Siddarth Kumar Sampath Kumaar

February 1, 2018

1 INTERNATIONAL PREMIER BANK ANALYIS

In [ ]:
```
name : Siddarth Kumar S
email : ssampath@syr.edu
```

2 INSTALLING AND LOADING LIBRARIES

In [60]:
```
install.packages('randomForest', repos='http://cran.us.r-project.org')
install.packages('readr', repos='http://cran.us.r-project.org')
install.packages('e1071', repos='http://cran.us.r-project.org')
install.packages('caret', repos='http://cran.us.r-project.org')
install.packages('ggplot2', repos='http://cran.us.r-project.org')
install.packages('rpart', repos='http://cran.us.r-project.org')
install.packages('rpart.plot', repos='http://cran.us.r-project.org')
install.packages('corrplot', repos='http://cran.us.r-project.org')
install.packages('pROC',repos='http://cran.us.r-project.org')
install.packages("survival", repos='http://cran.us.r-project.org')
install.packages('gbm', repos='http://cran.us.r-project.org')
install.packages("caret", repos='http://cran.us.r-project.org',dep=TRUE)
#install.packages("devtools", repos='http://cran.us.r-project.org',dep=TRUE)
```

```
Installing package into 'C:/Users/prash/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
Warning message:
"package 'randomForest' is in use and will not be installed"Installing package into 'C:/Users/p
(as 'lib' is unspecified)
Warning message:
"package 'readr' is in use and will not be installed"Installing package into 'C:/Users/prash/Do
(as 'lib' is unspecified)


package 'e1071' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\prash\AppData\Local\Temp\RtmpYxHUXB\downloaded_packages
```

```
Installing package into 'C:/Users/prash/Documents/R/win-library/3.4'
(as 'lib' is unspecified)
Warning message:
"package 'caret' is in use and will not be installed"Installing package into 'C:/Users/prash/Do
(as 'lib' is unspecified)
Warning message:
"package 'ggplot2' is in use and will not be installed"Installing package into 'C:/Users/prash/
(as 'lib' is unspecified)
Warning message:
"package 'rpart' is in use and will not be installed"Installing package into 'C:/Users/prash/Do
(as 'lib' is unspecified)
Warning message:
"package 'rpart.plot' is in use and will not be installed"Installing package into 'C:/Users/pra
(as 'lib' is unspecified)
Warning message:
"package 'corrplot' is in use and will not be installed"Installing package into 'C:/Users/prash
(as 'lib' is unspecified)
Warning message:
"package 'pROC' is in use and will not be installed"Installing package into 'C:/Users/prash/Doc
(as 'lib' is unspecified)
Warning message:
"package 'survival' is in use and will not be installed"Installing package into 'C:/Users/prash
(as 'lib' is unspecified)
Warning message:
"package 'gbm' is in use and will not be installed"Installing package into 'C:/Users/prash/Docu
(as 'lib' is unspecified)
also installing the dependencies 'earth', 'party', 'testthat'



  There are binary versions available but the source versions are later:
         binary source needs_compilation
earth     4.5.1  4.6.0              TRUE
party     1.2-3  1.2-4              TRUE
testthat  1.0.2  2.0.0              TRUE



Warning message:
"package 'caret' is in use and will not be installed"installing the source packages 'earth', '

Warning message:
"running command '"C:/Users/prash/Anaconda2/R/bin/x64/R" CMD INSTALL -l "C:\Users\prash\Documen
"installation of package 'earth' had non-zero exit status"Warning message:
"running command '"C:/Users/prash/Anaconda2/R/bin/x64/R" CMD INSTALL -l "C:\Users\prash\Documen
"installation of package 'party' had non-zero exit status"Warning message:
"running command '"C:/Users/prash/Anaconda2/R/bin/x64/R" CMD INSTALL -l "C:\Users\prash\Documen
"installation of package 'testthat' had non-zero exit status"
```

```
In [61]: library(randomForest)
         library(readr)
         library(reshape)
         library(ggplot2)
         library(rpart)
         library(rpart.plot)
         library(corrplot)
         library(pROC)
         library(survival)
         library(gbm)
         library(caret)
         #library(devtools)
```

3 LOADING DATA

```
In [1]: data <- read_delim("~/R working directory/Camino/Archive/bank-additional/bank-additiona
                                              ";", escape_double = FALSE, trim_ws = TRUE)

        data1 = data
```

```
Parsed with column specification:
cols(
  .default = col_character(),
  age = col_integer(),
  duration = col_integer(),
  campaign = col_integer(),
  pdays = col_integer(),
  previous = col_integer(),
  emp.var.rate = col_double(),
  cons.price.idx = col_double(),
  cons.conf.idx = col_double(),
  euribor3m = col_double(),
  nr.employed = col_integer()
)
See spec(...) for full column specifications.
Warning message in rbind(names(probs), probs_f):
number of columns of result is not a multiple of vector length (arg 1)Warning message:
33425 parsing failures.
row # A tibble: 5 x 5 col     row         col               expected actual expected   <int>
... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ...
See problems(...) for more details.
```

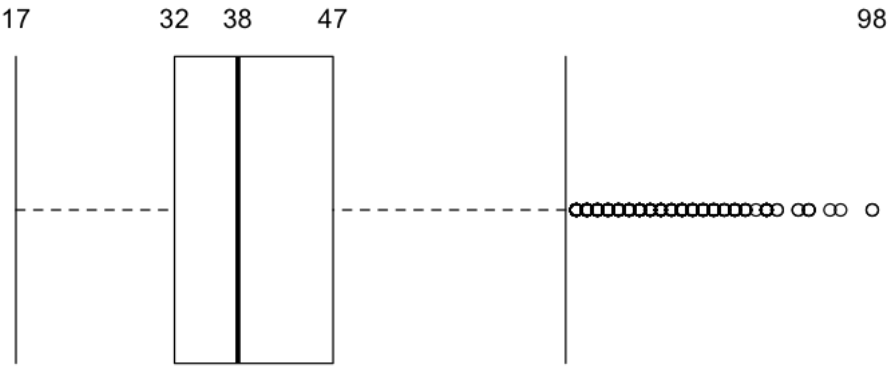
4 Binnning Age Group

```
In [2]: summary(data1$age)
        boxplot(data1$age, horizontal = TRUE, axes = FALSE, staplewex = 1)
```

```
text(x=fivenum(data1$age), labels =fivenum(data1$age), y=1.25)

data1$Age_Binned = ifelse(data1$age>=17 & data1$age<=32,'17-32',
                          ifelse(data1$age>=33 & data1$age<=38,'33-38',
                                 ifelse(data1$age>=39 & data1$age<=47,'39-47','48 & Ab
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  17.00   32.00   38.00   40.02   47.00   98.00
```



WE ARE CATEGORIZING THE AGE GROUP BASED ON THE QUARTILE INFORMATION OBTAINED FROM THE SUMMARY OF THE DATA. THIS WILL HELP US ANALYZE WHICH AGE GROUP IS THE HIGHEST BUYER OF CERTIFICATE OF DEPOSIT THIS WILL ALSO HELP IMPROVE THE ACCURACY OF OUR MODELS BY INTRODUCING AGE AS A FACTOR VARIABLE

