Project - Forecasting KPIs using Linear Regression

##Objective : Predict Lead Volumes more accurately in order to meet client expectations. Dataset :

Part-1 : Loading Necessary Libraries

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

Part-2: Reading Data and Data Manipulation

```
Client = read.csv('TrainWelspun.csv')
str(Client)
```

```
## 'data.frame':    1215 obs. of  8 variables:
##  $ Day        : chr  "01-08-2020" "01-08-2020" "01-08-2020" "01-08-2020" ...
##  $ Budget     : int  1200 750 1000 350 720 640 600 530 350 720 ...
##  $ Cost       : num  522.5 0 89.5 0 0 ...
##  $ Impressions: int  219 0 79 0 0 0 0 0 0 0 ...
##  $ Clicks     : int  26 0 29 0 0 0 0 0 0 0 ...
##  $ CTR        : num  0.119 0 0.367 0 0 0 0 0 0 0 ...
##  $ Avg..CPC   : num  20.1 0 3.08 0 0 0 0 0 0 0 ...
##  $ Conversions: num  1 0 1 0 0 0 0 0 0 0 ...
```

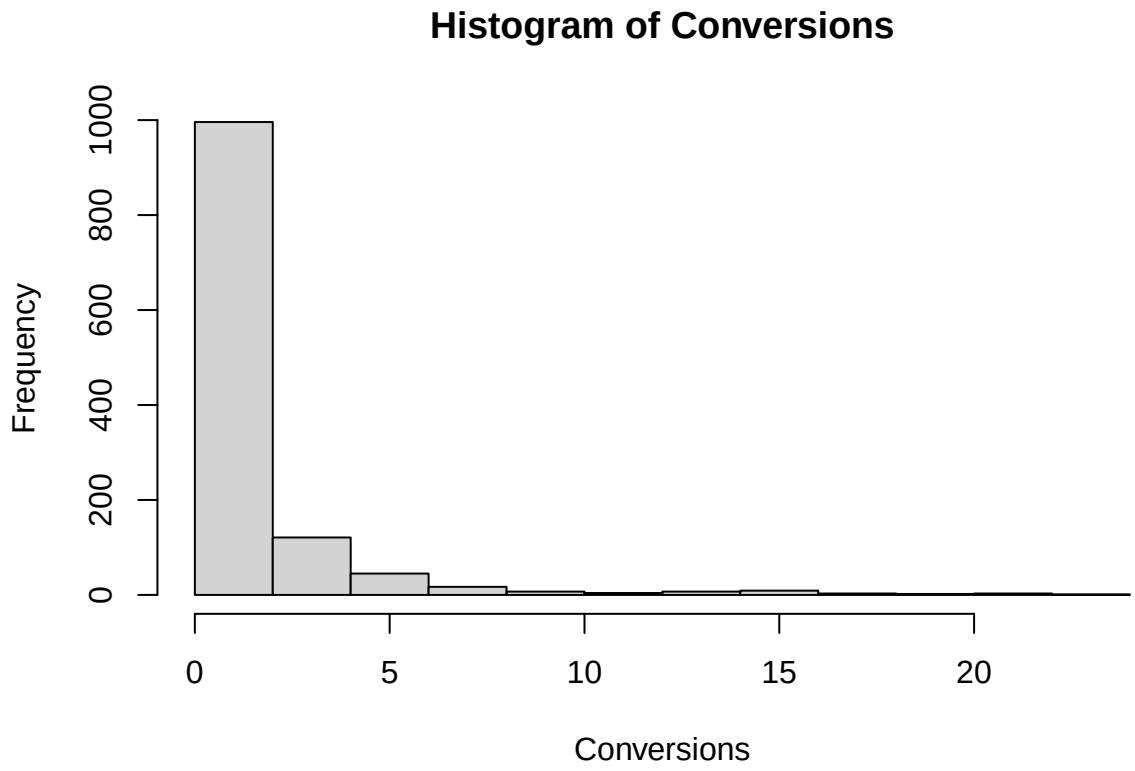
```r
#Missing values
colSums(is.na(Client))
```

```
##         Day      Budget        Cost Impressions      Clicks         CTR
##           0           0           0           0           0           0
##    Avg..CPC Conversions
##           0           0
```

```r
colSums(Client=="")
```

```
##         Day      Budget        Cost Impressions      Clicks         CTR
##           0           0           0           0           0           0
##    Avg..CPC Conversions
##           0           0
```

```r
#Changing lead values to integer
Client$Conversions = round(Client$Conversions)
hist(Client$Conversions, xlab = "Conversions",
     main = "Histogram of Conversions")
```




```r
#Date Manipulation
Client$Day = as.Date(Client$Day,"%d-%m-%y")
Client$Month = as.factor(format(Client$Day,"%B") )
Client$Month = factor(Client$Month ,
```

```
                      levels = c("August", "September", "October", "November"))

#CTR improves MOM implying optimization work.
with(Client,tapply(CTR, Month,mean))
```

```
##     August  September    October   November
## 0.06857416 0.17701667 0.16349802 0.23011066
```

```
#Lead Volumes Increase.
with(Client,tapply(Conversions,Month,sum))
```

```
##    August September   October  November
##       261       408       342       720
```

