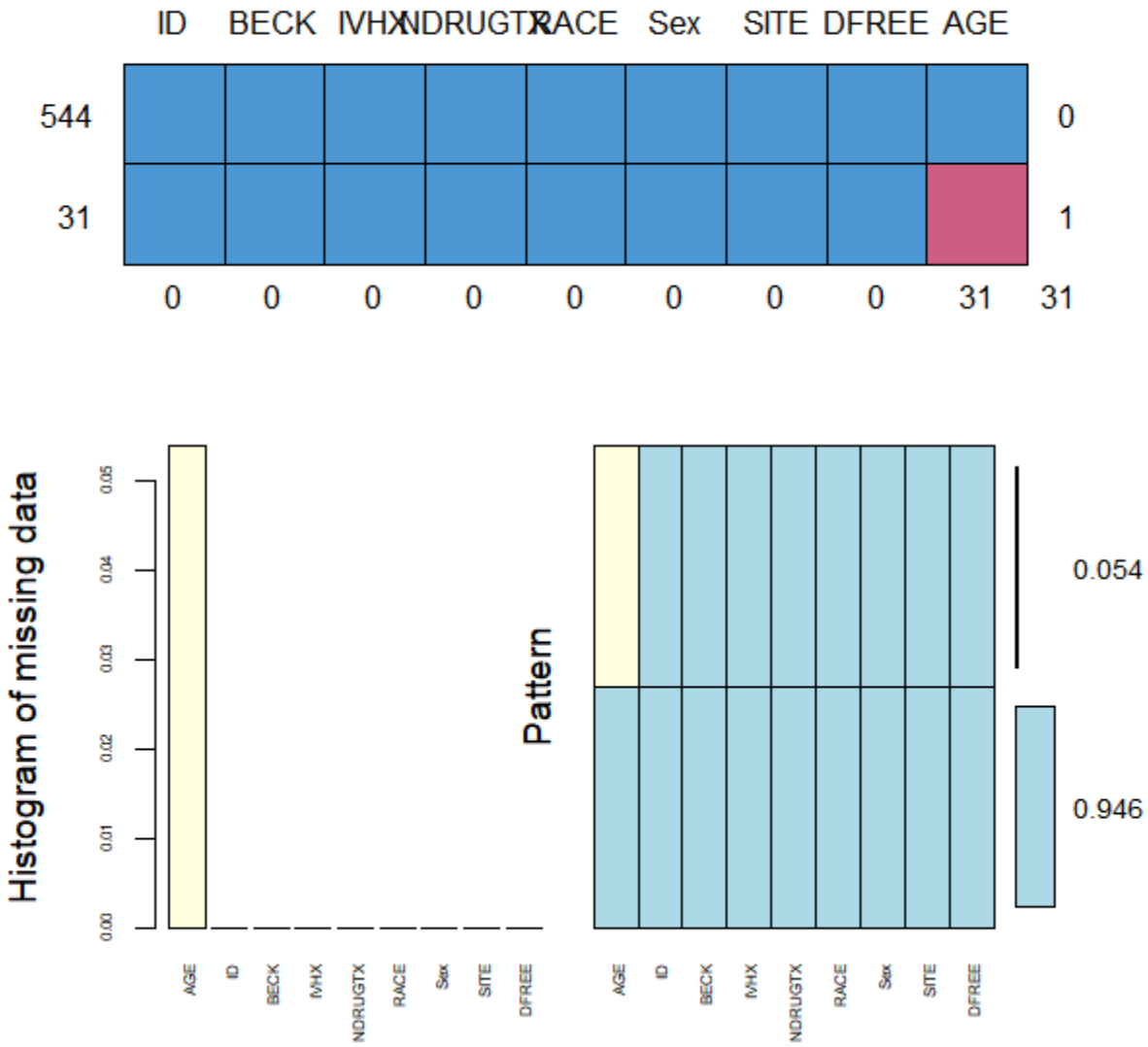
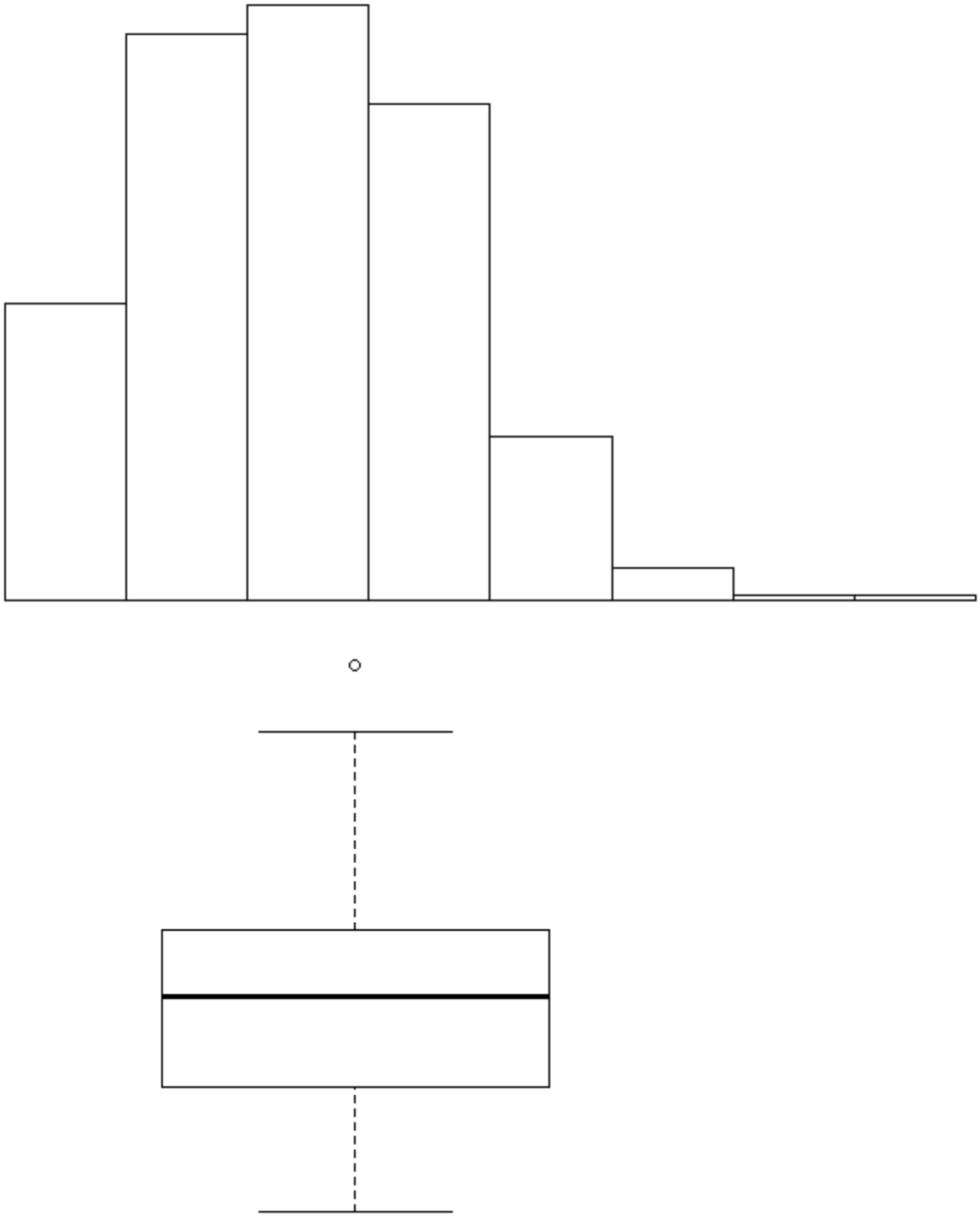
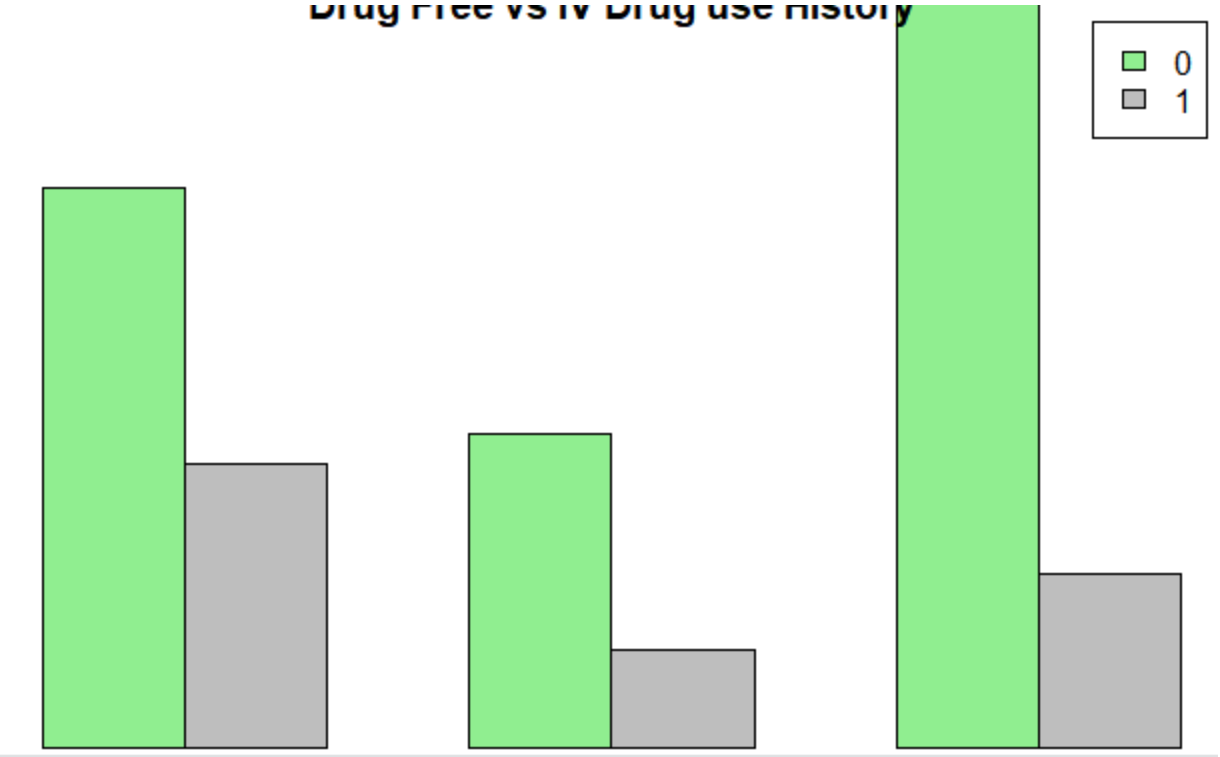
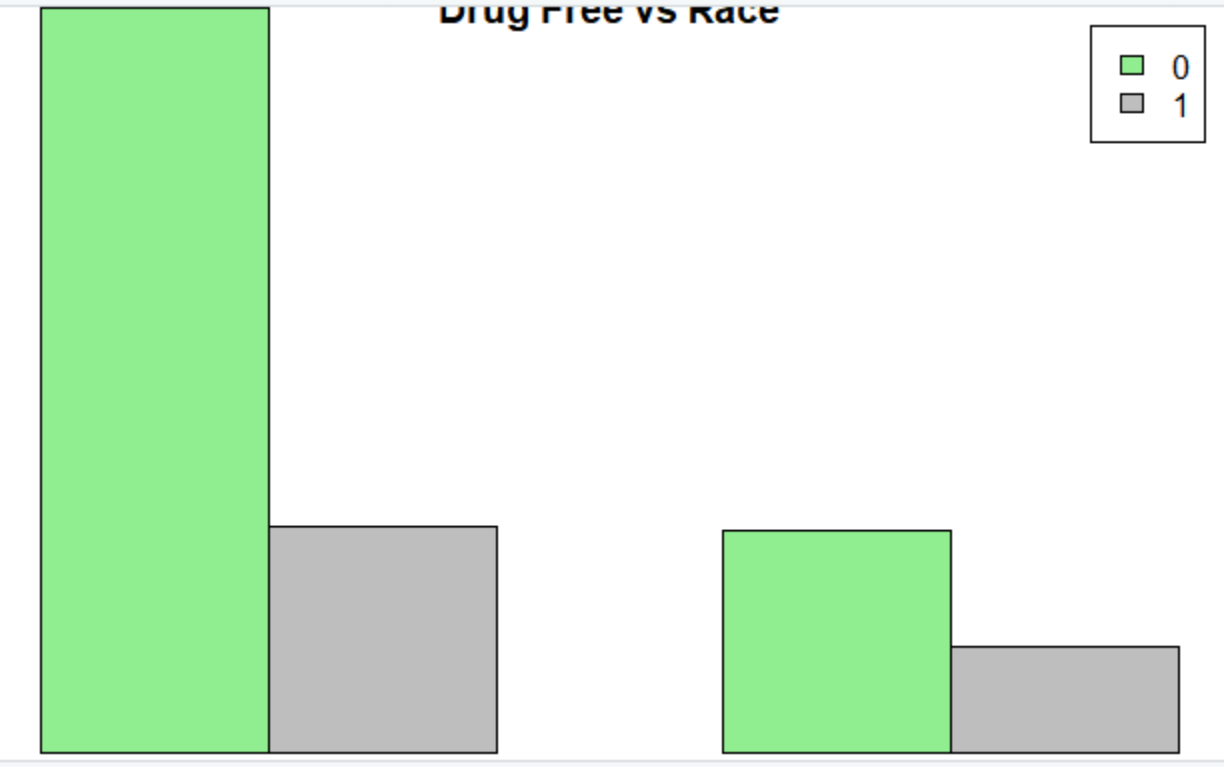
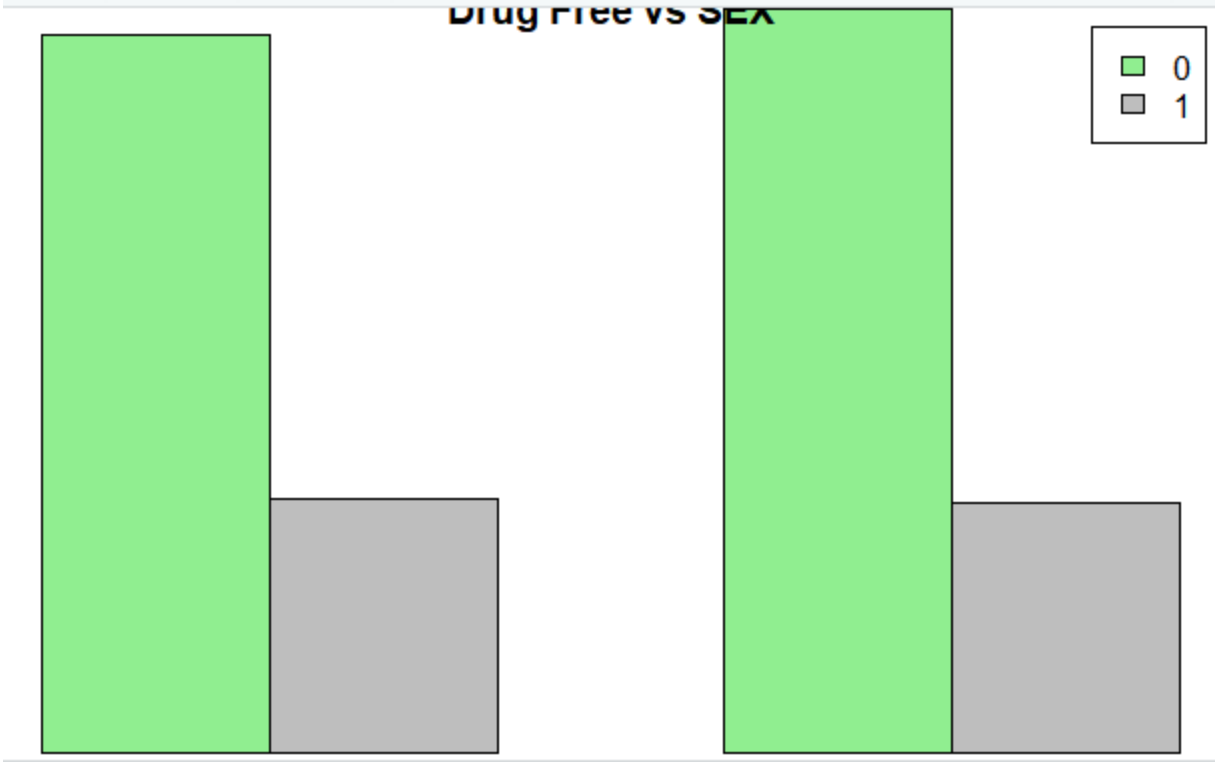
Graph Output

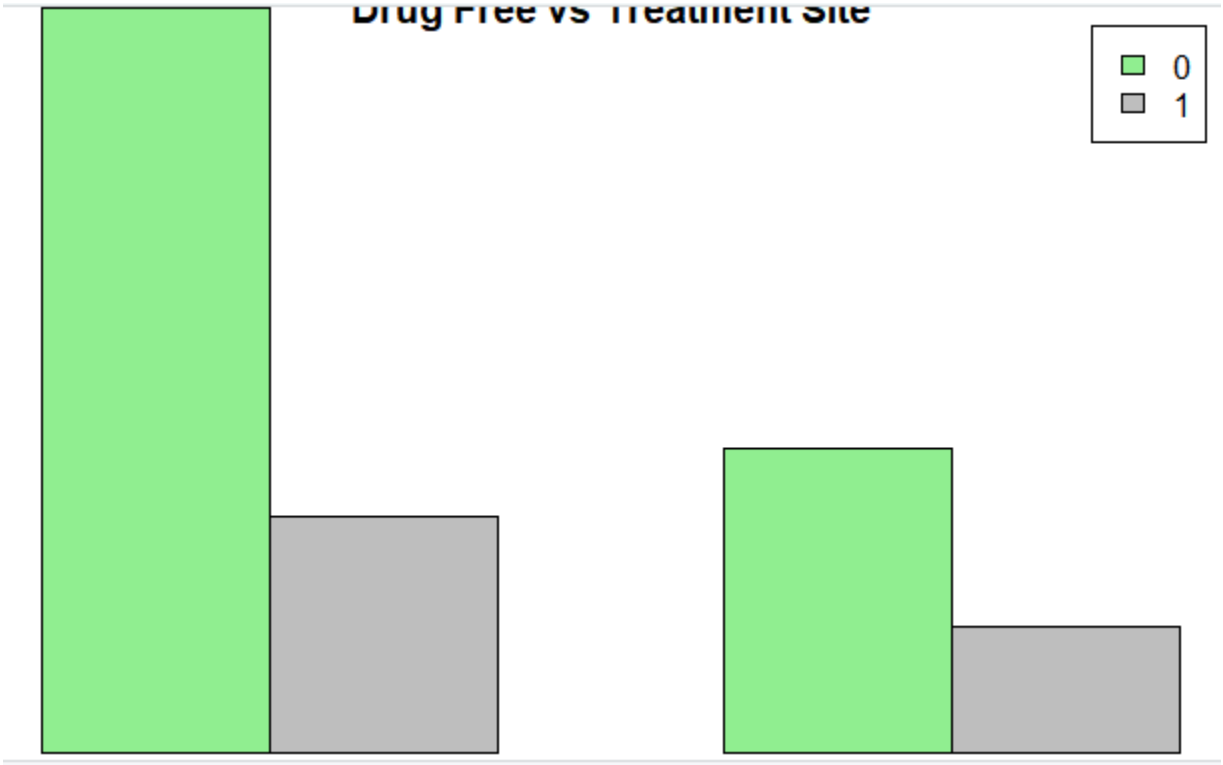


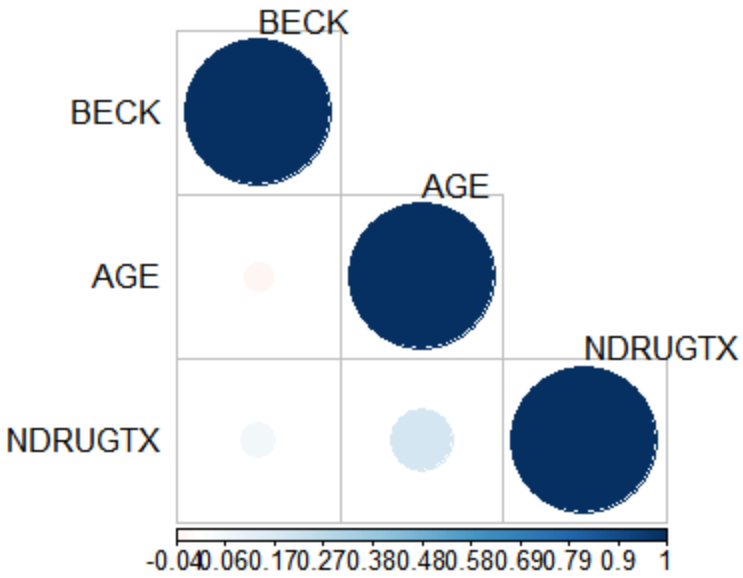


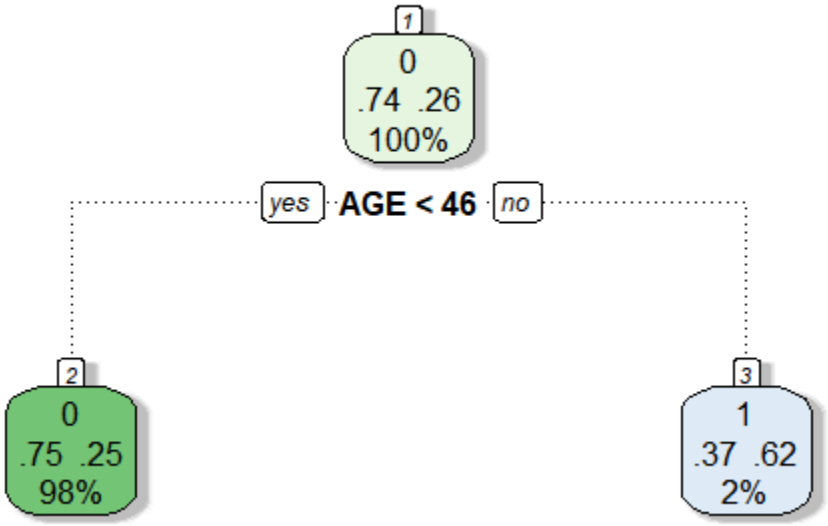


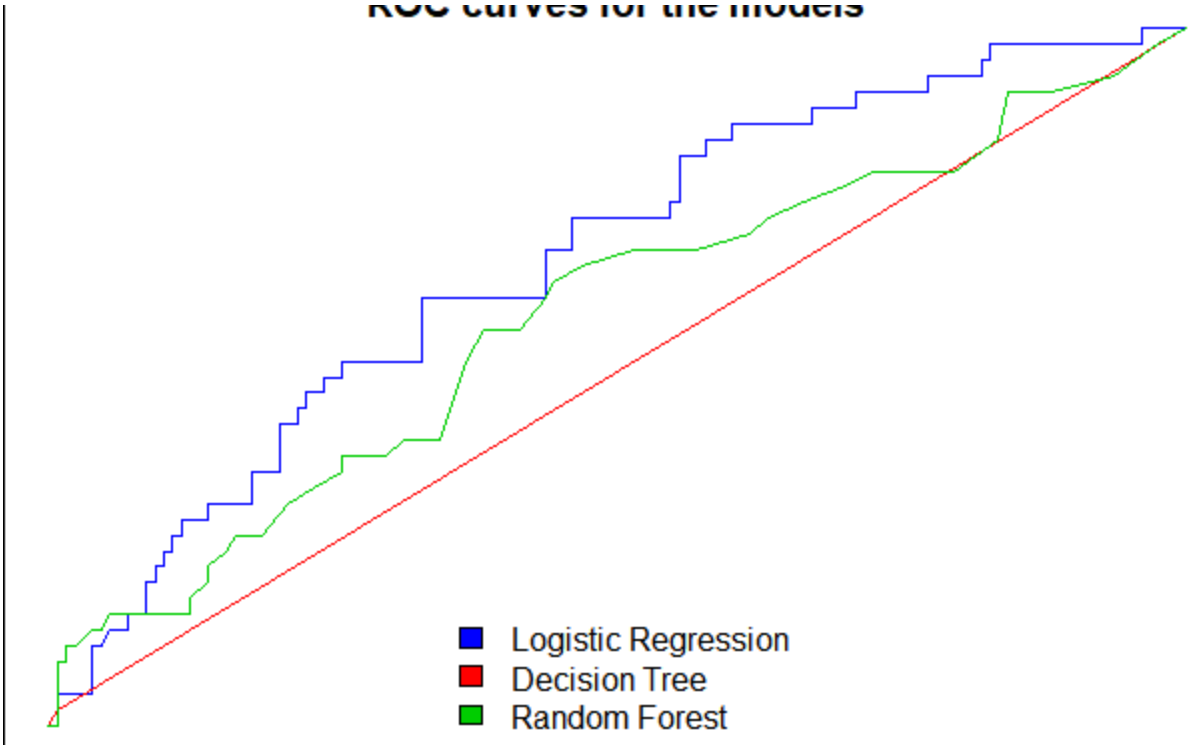







BECK
BECK
AGE
AGE
NDRUGTX
NDRUGTX