In [3]:
```
data1 = data1[,-c(1,11,20)]
data1 = data.frame(data1)
data1 <- as.data.frame(unclass(data1))
```

5 DESCRIPTIVE STATISTICS

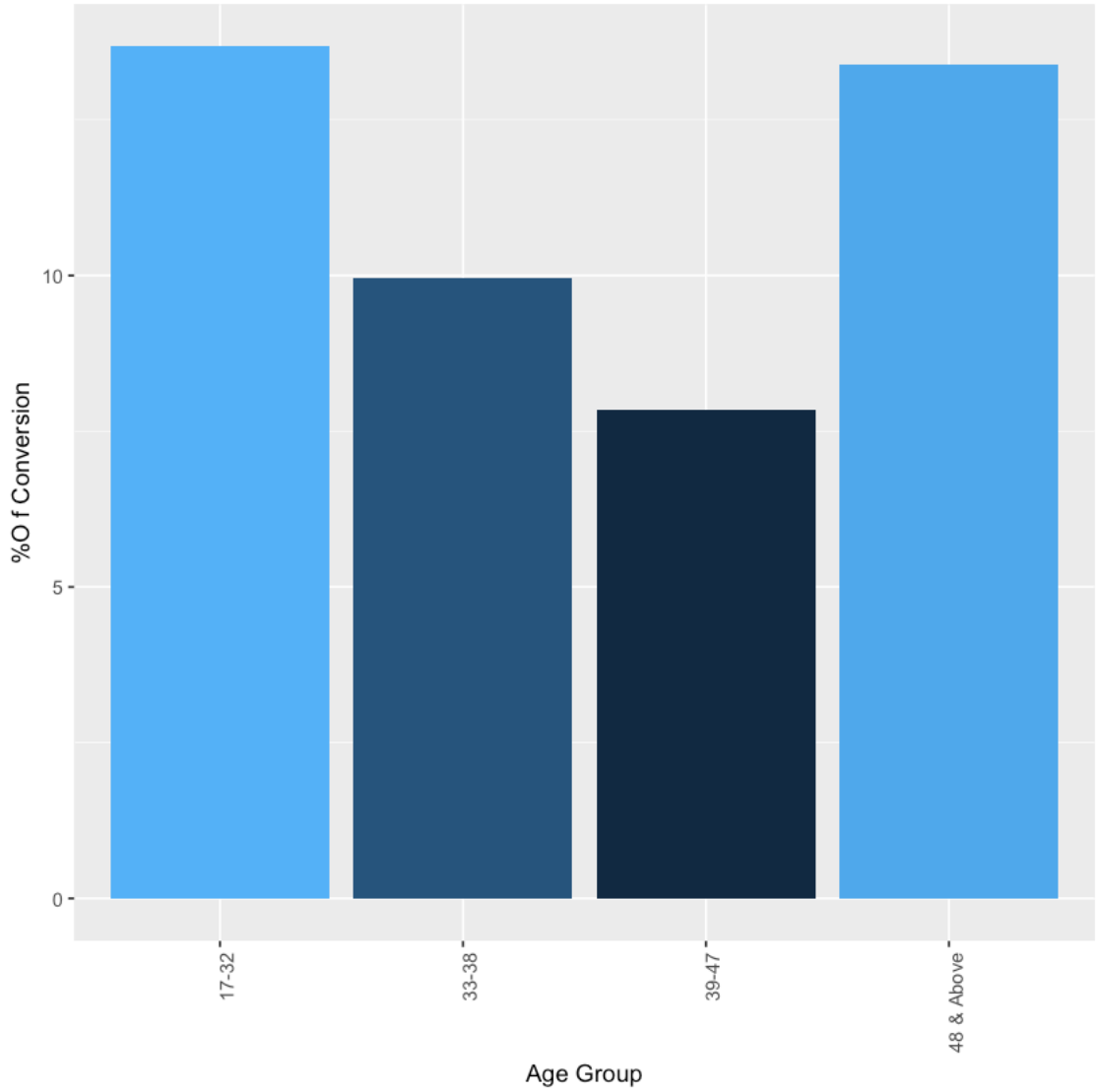
6 WHICH AGE GROUP BUYS MORE CERTIFICATE OF DEPOSIT ?

In [4]:
```
####### Age group ################
table(data1$Age_Binned,data1$y) #### age ggplot
t = data.frame(table(data1$Age_Binned,data1$y))
t1 = cast(t,Var1~Var2,mean)
t1$per = (t1$yes/(t1$yes+t1$no))*100

h <- ggplot(t1,aes(x= t1$Var1,y=t1$per)) + geom_col(aes(fill=t1$per))
h <- h + guides(fill=FALSE)
h <- h+theme (axis.text.x = element_text(angle = 90,hjust = 1))
h <- h+labs(x="Age Group",y="%0 f Conversion")
h
```

```
              no  yes
  17-32      9648 1528
  33-38      9004  995
  39-47      9344  796
  48 & Above 8552 1321
```

```
Using Freq as value column.  Use the value argument to cast to override this choice
```



FROM THE ABOVE VISUALIZATION, WE CAN INFER THAT THE PEOPLE BELONGING TO AGE GROUPS OF 17-32 ARE THE HIGHEST CERTIFICATE OF DEPOSIT BUYERS, FOLLOWED BY PEOPLE BELONGING TO AGE GROUP 48 & ABOVE.