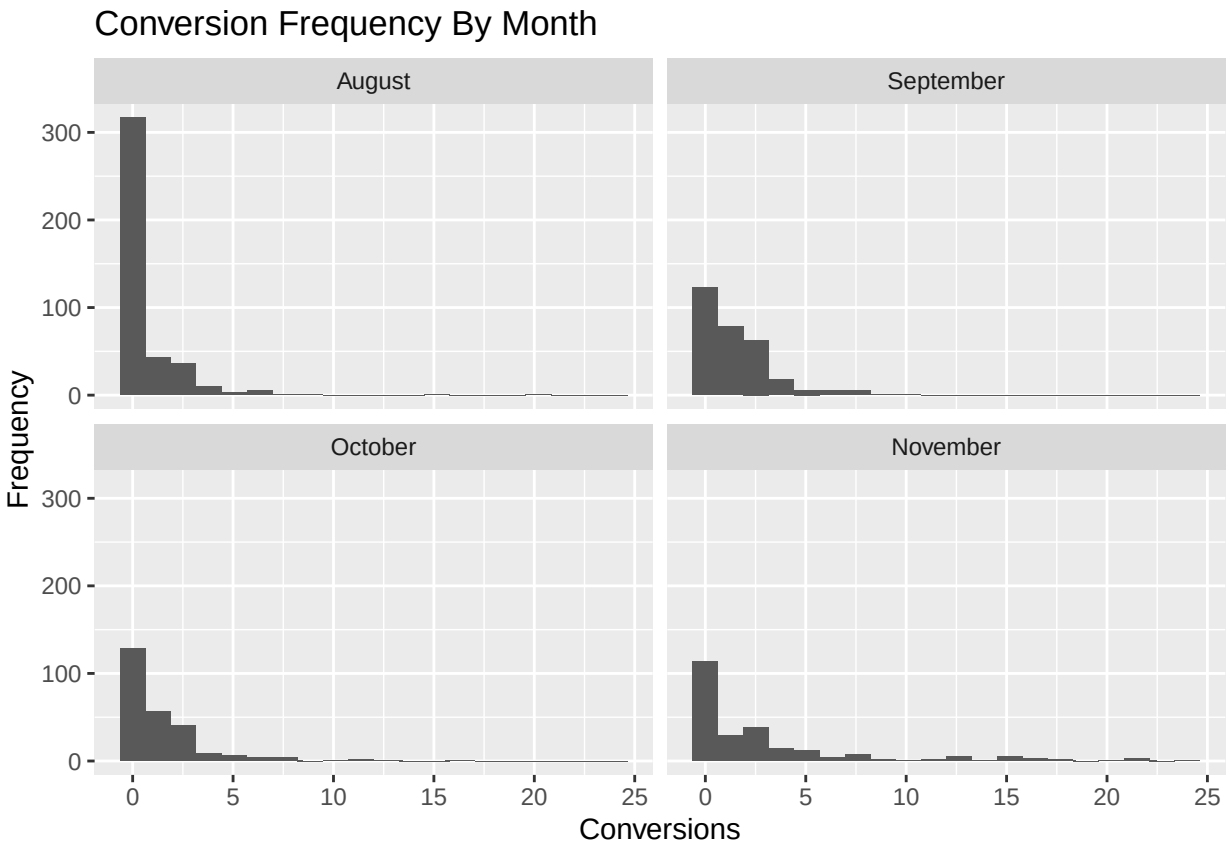
```
#CPL Trend
with(Client,tapply(Cost,Month,sum))/ with(Client,tapply(Conversions,Month,sum))
```

```
##    August September   October  November
##  700.4207  918.2482  815.5386  668.7543
```

```
#Classification Bins
Client$CostFactor = cut(Client$Cost,5,include.lowest = T,
                        labels = c("Lowest","Low","Medium","High","V High"))
Client$ImprFactor = cut(Client$Impressions,5,include.lowest = T,
                        labels = c("Lowest","Low","Medium","High","V High"))
Client$ClickFactor = cut(Client$Clicks,5,include.lowest = T,
                         labels = c("Lowest","Low","Medium","High","V High"))
Client$ConvFactor = cut(Client$Cost,3,include.lowest = T,
                        labels = c("low","Medium","High"))

#Checking how many features we can move to factors
apply(Client,2, function(x) length(unique(x)))
```

```
##         Day      Budget        Cost Impressions      Clicks         CTR
##         122          25         852         573         267         303
##    Avg..CPC Conversions       Month  CostFactor  ImprFactor ClickFactor
##         689          23           4           5           5           5
##  ConvFactor
##           3
```

```
#Understanding Distribution of Conversions
ggplot(data = Client,aes(x=(Conversions)))+
geom_bar(position="fill")+
ylab("Frequency")+
facet_wrap(~Month, ncol=2)+
labs(x = "Conversions", title = 'Conversion Frequency By Month')+
geom_histogram(bins=20)
```