-0.04 0.06 0.17 0.27 0.38 0.48 0.58 0.69 0.79 0.9 1


1
0
.74 .26
100%
yes
AGE < 46
no
2
0
.75 .25
98%
3
1
.37 .62
2%



Console Output

```
> set.seed(108)
> work_path = "D:/vaibhav/trend nxt/topgear/R Community/Predictive Model Base
d Logistic Regression-DrugData"
> setwd(work_path)
>
> # Loading & cleaning the dataset
> data <- read.csv("Drug_data.csv", header=T, na.strings=c("","  ","NA"))
> data$X=NULL
> data<-data[!(is.na(data$ID)),] #  Last row is empty
> original_data=data
> head(data)
  ID AGE  BECK IVHX NDRUGTX RACE Sex SITE DFREE
1  1  39  9.00    R       1    0   M    A     0
2  2  33 34.00    P       8    0   M    A     0
3  3  33 10.00    R       3    0   M    A     0
4  4  32 20.00    R       1    0   F    A     0
5  5  NA  5.00    N       5    1   F    A     1
6  6  30 32.55    R       1    0   M    A     0
> data$X=NULL
> summary(data)
       ID              AGE             BECK       IVHX       NDRUGTX
 Min.   :  1.0   Min.   :20.00   Min.   : 0.00   N:223   Min.   : 0.000
 1st Qu.:144.5   1st Qu.:27.75   1st Qu.:10.00   P:109   1st Qu.: 1.000
 Median :288.0   Median :33.00   Median :17.00   R:243   Median : 3.000
 Mean   :288.0   Mean   :32.49   Mean   :17.37           Mean   : 4.543
 3rd Qu.:431.5   3rd Qu.:37.00   3rd Qu.:23.00           3rd Qu.: 6.000
 Max.   :575.0   Max.   :56.00   Max.   :54.00           Max.   :40.000