7 IMPACT OF MARITAL STATUS

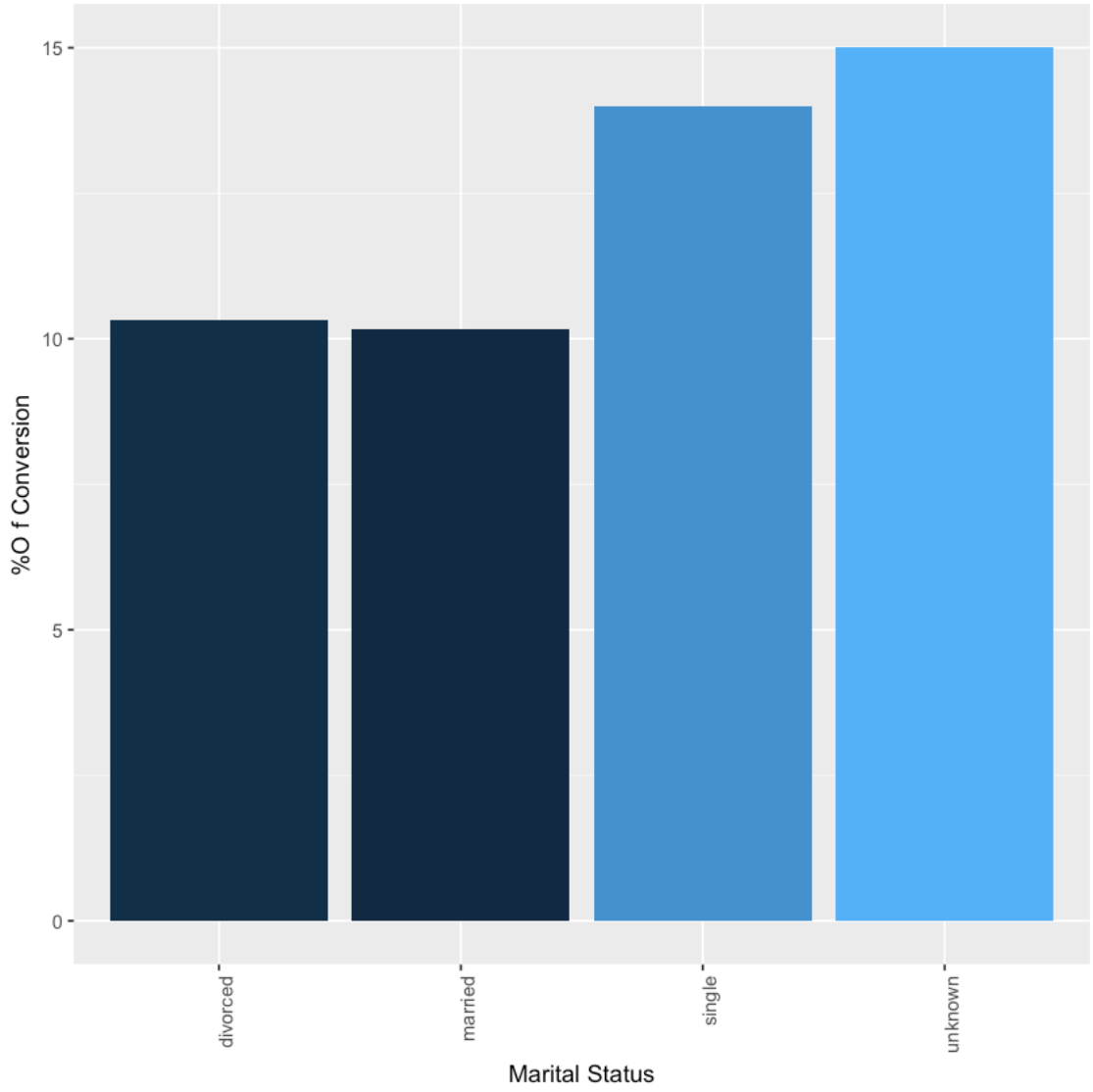
```
In [18]: ############## Marital Status ##############################
         table(data1$marital,data1$y) #### age ggplot
         y = data.frame(table(data1$marital,data1$y))
         y1 = cast(y,Var1~Var2,mean)
         y1$per = (y1$yes/(y1$yes+y1$no))*100

         jj <- ggplot(y1,aes(x= y1$Var1,y=y1$per)) +geom_col(aes(fill=y1$per))
```

```
jj <- jj + guides(fill=FALSE)
jj<- jj+theme (axis.text.x = element_text(angle = 90,hjust = 1))
jj <- jj+labs(x="Marital Status",y="%O f Conversion")
jj
```

```
           no   yes
divorced  4136   476
married  22396  2532
single    9948  1620
unknown     68    12
```

```
Using Freq as value column.  Use the value argument to cast to override this choice
```



FROM THE ABOVE CHART, WE CAN SEE THAT APART FROM UNKNOWN MARITAL STATUS, PEOPLE WHO ARE SINGLE BUY MORE NUMBER OF CERTIFICATE OF DEPOSITS.

8 LITERACY LEVEL OF BUYERS

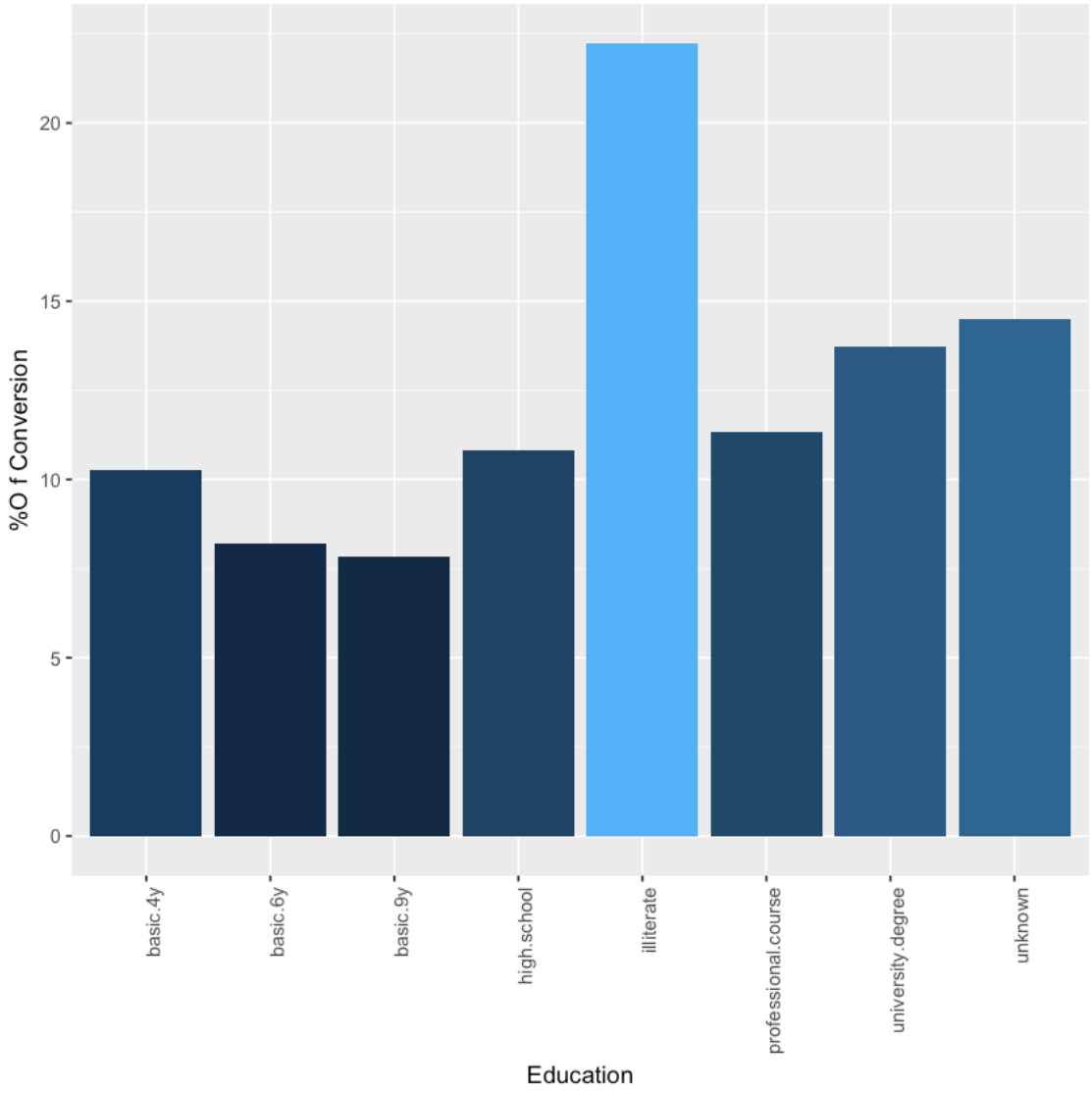
In [12]:
```
###################education####################

        table(data1$education,data1$y) #### age ggplot
        e = data.frame(table(data1$education,data1$y))
        e1 = cast(e,Var1~Var2,mean)
        e1$per = (e1$yes/(e1$yes+e1$no))*100

        jj <- ggplot(e1,aes(x= e1$Var1,y=e1$per)) +geom_col(aes(fill= e1$per))
        jj <- jj + guides(fill=FALSE)
        jj<- jj+theme (axis.text.x = element_text(angle = 90,hjust = 1))
        jj <- jj+labs(x="Education",y="%0 f Conversion")
        jj
```

