e-03 2.654e-04 4.103 5.21e-05 \*\*\*  b2 8.257e-06 6.972e-05 0.118 0.906  b3 -2.117e-05 1.074e-04 -0.197 0.844  b4 -1.219e-03 8.215e-04 -1.484 0.139  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 7.391 on 314 degrees of freedom  Number of iterations to convergence: 2  Achieved convergence tolerance: 1.49e-08  > number\_v11\_DMA\_nls  [1] 34  > score\_v11\_DMA\_nls[number\_v11\_DMA\_nls]  [[1]]  [1] 4.142687  > scoretrain\_v11\_DMA\_nls[number\_v11\_DMA\_nls]  [[1]]  [1] 7.332932  > bestscore\_v11\_DMA\_nls  [[1]]  [1] 4.142687 |
|  |
| |  | | --- | | > | |

# Model Comparisons