                 NA's   :31
      RACE        Sex      SITE        DFREE
```

```
 Min.   :0.0000   F:284   A:400   Min.   :0.0000
 1st Qu.:0.0000   M:291   B:175   1st Qu.:0.0000
 Median :0.0000                   Median :0.0000
 Mean   :0.2522                   Mean   :0.2557
 3rd Qu.:1.0000                   3rd Qu.:1.0000
 Max.   :1.0000                   Max.   :1.0000

> str(data)
'data.frame':   575 obs. of  9 variables:
 $ ID     : int  1 2 3 4 5 6 7 8 9 10 ...
 $ AGE    : int  39 33 33 32 NA 30 39 27 40 36 ...
 $ BECK   : num  9 34 10 20 5 ...
 $ IVHX   : Factor w/ 3 levels "N","P","R": 3 2 3 3 1 3 3 3 3 3 ...
 $ NDRUGTX: int  1 8 3 1 5 1 34 2 3 7 ...
 $ RACE   : int  0 0 0 0 1 0 0 0 0 0 ...
 $ Sex    : Factor w/ 2 levels "F","M": 2 2 2 1 1 2 2 1 2 1 ...
 $ SITE   : Factor w/ 2 levels "A","B": 1 1 1 1 1 1 1 1 1 1 ...
 $ DFREE  : int  0 0 0 0 1 0 1 0 0 0 ...