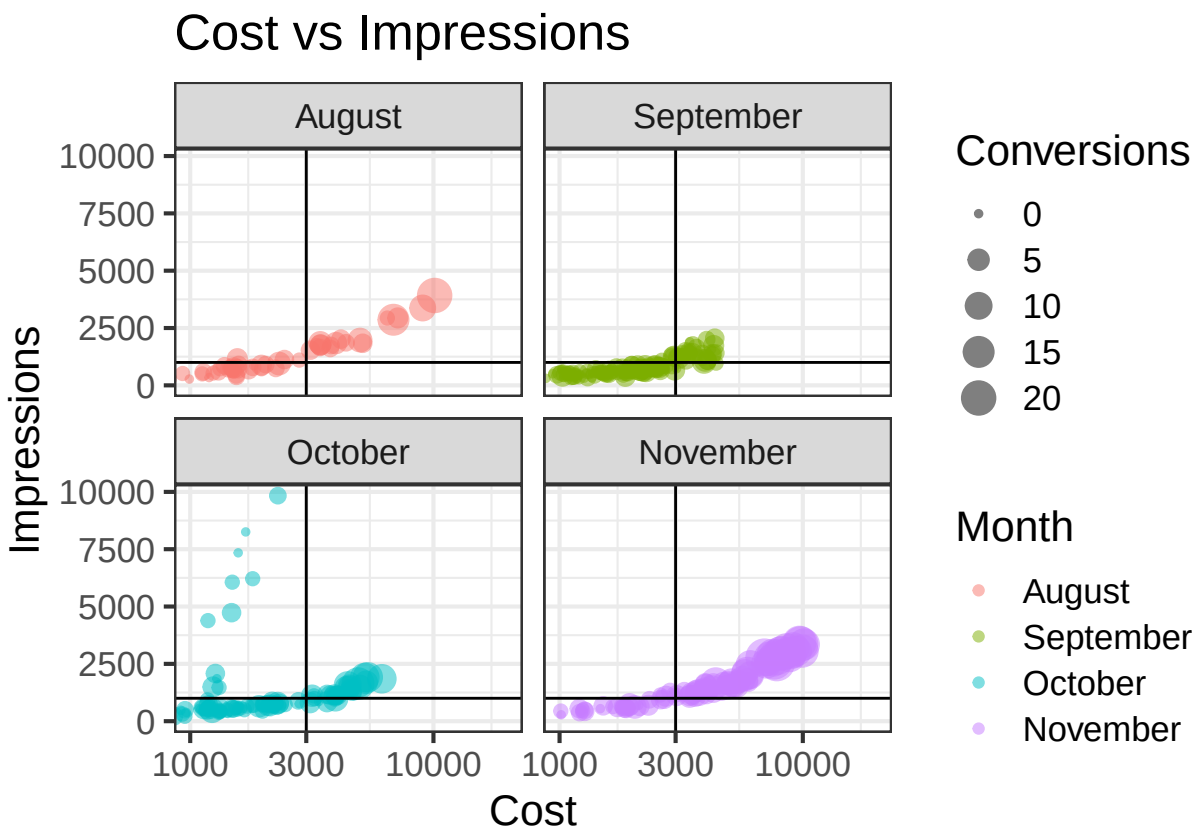


1.Data is skewed towards the left. Points having zero leads are very high. Let us check this data