	no	yes
basic.4y	3748	428
basic.6y	2104	188
basic.9y	5572	473
high.school	8484	1031
illiterate	14	4
professional.course	4648	595
university.degree	10498	1670
unknown	1480	251

```
Using Freq as value column.  Use the value argument to cast to override this choice
```



FROM THE ABOVE ANALYSIS, IT IS CLEAR THAT PEOPLE WHO ARE ILLITERATE ARE BUYING MORE CERTIFICATE OF DEPOSITS. THIS ANALYSIS HIGHLIGHTS THE EASE OF PURCHASING A CERTIFICATE DEPOSIT AND ALSO THE EFFICIENCY OF THE BANK'S MARKETING.

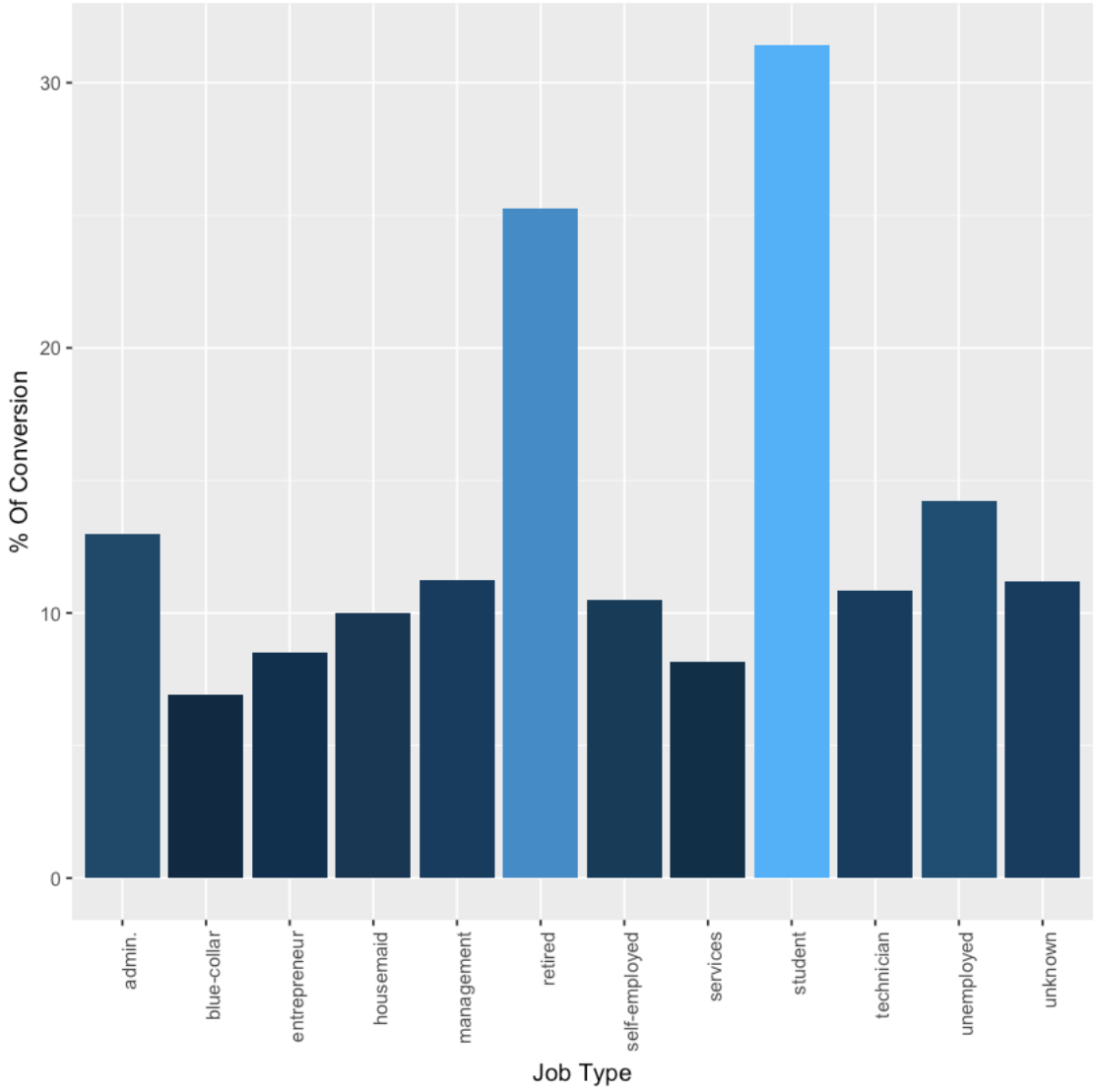
9 Impact of Job Type

In [13]: ```
############ which Job is having higher conversion rate ?##################
```

```
tt = data.frame(table(data1$job,data1$y))
tt1 = cast(tt,Var1~Var2,mean)
tt1$per =(tt1$yes/(tt1$yes+tt1$no))*100
```
```

```
g <- ggplot(tt1,aes(x= tt1$Var1,y=tt1$per)) +geom_col(aes(fill=tt1$per))
g <- g + guides(fill=FALSE)
g <- g+theme (axis.text.x = element_text(angle = 90,hjust = 1))
g <- g+labs(x="Job Type",y="% Of Conversion")
g
```

```
Using Freq as value column.  Use the value argument to cast to override this choice
```



FROM THE ANALYSIS, WE CAN INFER THAT STUDENTS ARE THE HIGHEST BUYERS OF CERTIFICATE OF DEPOSIT, FOLLOWED BY PEOPLE WHO HAVE RETIRED. THIS ANALYSIS BOLSTERS THE AGE GROUP ANALYSIS.