> nrow(data)
[1] 575
> ncol(data)
[1] 9
> any(is.na(data))
[1] TRUE
> sum(is.na(data))
[1] 31
>
> # HAving a look at missing data
> md.pattern(data)
    ID BECK IVHX NDRUGTX RACE Sex SITE DFREE AGE
544  1    1    1       1    1   1    1     1   1  0
31   1    1    1       1    1   1    1     1   0  1
     0    0    0       0    0   0    0     0  31 31
> aggr_plot <- aggr(data, col=c('LightBlue','LightYellow'), numbers=TRUE, sor
tVars=TRUE, labels=names(data), cex.axis=.5,cex.numbers=.9, gap=1, ylab=c("Hi
stogram of missing data","Pattern"))

 Variables sorted by number of missings:
 Variable      Count
      AGE 0.05391304
       ID 0.00000000
     BECK 0.00000000
     IVHX 0.00000000
  NDRUGTX 0.00000000
     RACE 0.00000000
      Sex 0.00000000
     SITE 0.00000000
    DFREE 0.00000000
>
> # Replacing the NAs
> hist(data$AGE)
> boxplot(data$AGE)
> Age =subset(data, is.na(data$AGE)== F,select=c(AGE))$AGE
> range(Age)
[1] 20 56
> mean(Age)
[1] 32.49265
> m=median(Age)
> data$AGE[is.na(data$AGE)] = m
>
> # cleaning further
> data$ID= NULL
> data$RACE= as.factor(data$RACE)
> data$DFREE=as.factor(data$DFREE)
```

```
>
> # Exploratory Analysis
>
> plottable1=table(data$DFREE,data$IVHX)
> barplot(plottable1, main="Drug Free vs IV Drug use History", xlab="Drug Use
History",col=c("LightGreen","Grey"),legend=rownames(plottable1),beside = TRUE
)
> plottable2=table(data$DFREE,data$RACE)
> barplot(plottable2, main="Drug Free vs Race", xlab="Race",col=c("LightGreen
","Grey"),legend=rownames(plottable2),beside = TRUE)
> plottable3=table(data$DFREE,data$Sex)
> barplot(plottable3, main="Drug Free vs SEX", xlab="SEX",col=c("LightGreen",
"Grey"),legend=rownames(plottable3),beside = TRUE)
> plottable4=table(data$DFREE,data$SITE)
> barplot(plottable4, main="Drug Free vs Treatment Site", xlab="Treatment Sit
e",col=c("LightGreen","Grey"),legend=rownames(plottable4),beside = TRUE)
>
> # Correlation Analysis
> numeric_features= data[c("AGE","BECK","NDRUGTX")]
> corTable=cor(numeric_features)
> corTable
                AGE         BECK    NDRUGTX
AGE       1.00000000 -0.04108021 0.18957761
BECK     -0.04108021  1.00000000 0.05925075
NDRUGTX   0.18957761  0.05925075 1.00000000
> corrplot( cor(as.matrix(numeric_features), method = "pearson", use = "compl
ete.obs") ,is.corr = FALSE, type = "lower", order = "hclust", tl.col = "black
", tl.srt = 360)
>
> # Splitting the Data
> set.seed(108)
> split=sample.split(data$DFREE,SplitRatio = .7)
> train=subset(data,split==T)
> test=subset(data,split==F)
>
> ## Model Building & CV
>
> # Before selecting attributes, we try to check the attributes significance
> cf1 <- cforest(DFREE ~. , data=train, control=cforest_unbiased(mtry=2,ntree
=50))
> varimp(cf1) # get variable importance, based on mean decrease in accuracy
         AGE          BECK          IVHX      NDRUGTX          RACE           Sex
 0.003648649  0.001891892  0.005675676  0.003918919  0.001756757  0.002972973
        SITE
-0.003108108
>
> # GLM
> model1=glm(DFREE~.-SITE-Sex-RACE,data=train,family = binomial)
> predGlm=predict(model1,type="response",newdata=test)
> summary(model1)

Call:
glm(formula = DFREE ~ . - SITE - Sex - RACE, family = binomial,
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.2985  -0.8094  -0.6481   1.2421   2.2755

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.969535   0.710591  -2.772  0.00558 **
AGE          0.049489   0.021258   2.328  0.01991 *
BECK        -0.006284   0.012921  -0.486  0.62675
```

```
IVHXP       -0.537674   0.338697  -1.587  0.11240
IVHXR       -0.821401   0.290644  -2.826  0.00471 **
NDRUGTX     -0.045020   0.027652  -1.628  0.10351
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 458.12  on 402  degrees of freedom
Residual deviance: 440.45  on 397  degrees of freedom
AIC: 452.45

Number of Fisher Scoring iterations: 4

> # Computing accuracy
> table(test$DFREE,predGlm>.5)

    FALSE TRUE
  0   127    1
  1    44    0
> (127+0)/(127+0+1+44)
[1] 0.7383721
>
> # Decision Tree Model
> set.seed(108)
> numFolds = trainControl( method = "cv", number = 10 )
> cpGrid = expand.grid( .cp = seq(0.01,0.5,0.01))
> train(DFREE~AGE,data=train,method="rpart",trControl=numFolds,tuneGrid=cpGri
d)
CART

403 samples
  1 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 363, 363, 363, 362, 362, 363, ...
