Naive Bayes

Arman Aghamyan

May 13, 2019

Problem 1

```r
library(caret)
```

```
## Loading required package: lattice
## Loading required package: ggplot2
```

```r
library(ROCR)
```

```
## Loading required package: gplots
##
## Attaching package: 'gplots'
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(e1071)

Income=c(125,100,70,120,150,60,220,85,75,90,180,200,250,50)
Howner=c("Yes","No","No","Yes","No","No","Yes","No","No","No","Yes","Yes","Yes","No")
Default=c("No","No","No","No","Yes","No","No","Yes","No","Yes","Yes","No","No","Yes")
Mstatus=c("Single","Married","Single","Married","Divorced","Married","Divorced","Single","Married","Sing


income_no=c(60,70,75,100,120,125,200,220,250)
mean(income_no)
```

```
## [1] 135.5556
```

```r
sd(income_no)
```

```
## [1] 70.42036
```

```r
income_yes=c(50,85,90,150,180)
mean(income_yes)
```

```
## [1] 111
```

```r
sd(income_yes)
```

```
## [1] 52.72571
```

```r
# P(Hone Owner: Yes, Marital Status: Single, Annual Income: 158)

P_income_no=pnorm(158,135.556,70.42036)
P_income_yes=pnorm(158,111,52.72571)
P_income_yes
```

```
## [1] 0.8136442
```

```r
P_income_no
```

```
## [1] 0.6250285
```

```r
table(Default)
```

```
## Default
##  No Yes
##   9   5
```

```r
addmargins(table(Howner,Default))
```

```
##       Default
## Howner No Yes Sum
##    No   4   4   8
##    Yes  5   1   6
##    Sum  9   5  14
```

```r
addmargins(table(Mstatus,Default))
```

```
##           Default
## Mstatus    No Yes Sum
##   Divorced  2   2   4
##   Married   5   1   6
##   Single    2   2   4
##   Sum       9   5  14
```

```r
P_default_yes<-5/14
P_default_no<-9/14
P_how_yes<-1/5
P_how_no<-5/9
p_mst_single_yes<-2/5
p_mst_single_no<-2/9
p_mst_married_yes<-1/5
p_mst_married_no<-5/9
p_mst_divorced_yes<-2/5
p_mst_divorced_no<-2/9



P_h_yes=prod(P_how_yes,p_mst_single_yes,P_income_yes,P_default_yes)
P_h_yes
```

```
## [1] 0.02324698
```

```r
P_h_no=prod(P_how_no,p_mst_single_no,P_income_no,P_default_no)
P_h_no
```

```
## [1] 0.04960544
```

```r
max(P_h_no,P_h_yes)
```

```
## [1] 0.04960544
```

```r
# so in this case we would predict no
```

A)

P(c|x)=P(c|x)*P(c)/P(x)

P(c|x) is the posterior probability of class (c,target) given predictor (x, attributes). P(c) is the prior probability of class. P(x|c) is the likelihood which is the probability of predictor given class.- class conditional P(x) is the prior probability of predictor.

As P(x) is the same for all categories it can be ignored and we maximize only P(c).

B) Prediction is no, because probability of h_no was higher than h_yes.

C) R and B are conditionally independent [given Y] if and only if, given knowledge of whether Y occurs, knowledge of whether R occurs provides no information on the likelihood of B occurring, and knowledge of whether B occurs provides no information on the likelihood of R occurring.

Example

Linda was comming to AUA by train,while John was driving to AUA. Suppose {Linda late} variable and {John late} variable are conditionally independent given {Train strike} variable if and only if,given knowledge that {Train strike} occurs,knowledge of whether {Linda} late occurs provides no information on the likelihood of {John late } occurring, and knowledge of whether {John late} occurs provides no information on the likelihood of {Linda late } occuring.