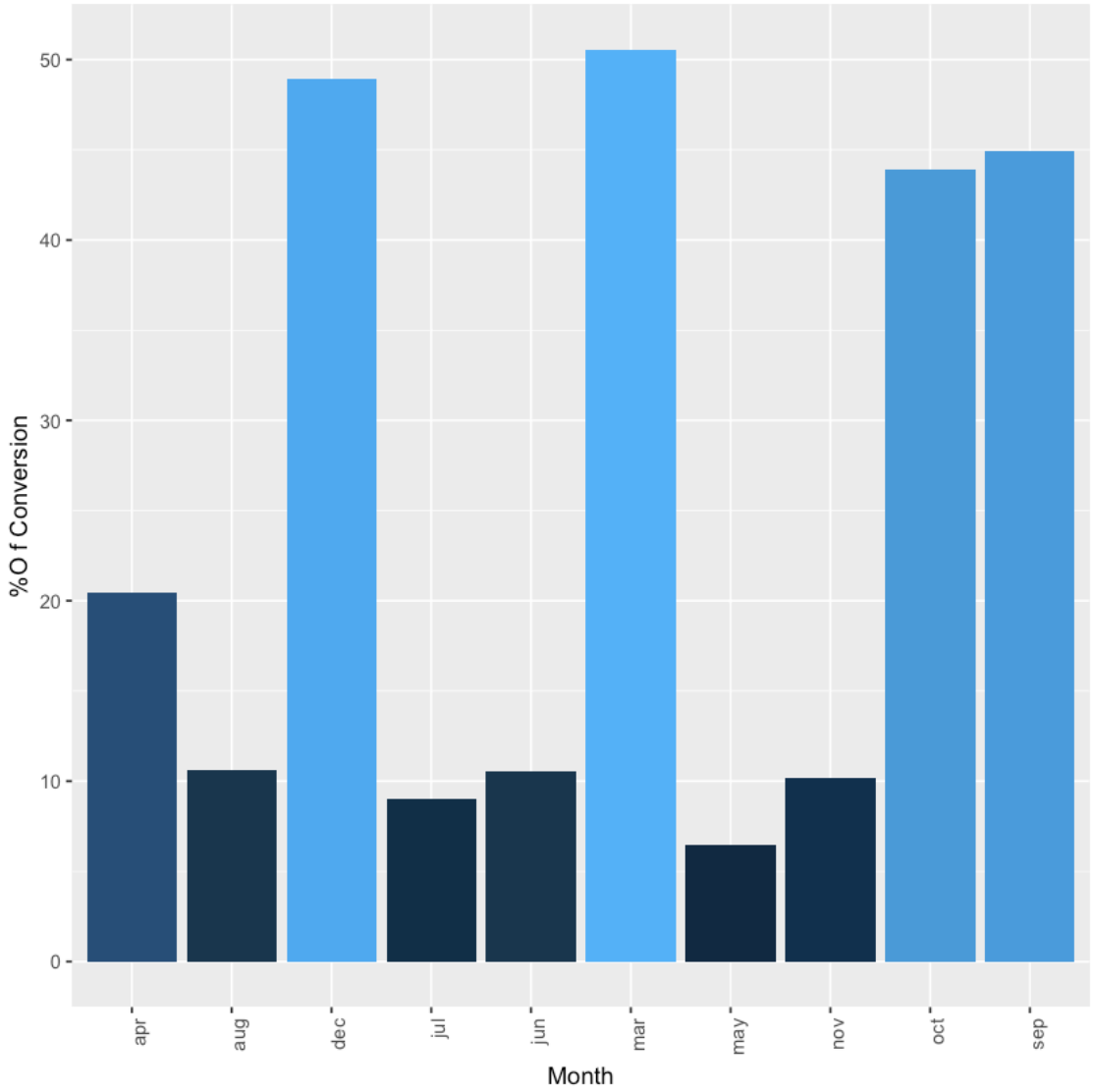
10 Monthly Performance

In [16]:
```
############################## Month Analysis ###############
table(data1$month ,data1$y) #### age ggplot
mn = data.frame(table(data1$month,data1$y))
mn = cast(mn,Var1~Var2,mean)
mn$per = (mn$yes/(mn$yes+mn$no))*100

jt <- ggplot(mn,aes(x= mn$Var1,y=mn$per)) +geom_col(aes(fill=mn$per))
jt <- jt + guides(fill=FALSE)
jt<- jt+theme (axis.text.x = element_text(angle = 90,hjust = 1))
jt <- jt+labs(x="Month",y="%O f Conversion")
jt
```

```
        no   yes
  apr  2093   539
  aug  5523   655
  dec    93    89
  jul  6525   649
  jun  4759   559
  mar   270   276
  may 12883   886
  nov  3685   416
  oct   403   315
  sep   314   256
```

```
Using Freq as value column.  Use the value argument to cast to override this choice
```



BASED ON THE ABOVE MONTHLY ANALYSIS, WE CAN DETERMINE THAT MARCH HAS THE HIGHEST NUMBER OF BUYERS AND CLOSELY FOLLOWED BY DECEMBER, OCTOBER AND SEPTEMBER.

11 INFLUENCE OF CONTACT TYPE

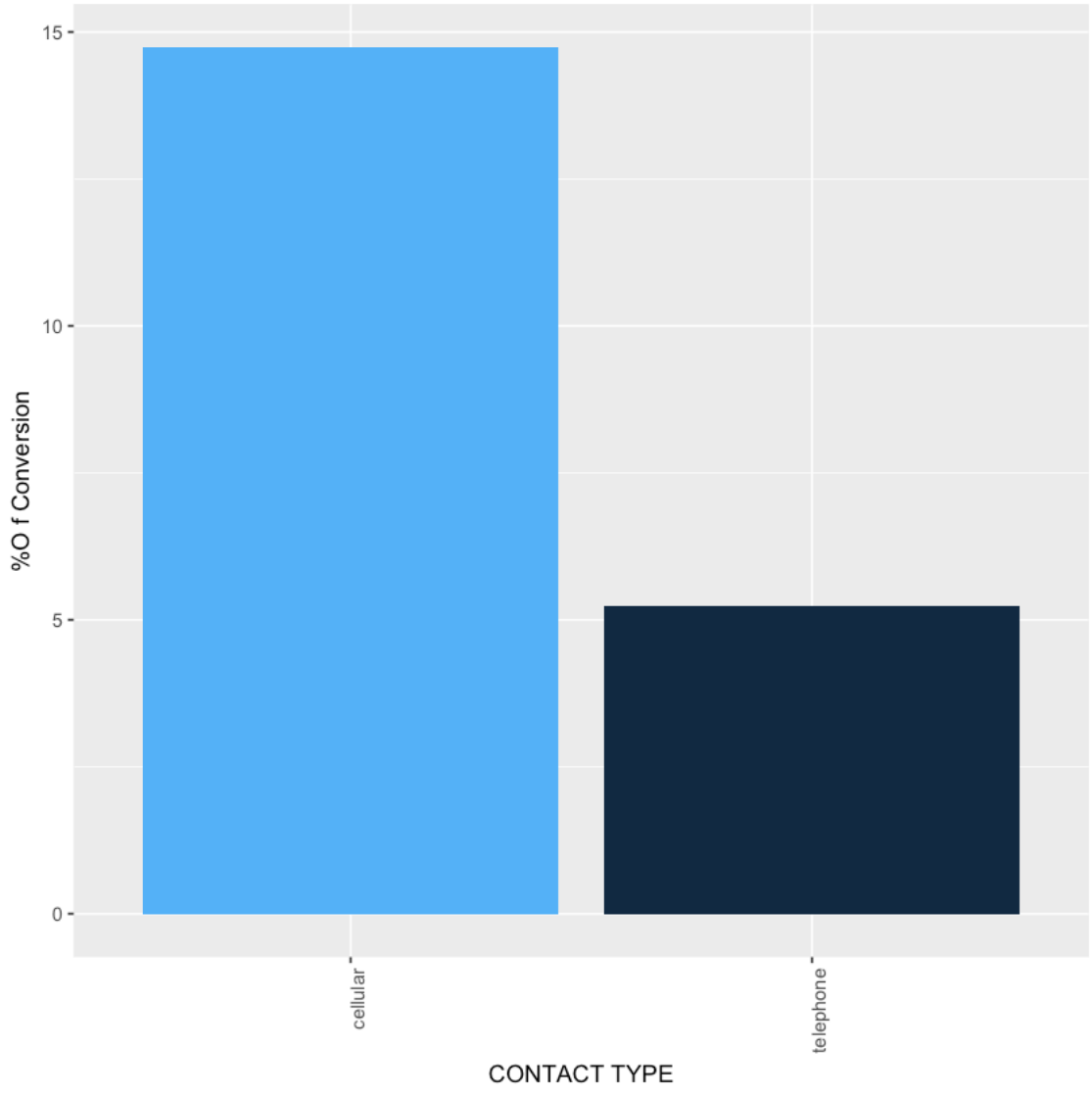
```
In [19]: ########################## Contact ########################################
         table(data1$contact ,data1$y) #### age ggplot
         mt = data.frame(table(data1$contact,data1$y))
         mt = cast(mt,Var1~Var2,mean)
         mt$per = (mt$yes/(mt$yes+mt$no))*100

         jn <- ggplot(mt,aes(x= mt$Var1,y=mt$per)) +geom_col(aes(fill=mt$per))
```

```
jn <- jn + guides(fill=FALSE)
jn<- jn+theme (axis.text.x = element_text(angle = 90,hjust = 1))
jn <- jn+labs(x="CONTACT TYPE",y="%O f Conversion")
jn
```

```
             no   yes
cellular  22291  3853
telephone 14257   787
```

```
Using Freq as value column.  Use the value argument to cast to override this choice
```