2. About 361 have 0 spends and thus have no influence on our model.Removing zero spend data points.

Part-3 : Visualizing Data

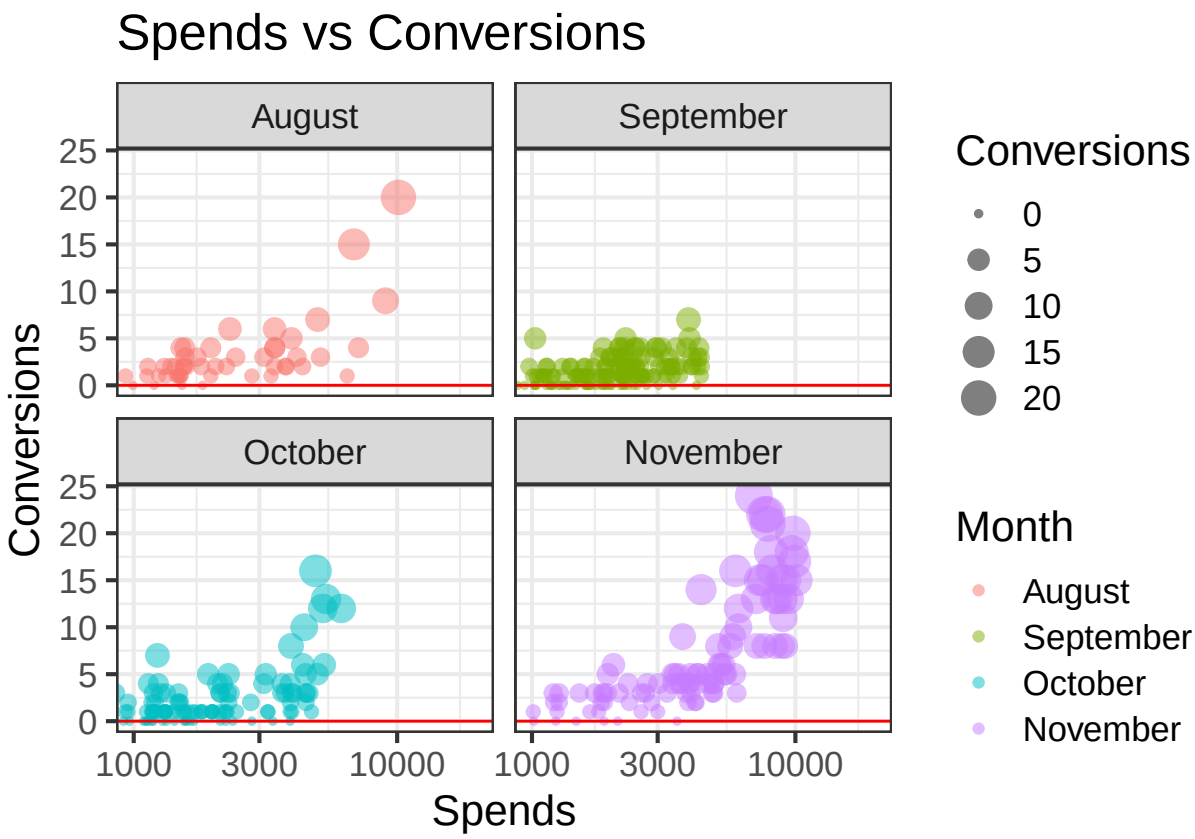
```
#Understanding relation with Impressions
 Client %>% ggplot(aes(x =(Cost) , y = (Impressions),
                       color = Month, size = Conversions))+
 geom_point(alpha=.5)+
 labs(x = "Cost", y="Impressions", title = 'Cost vs Impressions')+
 scale_x_log10()+
 coord_cartesian(xlim = c(1000,20000))+
 theme_bw(base_size = 16)+
 facet_wrap(~Month, ncol=2)+
 geom_hline(yintercept = 1000)+geom_vline(xintercept = 3000)
```




```
Outlier = nrow(filter(Client,Cost >= 3000 & Impressions <1000))
signific = Outlier/nrow(Client)*100
```

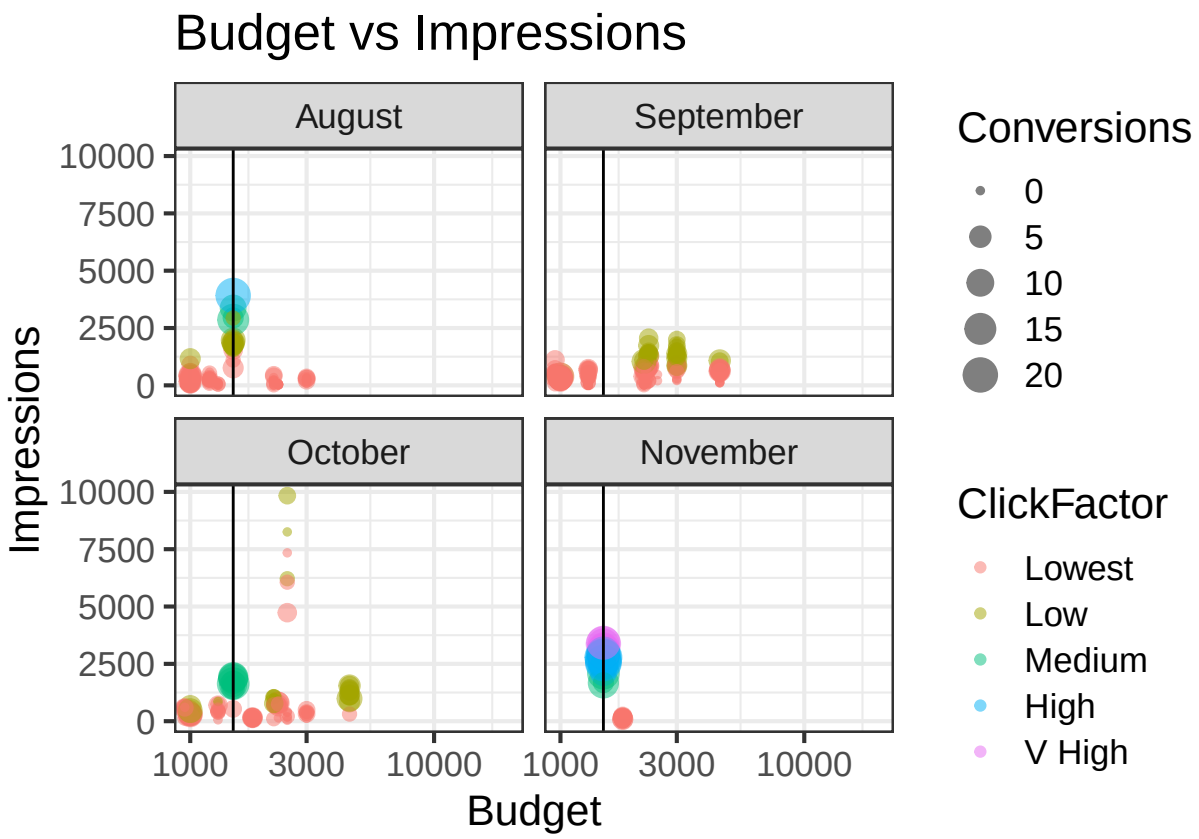
1. By setting x and y intercepts, we can verify that we would always get atleast 1000 impressions for every 3000 we spend. **To verify:**
2. Only 7 rows out of 854 break this rule which is about 0.82% hence insignificant.

```
Client %>% ggplot(aes(x =Cost , y = Conversions,
                      color = Month, size = Conversions))+
geom_point(alpha=.5)+
labs(x = "Spends", y="Conversions", title = 'Spends vs Conversions')+
scale_x_log10()+
coord_cartesian(xlim = c(1000,20000))+
theme_bw(base_size = 16)+
facet_wrap(~Month, ncol=2)+
geom_hline(yintercept = 0, col = 'red')
```