Problem 2

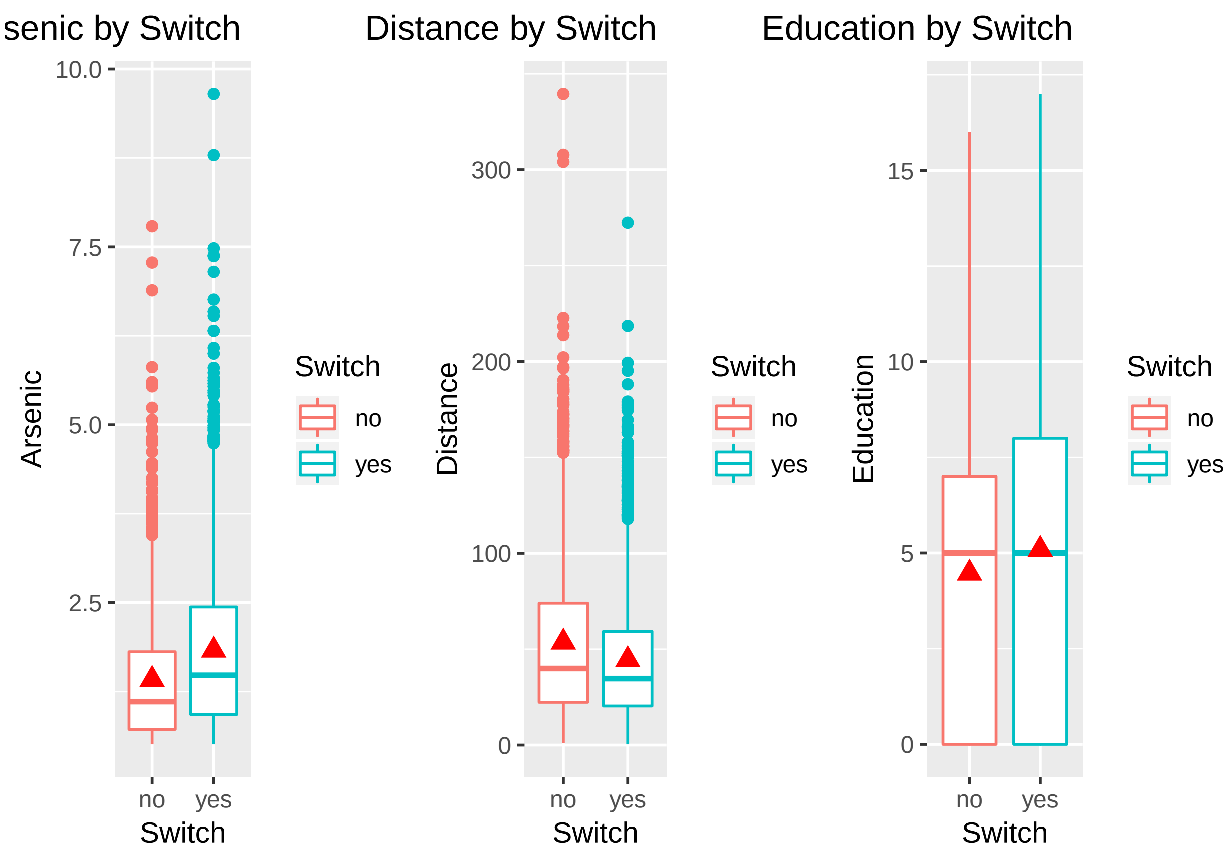
```
library(carData)
library(gridExtra)

# A)
data("Wells")
head(Wells)
```

```
##   switch arsenic distance education association
## 1    yes    2.36   16.826         0          no
## 2    yes    0.71   47.322         0          no
## 3     no    2.07   20.967        10          no
## 4    yes    1.15   21.486        12          no
## 5    yes    1.10   40.874        14         yes
## 6    yes    3.90   69.518         9         yes
```