FROM THE ABOVE VISUALS, WE CAN CLEARLY INFER THAT CONTACTING CUSTOMERS THROUGH THEIR CELLLULAR PHONES IS AN IDEAL WAY TO CONVERT THEM INTO BUYERS, RATHER THAN CONTACTING THEM THROUGH TELEPHONE

12 BUILDING DECISION TREES

In [72]:
```
##############################Building Decision trees#################################
```

```
fit <- rpart(y~.,data = data1,method = "class")
summary(fit)
```

```
Call:
rpart(formula = y ~ ., data = data1, method = "class")
  n= 41188

          CP nsplit rel error xerror       xstd
1 0.05226293      0 1.0000000    1.0 0.01382889
2 0.01000000      2 0.8954741    0.9 0.01320226

Variable importance
     euribor3m  cons.conf.idx cons.price.idx          pdays   emp.var.rate
            30             18             15             11             11
         month       poutcome       previous
             9              5              1

Node number 1: 41188 observations,    complexity param=0.05226293
  predicted class=no   expected loss=0.1126542  P(node) =1
    class counts: 36548  4640
   probabilities: 0.887 0.113
  left son=2 (36883 obs) right son=3 (4305 obs)
  Primary splits:
      euribor3m     < 1.2395  to the right, improve=1130.2850, (0 missing)
      pdays         < 513     to the right, improve= 869.1165, (0 missing)
      poutcome      splits as  LLR,         improve= 823.6737, (0 missing)
      emp.var.rate  < -0.65   to the right, improve= 698.6193, (0 missing)
      cons.conf.idx < -35.45  to the left,  improve= 550.1827, (0 missing)
  Surrogate splits:
      cons.conf.idx  < -35.45  to the left,  agree=0.959, adj=0.611, (0 split)
      cons.price.idx < 92.7345 to the right, agree=0.949, adj=0.509, (0 split)
      emp.var.rate   < -2.35   to the right, agree=0.934, adj=0.369, (0 split)
      month          splits as  LLRLLLLLRR,  agree=0.927, adj=0.306, (0 split)
      pdays          < 513     to the right, agree=0.916, adj=0.195, (0 split)

Node number 2: 36883 observations
  predicted class=no   expected loss=0.07263509  P(node) =0.8954793
    class counts: 34204  2679
```

```
   probabilities: 0.927 0.073

Node number 3: 4305 observations,    complexity param=0.05226293
  predicted class=no   expected loss=0.4555168  P(node) =0.1045207
    class counts:  2344  1961
   probabilities: 0.544 0.456
  left son=6 (3154 obs) right son=7 (1151 obs)
  Primary splits:
      pdays          < 16.5    to the right, improve=204.58530, (0 missing)
      poutcome       splits as  LLR,         improve=203.28370, (0 missing)
      previous       < 0.5     to the left,  improve= 45.75045, (0 missing)
      contact        splits as  RL,          improve= 40.95878, (0 missing)
      cons.price.idx < 92.559  to the left,  improve= 39.00660, (0 missing)
  Surrogate splits:
      poutcome splits as  LLR,         agree=0.972, adj=0.894, (0 split)
      previous < 1.5     to the left,  agree=0.803, adj=0.263, (0 split)

Node number 6: 3154 observations
  predicted class=no   expected loss=0.362397  P(node) =0.0765757
    class counts:  2011  1143
   probabilities: 0.638 0.362

Node number 7: 1151 observations
  predicted class=yes  expected loss=0.2893136  P(node) =0.02794503
    class counts:   333   818
   probabilities: 0.289 0.711
```

FROM THE DECISION TREE,, WE CAN ANALYZE THE VARIABLE IMPORATANCE FIELD TO DETERMINE THE IMPORTANT VARIABLES THAT INFLUENCES THE CERTIFICATE OF DEPOSIT BUYER

13 CHI-SQUARED TEST

In [73]:
```
###################################Chi-squrared Test#####################################

        Cate_ChiSq = data1[,sapply(data1,is.factor)]
        chisqallpvalues <- apply(Cate_ChiSq[-1] , 2 , function(i) stats::chisq.test(table(Cate
        chisqallstatvals <- apply(Cate_ChiSq[-1] , 2 , function(i) stats::chisq.test(table(Cat
        chisq <- data.frame(VARS=names(chisqallpvalues),pval=chisqallpvalues,chistat=chisqalls
        row.names(chisq) <- NULL
        chisq
```

```
Warning message in stats::chisq.test(table(Cate_ChiSq$y, i)):
"Chi-squared approximation may be incorrect"Warning message in stats::chisq.test(table(Cate_Chi
"Chi-squared approximation may be incorrect"Warning message in stats::chisq.test(table(Cate_Chi
```

"Chi-squared approximation may be incorrect"Warning message in stats::chisq.test(table(Cate_Chi
"Chi-squared approximation may be incorrect"