Cost and Conversions have a strong correlation

```
#Understanding the relation with Budget
  Client %>% ggplot(aes(x =(Budget) , y = (Impressions),
                        color = ClickFactor, size = Conversions))+
  geom_point(alpha=.5)+
  labs(x = "Budget", y="Impressions", title = 'Budget vs Impressions')+
  scale_x_log10()+
  coord_cartesian(xlim = c(1000,20000))+
  theme_bw(base_size = 16)+
  facet_wrap(~Month, ncol=2)+
  geom_vline(xintercept = 1500)
```



By setting the x intercept we can say that a budget of **1500** is the most optimum. We have *higher lead counts as indicated by conversions* and *blue-violet colors indicating higher clicks.*

To verify:

1. We got a CPL 34.82% higher when we had a budget $>$ or $<$ 1500.
2. Our lead count for budget of 1500 was 622.15% higher on average.

Part-4 : Creating Linear Regression Model

```
#Removing Day Column since we wont be doing time series analysis and using this to find correlation mat
subset = select(Client,Budget,Cost,Impressions,Clicks,CTR,Avg..CPC,Conversions)
cor(subset)
```

```
##                  Budget       Cost Impressions      Clicks         CTR
## Budget       1.00000000  0.2130029  0.18378502  0.13306134 -0.1848569
## Cost         0.21300293  1.0000000  0.74060515  0.95219746 -0.2126674
## Impressions  0.18378502  0.7406052  1.00000000  0.73512908 -0.2489259
## Clicks       0.13306134  0.9521975  0.73512908  1.00000000 -0.1085695
## CTR         -0.18485688 -0.2126674 -0.24892588 -0.10856955  1.0000000
## Avg..CPC     0.26769704  0.2204017  0.09719505  0.01844036 -0.6257049
## Conversions  0.04139174  0.7364331  0.54555007  0.83045712  0.0140580
##                Avg..CPC Conversions
## Budget       0.26769704  0.04139174
```

```
## Cost         0.22040168  0.73643309
## Impressions  0.09719505  0.54555007
## Clicks       0.01844036  0.83045712
## CTR         -0.62570487  0.01405800
## Avg..CPC     1.00000000 -0.13816572
## Conversions -0.13816572  1.00000000
```

```r
#Splitting data into train and test sets
split = sample.split(Client$Conversions, SplitRatio = 0.7)
train = subset(Client, split == TRUE)
test = subset(Client, split == FALSE)

#Model
model1 = lm((Conversions) ~ as.factor(Budget) + Clicks + Cost + CTR + Avg..CPC, data = train)
summary(model1)
```

```
##
## Call:
## lm(formula = (Conversions) ~ as.factor(Budget) + Clicks + Cost +
##     CTR + Avg..CPC, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0245 -0.7446 -0.0581  0.6727 11.0917
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             0.2225469  0.6110982   0.364    0.716
## as.factor(Budget)300    0.5316111  0.5356663   0.992    0.321
## as.factor(Budget)400    0.2534959  1.0476662   0.242    0.809
## as.factor(Budget)500    0.2476287  0.4952516   0.500    0.617
## as.factor(Budget)600    0.1153593  0.5286400   0.218    0.827
## as.factor(Budget)700   -0.2682136  0.5255354  -0.510    0.610
## as.factor(Budget)800   -0.5387847  0.7205241  -0.748    0.455
## as.factor(Budget)950    0.4702380  0.5328245   0.883    0.378
## as.factor(Budget)1000   0.8839107  0.5641010   1.567    0.118
## as.factor(Budget)1200   0.5521048  0.6058260   0.911    0.363
## as.factor(Budget)1300   0.2180125  0.5572961   0.391    0.696
## as.factor(Budget)1500   2.8630134  0.6040541   4.740 2.70e-06 ***
## as.factor(Budget)1800   0.5437083  0.5859063   0.928    0.354
## as.factor(Budget)2200   0.0690812  0.5404706   0.128    0.898
## as.factor(Budget)2300   0.2231914  0.5362973   0.416    0.677
## as.factor(Budget)2500  -0.8422062  0.6581311  -1.280    0.201
## as.factor(Budget)3000  -0.3753855  0.5264767  -0.713    0.476
## as.factor(Budget)4500   0.8831238  0.5717895   1.544    0.123
## Clicks                  0.0289850  0.0025512  11.361  < 2e-16 ***
## Cost                   -0.0007153  0.0001681  -4.255 2.44e-05 ***
## CTR                     0.7830125  0.6631512   1.181    0.238
## Avg..CPC               -0.0321246  0.0206126  -1.558    0.120
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.631 on 577 degrees of freedom
## Multiple R-squared:  0.7558, Adjusted R-squared:  0.7469
```

```
## F-statistic: 85.05 on 21 and 577 DF,  p-value: < 2.2e-16
```