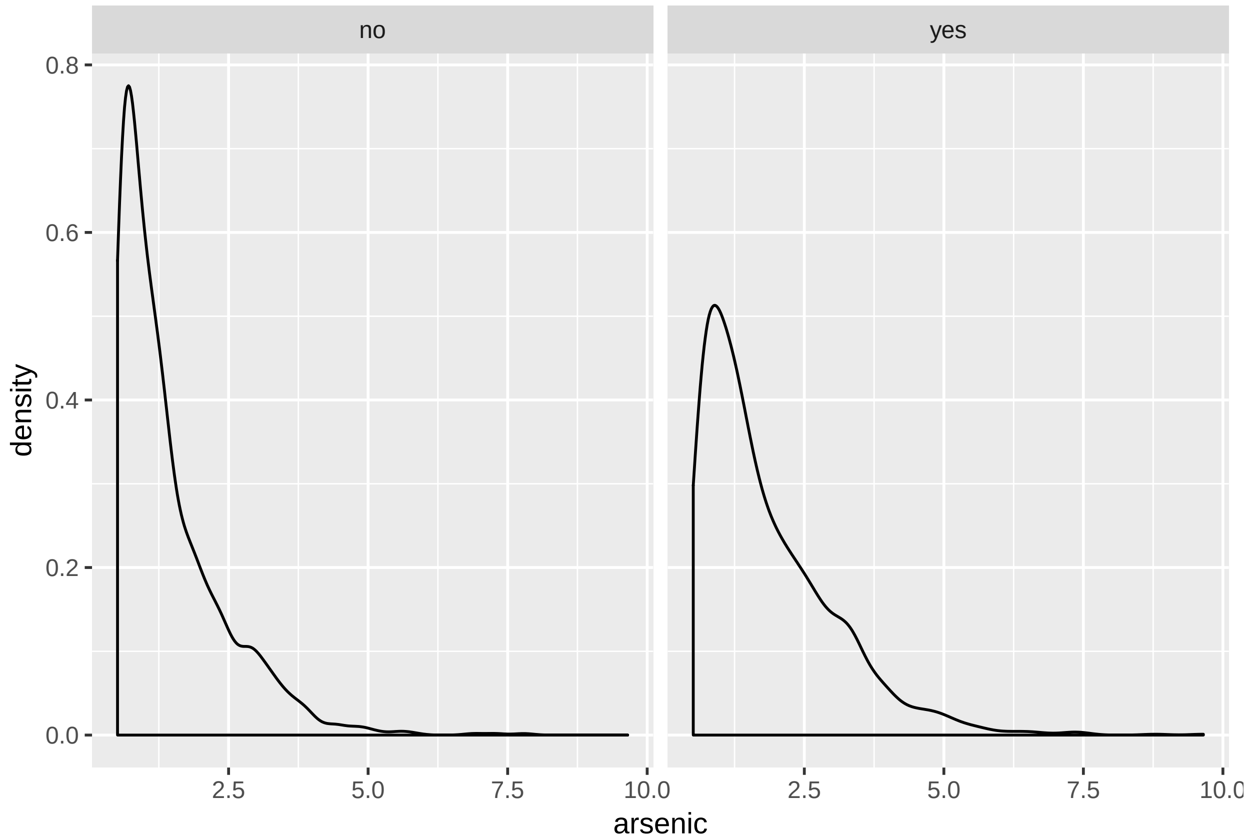
```
library(ggplot2)
b1<-ggplot(data = Wells, aes(y =arsenic, x = switch,color = switch))+
  geom_boxplot()+
  scale_x_discrete(labels= levels(Wells$switch) )+
  stat_summary(fun.y = "mean",geom = 'point',col='red', shape=17, size=3)+
  labs(title ="Arsenic by Switch " ,color =  "Switch",y='Arsenic',x="Switch")+
  theme(plot.title = element_text(hjust = 1))
b2<-ggplot(data = Wells, aes(y =distance, x = switch,color = switch))+
  geom_boxplot()+
  scale_x_discrete(labels= levels(Wells$switch) )+
  stat_summary(fun.y = "mean",geom = 'point',col='red', shape=17, size=3)+
  labs(title ="Distance by Switch " ,color =  "Switch",y='Distance',x="Switch")+
  theme(plot.title = element_text(hjust = 1))
b3<-ggplot(data = Wells, aes(y =education, x = switch,color = switch))+
  geom_boxplot()+
  scale_x_discrete(labels= levels(Wells$switch) )+
  stat_summary(fun.y = "mean",geom = 'point',col='red', shape=17, size=3)+
  labs(title ="Education by Switch " ,color =  "Switch",y='Education',x="Switch")+
  theme(plot.title = element_text(hjust = 1))

grid.arrange(b1,b2,b3,nrow=1)
```