Resampling results across tuning parameters:

  cp    Accuracy   Kappa
  0.01  0.7444512   0.008004053
  0.02  0.7420122  -0.004761905
  0.03  0.7420122  -0.004761905
  0.04  0.7445122   0.000000000
  0.05  0.7445122   0.000000000
  0.06  0.7445122   0.000000000
  0.07  0.7445122   0.000000000
  0.08  0.7445122   0.000000000
  0.09  0.7445122   0.000000000
  0.10  0.7445122   0.000000000
  0.11  0.7445122   0.000000000
  0.12  0.7445122   0.000000000
  0.13  0.7445122   0.000000000
  0.14  0.7445122   0.000000000
  0.15  0.7445122   0.000000000
  0.16  0.7445122   0.000000000
  0.17  0.7445122   0.000000000
  0.18  0.7445122   0.000000000
  0.19  0.7445122   0.000000000
  0.20  0.7445122   0.000000000
  0.21  0.7445122   0.000000000
  0.22  0.7445122   0.000000000
  0.23  0.7445122   0.000000000
```

```
  0.24  0.7445122   0.000000000
  0.25  0.7445122   0.000000000
  0.26  0.7445122   0.000000000
  0.27  0.7445122   0.000000000
  0.28  0.7445122   0.000000000
  0.29  0.7445122   0.000000000
  0.30  0.7445122   0.000000000
  0.31  0.7445122   0.000000000
  0.32  0.7445122   0.000000000
  0.33  0.7445122   0.000000000
  0.34  0.7445122   0.000000000
  0.35  0.7445122   0.000000000
  0.36  0.7445122   0.000000000
  0.37  0.7445122   0.000000000
  0.38  0.7445122   0.000000000
  0.39  0.7445122   0.000000000
  0.40  0.7445122   0.000000000
  0.41  0.7445122   0.000000000
  0.42  0.7445122   0.000000000
  0.43  0.7445122   0.000000000
  0.44  0.7445122   0.000000000
  0.45  0.7445122   0.000000000
  0.46  0.7445122   0.000000000
  0.47  0.7445122   0.000000000
  0.48  0.7445122   0.000000000
  0.49  0.7445122   0.000000000
  0.50  0.7445122   0.000000000

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.5.
> decisionTreeModel=rpart(DFREE~AGE,data=train,method="class",cp=.01)
> rpart.plot(decisionTreeModel,extra=104, box.palette="GnBu",branch.lty=3, sh
adow.col="gray", nn=TRUE)
> # Accuracy of Decision Tree
> predDT=predict(decisionTreeModel,newdata = test,type = "class")
> table(test$DFREE,predDT)
   predDT
      0   1
  0 127   1
  1  43   1
> # Accuracy
> (127+1)/(127+1+1+43)
[1] 0.744186
>
> # Random Forest
> set.seed(108)
> randomForestModel=randomForest(DFREE~.-SITE-Sex-RACE,data=train,ntree=100,n
odesize=18)
> predictRF=predict(randomForestModel,newdata=test)
> table(test$DFREE,predictRF)
   predictRF
      0   1
  0 123   5
  1  38   6
> # Accuracy of RF
> (126+3)/(126+3+2+41)
[1] 0.75
>
> # AUC Calculation
> glm_ROC=predict(model1,test,type="response")
> pred_glm=prediction(glm_ROC,test$DFREE)
> perf_glm=performance(pred_glm,"tpr","fpr")
>
> dt_ROC=predict(decisionTreeModel,test)
```

```
> pred_dt=prediction(dt_ROC[,2],test$DFREE)
> perf_dt=performance(pred_dt,"tpr","fpr")
>
> RF_ROC=predict(randomForestModel,test,type="prob")
> pred_RF=prediction(RF_ROC[,2],test$DFREE)
> perf_RF=performance(pred_RF,"tpr","fpr")
>
> auc_glm <- performance(pred_glm,"auc")
> auc_glm <- round(as.numeric(auc_glm@y.values),3)
> print(paste('AUC of Logistic Regression:',auc_glm))
[1] "AUC of Logistic Regression: 0.668"
>
> auc_dt <- performance(pred_dt,"auc")
> auc_dt <- round(as.numeric(auc_dt@y.values),3)
> print(paste('AUC of Decision Tree:',auc_dt))
[1] "AUC of Decision Tree: 0.507"
>
> auc_RF <- performance(pred_RF,"auc")
> auc_RF <- round(as.numeric(auc_RF@y.values),3)
> print(paste('AUC of Random Forest:',auc_RF))
[1] "AUC of Random Forest: 0.585"
>
> # ROC Curves
> plot(perf_glm, main = "ROC curves for the models", col='blue')
> plot(perf_dt,add=TRUE, col='red')
> plot(perf_RF, add=TRUE, col='green3')
> legend('bottom', c("Logistic Regression", "Decision Tree", "Random Forest")
, fill = c('blue','red','green3'), bty='n')
```