1. The R Squared for this model is pretty close to 1 indicating a good model.
2. It can be improved by adding additional variables like - Ad quality, Location, Placement data , Campaign Name and Ad group info.

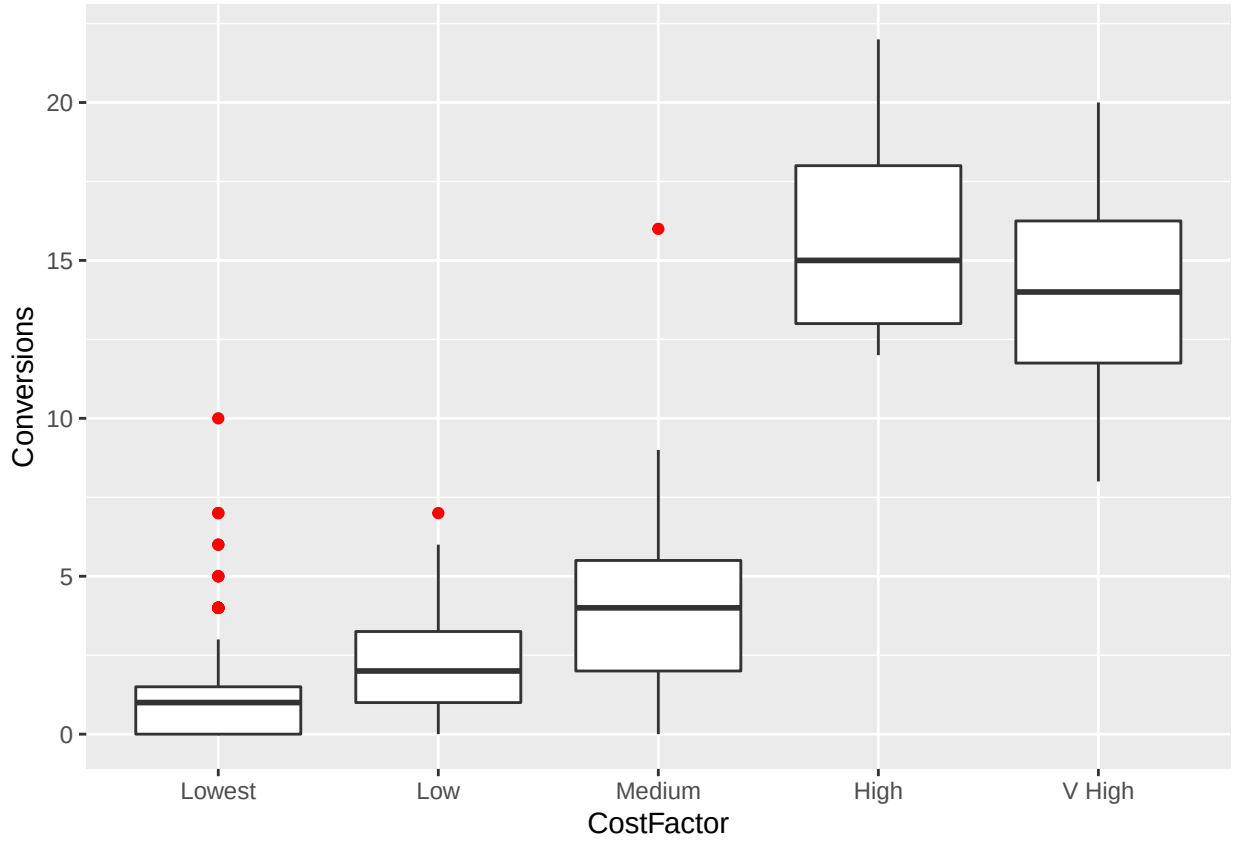
The SSE for this model is 1534.33 and RMSE is 1.6 *implying our model is off by this value on average.*

Part-4 : Testing Predictions

```
#Using data for predictions and confusion matrix on Testing Set
predTest = round(predict(model1, newdata = test))


#Leads cannot be less than 0 so changing those values
predTest [predTest  < 0] = 0
test$LeadPrediction =  predTest


#Model Accuracy on Training Set
confusionMatrixTest = table(test$Conversions,predTest)

#Variables defined
sumCorrectPred  = 0
rowCount = ncol(confusionMatrixTest)-1
colCount = nrow(confusionMatrixTest)-1

#Diagonal Sum
for(i in 1:(rowCount - 1)){
  for (j in 1:(colCount-1))
    if (i == j){
      sumCorrectPred = confusionMatrixTest[i,j]+sumCorrectPred;
    }
}

#Accuracy of Model
AccuracyTest<-sumCorrectPred/sum(test$Conversions)
```

The accuracy of the model for test set is 21.58%. This is pretty bad. Let us check the outliers in our dataset.