VARS	pval	chistat
marital	2.068015e-26	122.655152
education	3.305189e-38	193.105905
default	5.161958e-89	406.577515
housing	5.829448e-02	5.684496
loan	5.786753e-01	1.094028
contact	1.525986e-189	862.318364
month	0.000000e+00	3101.149351
day_of_week	2.958482e-05	26.144939
poutcome	0.000000e+00	4230.523798
y	0.000000e+00	41177.996927
Age_Binned	1.001547e-52	244.521609

WE PERFORM THE CHI SQUARED TEST TO CALCULATE THE CORRELATION BETWEEN THE CATEGORICAL VARIABLES FROM THE CHI - SQUARED TEST, WE CAN SEE THAT VALUES LESS 0.5 IN THE PVAL COLOUMN INDICATES THE MOST SIGNIFICAT CORRELATION

14 CORRELATION ANALYSIS

In [74]:
```
##########################Correlation PLots##############################################

m = data1[,c(10:12,14:18)]
m$y = as.character(m$y)
m$y = ifelse(m$y=='no',0,1)
cordata <- cor(m)
corrplot(cordata,method = "number")
```

FROM THE CORRELATION PLOT, WE CAN INFER THE CORRELATION BETWEEN THE MOST INFLUENCING VARIABLES WITH THE OUTPUT

15 BUILDING MACHINE LEARNING MODELS

```
In [75]: ########################Building Models#################################################
         data1$y = as.character(data1$y)
         data1$y = ifelse(data1$y=='no',0,1)

         k =5

         auc_glm = rep(NA,k)
         auc_rf = rep(NA,k)
```

```
auc_gbm = rep(NA,k)

i = 1
```

16 LOGISTIC REGRESSION

In [76]:
```
##### Logistic Regression ############

for(i in 1:k)
{

  intrain<-createDataPartition(y=data1$y,p=0.7,list=FALSE)
  cv.train<-data1[intrain,]
  cv.test<-data1[-intrain,]

  cols = c("job","marital","education","default","housing","loan","contact","month","
  for(j in cols)
  {
    id <- which(!(cv.test[,j] %in% levels(cv.train[,j])))
    cv.test[,j][id] <- NA
  }


  fit_glm = glm(y~., data = cv.train,family=binomial(link='logit'))
  pred = predict(fit_glm,cv.test)
  pred
  #acc_glm <- accuracy(cv.test[,18], pred>0.6)

  roc_obj <- roc(cv.test$y,pred)
  auc_glm[i] = roc_obj$auc
  print(paste("AUC Score of Fold ",i,"in Logistic Regression:", auc_glm[i]))

}
```

```
Warning message in predict.lm(object, newdata, se.fit, scale = 1, type = ifelse(type == :
"prediction from a rank-deficient fit may be misleading"
```

```
[1] "AUC Score of Fold  1 in Logistic Regression: 0.775465701261808"
```

```
Warning message in predict.lm(object, newdata, se.fit, scale = 1, type = ifelse(type == :
"prediction from a rank-deficient fit may be misleading"
```

```
[1] "AUC Score of Fold  2 in Logistic Regression: 0.792024208109305"
```

```
Warning message in predict.lm(object, newdata, se.fit, scale = 1, type = ifelse(type == :
"prediction from a rank-deficient fit may be misleading"
```

```
[1] "AUC Score of Fold  3 in Logistic Regression: 0.789076376495837"


Warning message in predict.lm(object, newdata, se.fit, scale = 1, type = ifelse(type == :
"prediction from a rank-deficient fit may be misleading"

[1] "AUC Score of Fold  4 in Logistic Regression: 0.803104130246336"


Warning message in predict.lm(object, newdata, se.fit, scale = 1, type = ifelse(type == :
"prediction from a rank-deficient fit may be misleading"

[1] "AUC Score of Fold  5 in Logistic Regression: 0.796947618078664"
```

LOGISTIC REGRESSION IS A REGRESSION MODEL WHERE THE DEPENDANT VARIABLE IS CATEGORICAL. AUC- AREA UNDER CURVE WE USE AUC SCORE TO CALCULATE THE ACCURACY OF THE MODEL. WE HAVE PERFORMNED A 5 FOLD CROSS VALIDATION FOR THE LOGISITIC REGRESSION AND WE DERIVED AN AVERAGE AUC SCORE OF 0.791.

17 RANDOM FOREST

In [77]:
```
###### Random Forest #########

        for(i in 1:k)
        {

          intrain<-createDataPartition(y=data1$y,p=0.7,list=FALSE)
          cv.train<-data1[intrain,]
          cv.test<-data1[-intrain,]

          fit_rf = randomForest(x = cv.train[,-18],y=as.factor(cv.train[,18]),ntree = 500)
          pred = predict(fit_rf,newdata = cv.test[,-18],type='prob')[,2]

          #acc_rf <- accuracy(cv.test[,18], pred>0.6)

          roc_obj <- roc(cv.test$y,pred)
          auc_rf[i] = roc_obj$auc

          print(paste("AUC Score of Fold ",i,"in Random Forest:", auc_rf[i]))

        }
```

```
[1] "AUC Score of Fold  1 in Random Forest: 0.787599466475685"
[1] "AUC Score of Fold  2 in Random Forest: 0.78952031095929"
[1] "AUC Score of Fold  3 in Random Forest: 0.778839534727809"
[1] "AUC Score of Fold  4 in Random Forest: 0.762347345998677"
[1] "AUC Score of Fold  5 in Random Forest: 0.78612877398247"
```

RANDOM FOREST OR RANDOM DECISION FOREST ARE AN ENSEMBLE LEARNING METHOD FOR CLASSIFICATION, REGRESSION AND OTHER TASKS, THAT OPERATE BY CONSTRUCTING A MULTITUDE OF DECISION TREES AT TRAINING TIME AND OUTPUTTING THE CLASS THAT IS THE MODE OF THE CLASSES (CLASSIFICATION) OR MEAN PREDICTION (REGRESSION) OF THE INDIVIDUAL TREES.

WE HAVE PERFORMNED A 5 FOLD CROSS VALIDATION FOR RANDOM FOREST AND WE DERIVED AN AVERAGE AUC SCORE OF 0.781.