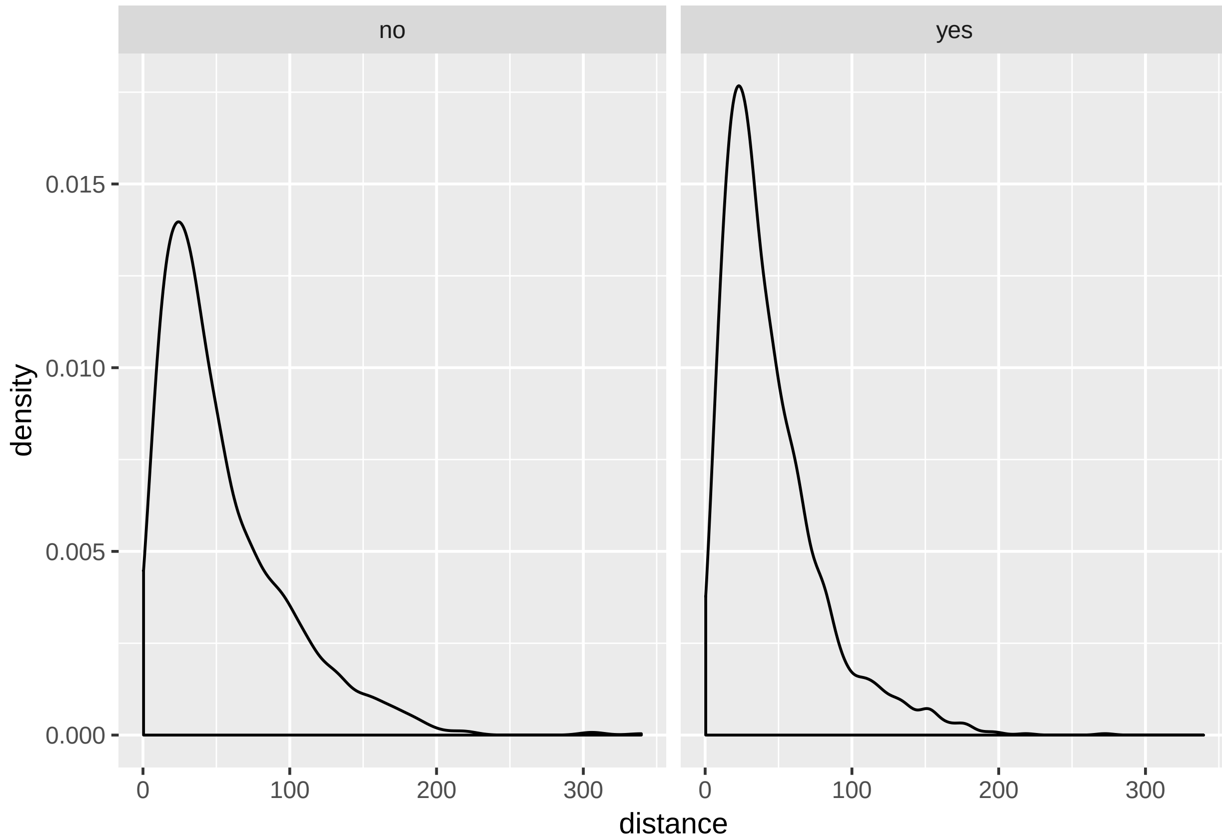


A) As we see by Arsenic and Education variables number of switched household is higher than unswitched(we have outliers),while by ditstance unswitched households get higher.