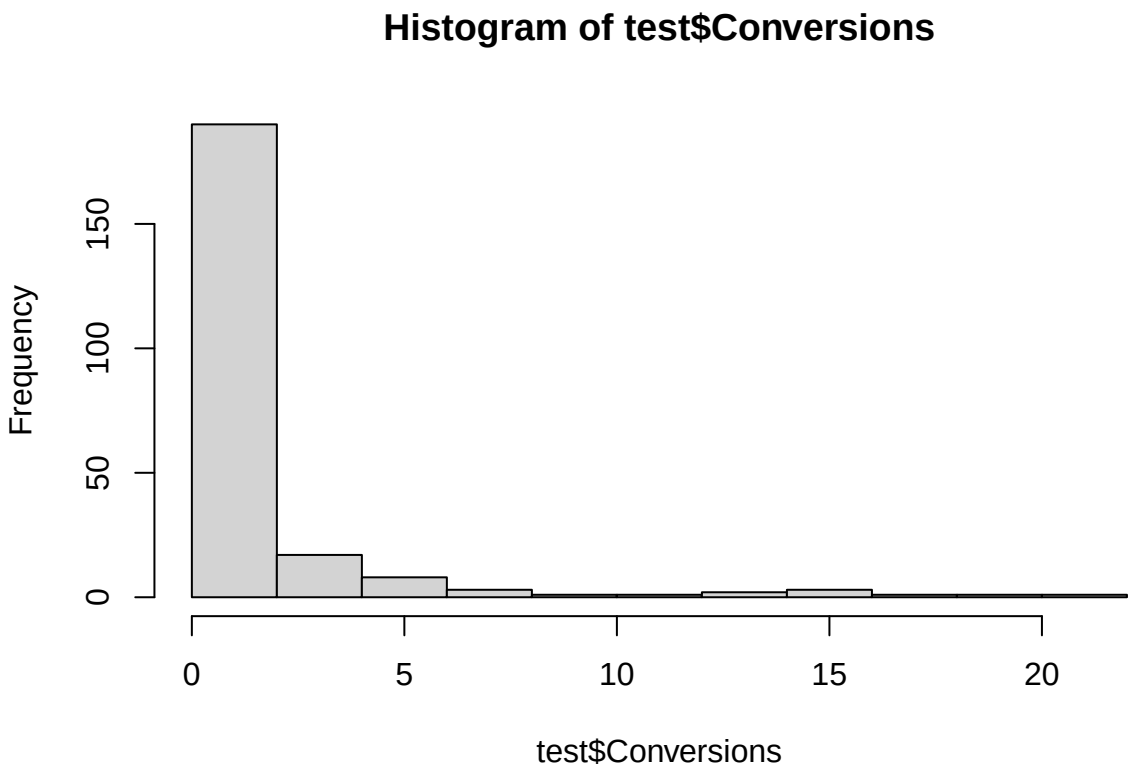
```
test %>% ggplot(aes(x= CostFactor , y = Conversions))+geom_boxplot(outlier.colour = 'red')
```



We can see that most of the outliers belong to the lowest cost category. This can mean Adwords incorrectly counted these as leads when they were based on some other conversion action.

Creating a new Dataset and retesting our predictions.

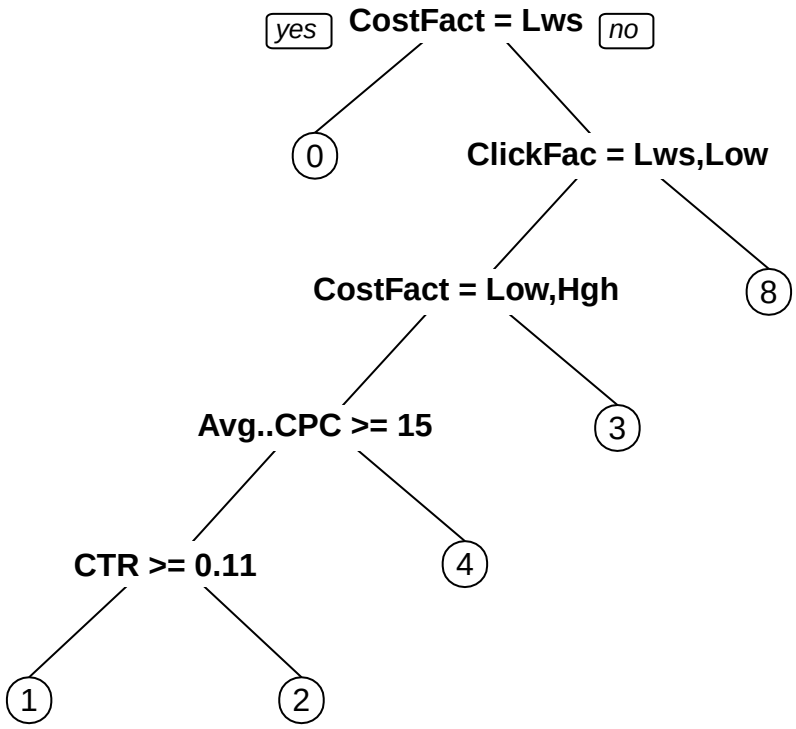
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.5732  1.0000  2.0000
```



The accuracy of the model for test set is 27.61% after removing outliers from the lowest cost category

Part-5 : Creating a tree model