18 Gradient Boosting

```
In [78]: for(i in 1:k)
         {

           intrain<-createDataPartition(y=data1$y,p=0.7,list=FALSE)
           cv.train<-data1[intrain,]
           cv.test<-data1[-intrain,]


           fit_gbm = gbm(formula = y ~.,
                         distribution = "bernoulli",
                         data = cv.train,
                         n.trees = 500,
                         shrinkage = .01
           )

           pred = predict(fit_gbm,cv.test,n.trees = 500,type='response')
           #acc_gbm[i] = accuracy(cv.test[,18], pred>0.6)

           roc_obj <- roc(cv.test$y,pred)
           auc_gbm[i] = roc_obj$auc

           print(paste("AUC Score of Fold ",i,"in Gradient Boosting:", auc_gbm[i]))
         }
```

```
[1] "AUC Score of Fold  1 in Gradient Boosting: 0.791035479624377"
[1] "AUC Score of Fold  2 in Gradient Boosting: 0.792196348167178"
[1] "AUC Score of Fold  3 in Gradient Boosting: 0.785082400733061"
[1] "AUC Score of Fold  4 in Gradient Boosting: 0.788149090435776"
[1] "AUC Score of Fold  5 in Gradient Boosting: 0.775436145793621"
```

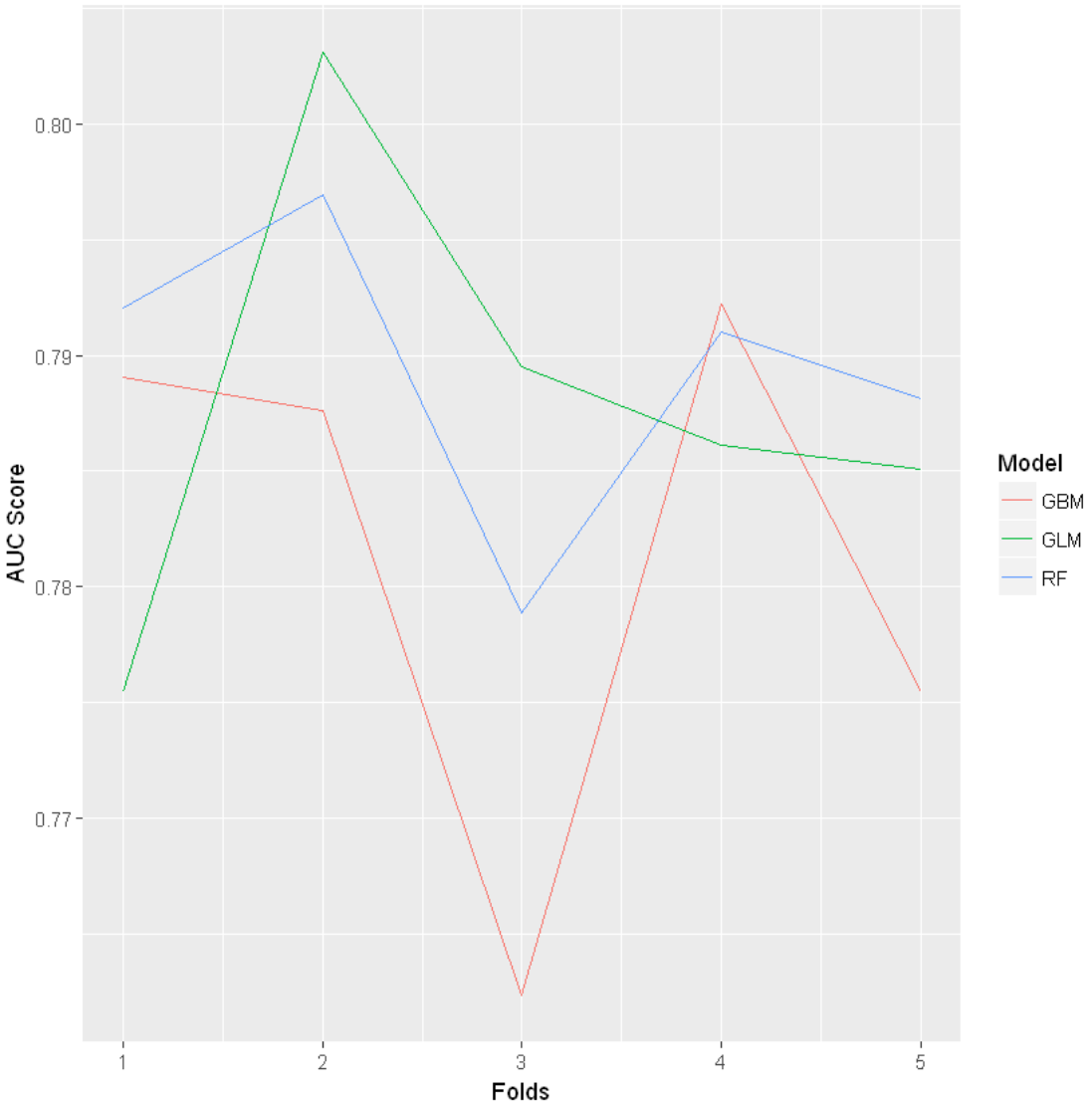
GRADIENT BOOSTING IS A BOOSTING TECHNIQUE. GRADIENT BOOSTING IS A MACHINE LEARNING TECHNIQUE FOR REGRESSION AND CLASSIFICATION PROBLEMS, WHICH PRODUCES A PREDICTION MODEL IN THE FORM OF AN ENSEMBLE OF WEAK PREDICTION MODELS, TYPICALLY DECISION TREES.

WE HAVE PERFORMNED A 5 FOLD CROSS VALIDATION FOR GRADIENT BOOSTING AND WE DERIVED AN AVERAGE AUC SCORE OF 0.786.

19 COMPARING THE PERFORMANCE OF MODELS

```
In [79]: ### Comparing Model Performace ###
         tt = data.frame(Sequence = c(1,2,3),Model = c('GLM','RF','GBM'),AUC_Values = c(auc_gl
         tt = tt[order(tt$Sequence),]
         tt$Seq2 = rep(seq(1,5),3)

         ggplot(data = tt, aes(x=Seq2, y=AUC_Values)) + geom_line(aes(colour=Model)) + labs(x =
```



THE ABOVE VISUALIZATION INDICATES THE AUC SCORE AT EACH CROSS FOLD VALIDATION, IT HELPS US UNDERSTAND WHICH MODEL IS PERFORMING BETTER AND PROVIDING BETTER PREDICTION ACCURACY

20 COMPARING THE PERFORMANCE OF DIFFERENT MODELS

```
In [80]: ### Average AUC Score of each Model
         print(paste("Average AUC after 5 fold CV in Logistic Regression:", round(mean(auc_glm)
         print(paste("Average AUC after 5 fold CV in Random Forest:", round(mean(auc_rf),3)))
         print(paste("Average AUC after 5 fold CV in GBM:", round(mean(auc_gbm),3)))
```

```
[1] "Average AUC after 5 fold CV in Logistic Regression: 0.791"
[1] "Average AUC after 5 fold CV in Random Forest: 0.781"
[1] "Average AUC after 5 fold CV in GBM: 0.786"
```

THE ABOVE RESULTS GIVES THE AVERAGE SCORE OF 5 FOLD CROSS VALIDATION FOR EACH MODEL. IT HELPS US ANALYZING WHICH MODEL HAS BETTER OVERALL PERFORMANCE.

21 WHY CONSIDER AUC- AREA UNDER CURVE INSTEAD OF CALCULATING ACCURACY OF THE MODELS ?

Overall Accuracy means the proportion of correct results that a classifier has achieved. If, from a data set, a classifier could correctly guess the label of half of the examples, then we say it's accuracy was 50%.

Overall Accuracy has something called as the accuracy paradox. When TruePositives < False Positives, then accuracy will always increase when we change a classification rule to always output "negative" category. Conversely, when True Negatives < False Negatives, the same will happen when we change our rule to always output "positive".

To avoid this accuracy paradox in evaluating a model's performance, we use AUC score to calculate the accuracy.