```
modTree = rpart(Conversions ~  CostFactor + ClickFactor + CTR + Avg..CPC , data = train , method = 'cla
prp(modTree)
```

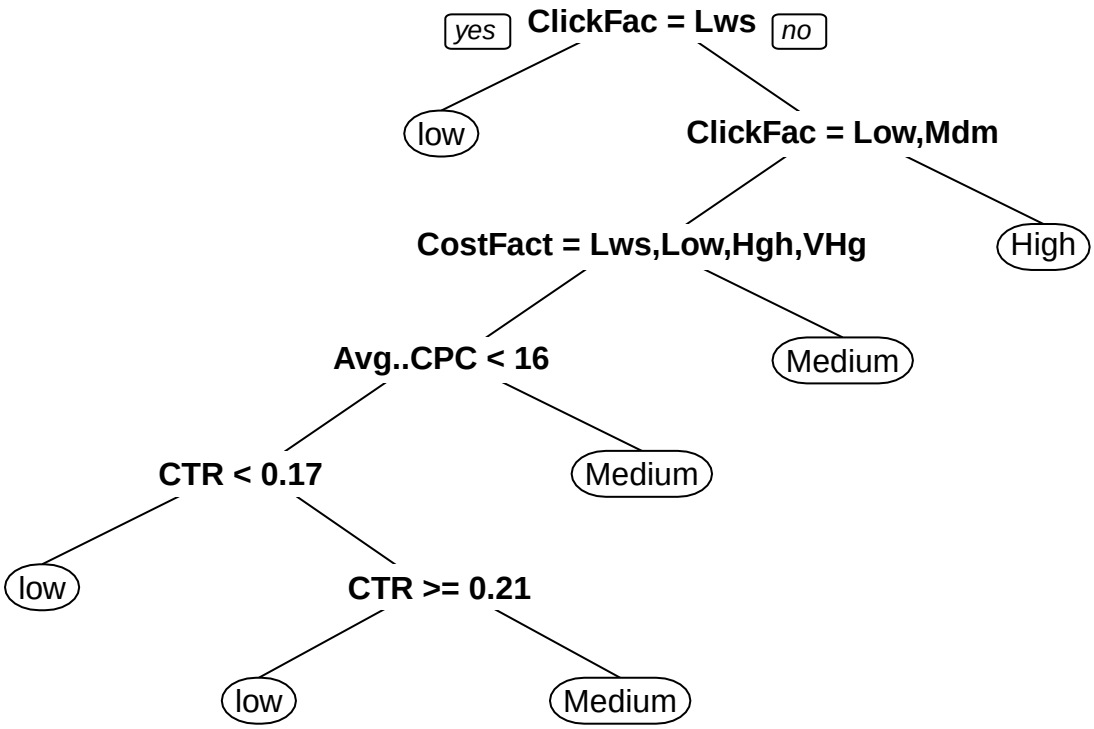



```
predictCART = predict(modTree, newdata = test, type = 'class')
CM2 <-table(test$Conversions, predictCART)

#Variables defined
sumCorrectPred2  = 0
rowCount2 = ncol(CM2)-1
colCount2 = nrow(CM2)-1

#Diagonal Sum
for(i in 1:(rowCount2)){
  for (j in 1:colCount2)
    if (i == j){
      sumCorrectPred2 = CM2[i,j]+sumCorrectPred2;
    }
}
#Accuracy of Model
AccuracyTest2<-sumCorrectPred2/nrow(test)
AccuracyTest2
```

```
## [1] 0.4517544
```

The accuracy of the model for test set is 45.18%. The Tree model performs twice as good as the linear regression model.

```
modTree2 = rpart(ConvFactor ~ CostFactor + ClickFactor + CTR + Avg..CPC , data = train , method = 'class
prp(modTree2)
```




```
predictCART = predict(modTree2, newdata = test, type = 'class')
CM3 <-table(test$ConvFactor, predictCART)
CM3
```

```
##          predictCART
##           low Medium High
##   low     196      2    0
##   Medium    1     19    2
##   High      0      0    8
```

```
#Variables defined
sumCorrectPred3  = 0
rowCount3 = nrow(CM3)-1
colCount3 = ncol(CM3)-1

#Diagonal Sum
for(i in 1:rowCount3){
  for (j in 1:colCount3){
    if (i == j){
      sumCorrectPred3 = CM3[i,j]+sumCorrectPred3;
    }}
```

```
}
#Accuracy of Model
AccuracyTest3<-sumCorrectPred3/nrow(test)
AccuracyTest3
```

```
## [1] 0.9429825
```

This model is built using previously created buckets for Clicks and Cost. The accuracy of the model for test set is 94.3%.