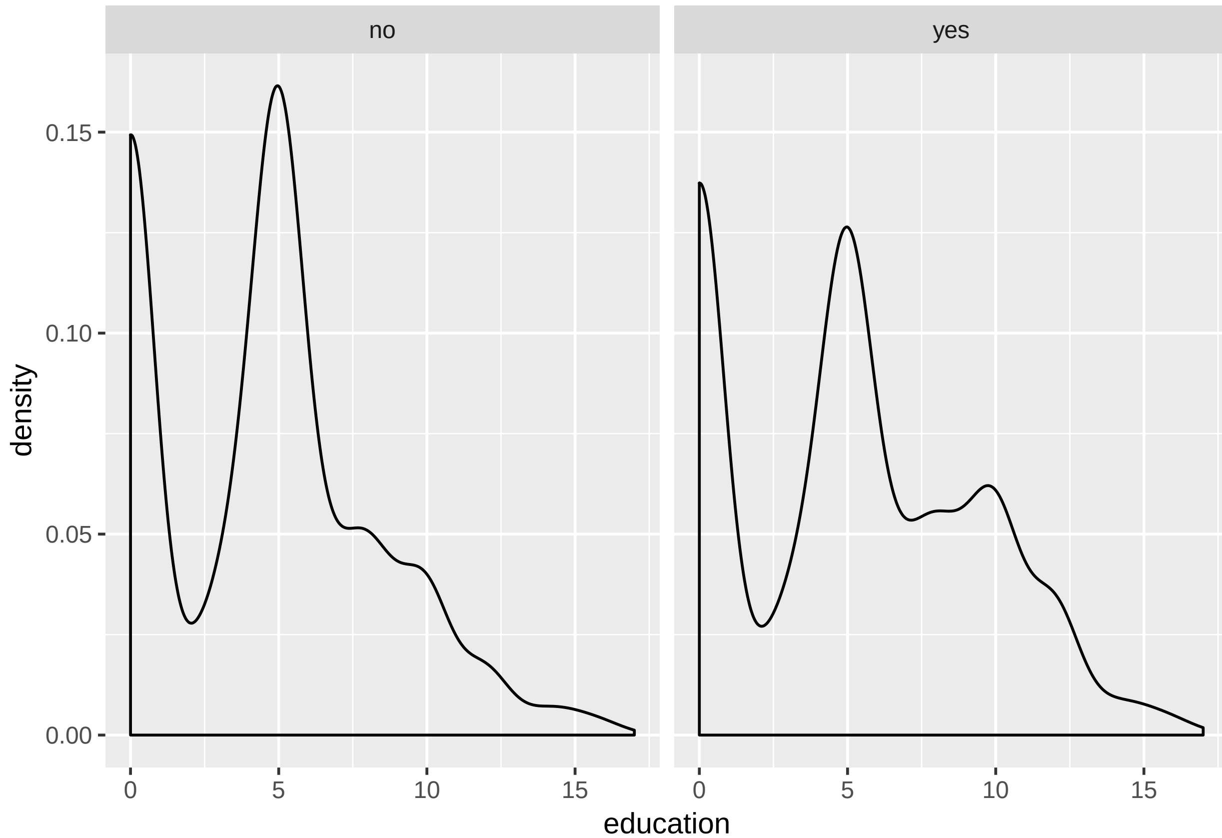
```
d1<-ggplot(Wells, aes(x =arsenic)) +
  geom_density() +
  facet_wrap(~switch)
d1
```




```
d2<-ggplot(Wells, aes(x =distance)) +
  geom_density() +
  facet_wrap(~switch)
d2
```




```
d3<-ggplot(Wells, aes(x =education)) +
  geom_density() +
  facet_wrap(~switch)
d3
```



From the Density plot it is seen that for arsenic,distance and education variables Switch Yes is wider than Switch No and Switch Yes has more variance than Switch No.

```
#B
set.seed(1)
index<-createDataPartition(Wells$switch,p=0.8,list = F)
Train<-Wells[index,]
Test<-Wells[-index,]

model<-naiveBayes(switch~.,data=Train,laplace = 1)
names(model)
```

```
## [1] "apriori" "tables"  "levels"  "call"
```

```
model$apriori
```

```
## Y
##   no  yes
## 1027 1390
```

```
Prob_Yes<-model$apriori[2]/sum(model$apriori)
Prob_Yes
```

```
##       yes
## 0.5750931
```

```
Prob_No<-model$apriori[1]/sum(model$apriori)
Prob_No
```

```
##        no
```

```
## 0.4249069
```

```r
model$tables
```

```
## $arsenic
##      arsenic
## Y          [,1]      [,2]
##   no  1.418647 0.9544372
##   yes 1.823986 1.1825658
##
## $distance
##      distance
## Y          [,1]     [,2]
##   no  54.10339 42.91848
##   yes 44.34009 34.14521
##
## $education
##      education
## Y          [,1]     [,2]
##   no  4.378773 3.674852
##   yes 5.076259 4.109193
##
## $association
##      association
## Y             no       yes
##   no  0.5558795 0.4441205
##   yes 0.5797414 0.4202586
```

```r
Pred_no<-pnorm(2.5,1.421801,0.9695603)
Pred_no
```

```
## [1] 0.8669416
```

```r
Pred_yes<-pnorm(2.5,1.823612,1.1798932)
Pred_yes
```

```
## [1] 0.7167664
```

```r
max(Pred_no,Pred_yes) # so by probabilities it predicts No
```

```
## [1] 0.8669416
```

```r
pred_test<-predict(model,newdata = Test)
confusionMatrix(pred_test,Test$switch,positive = "yes")
```

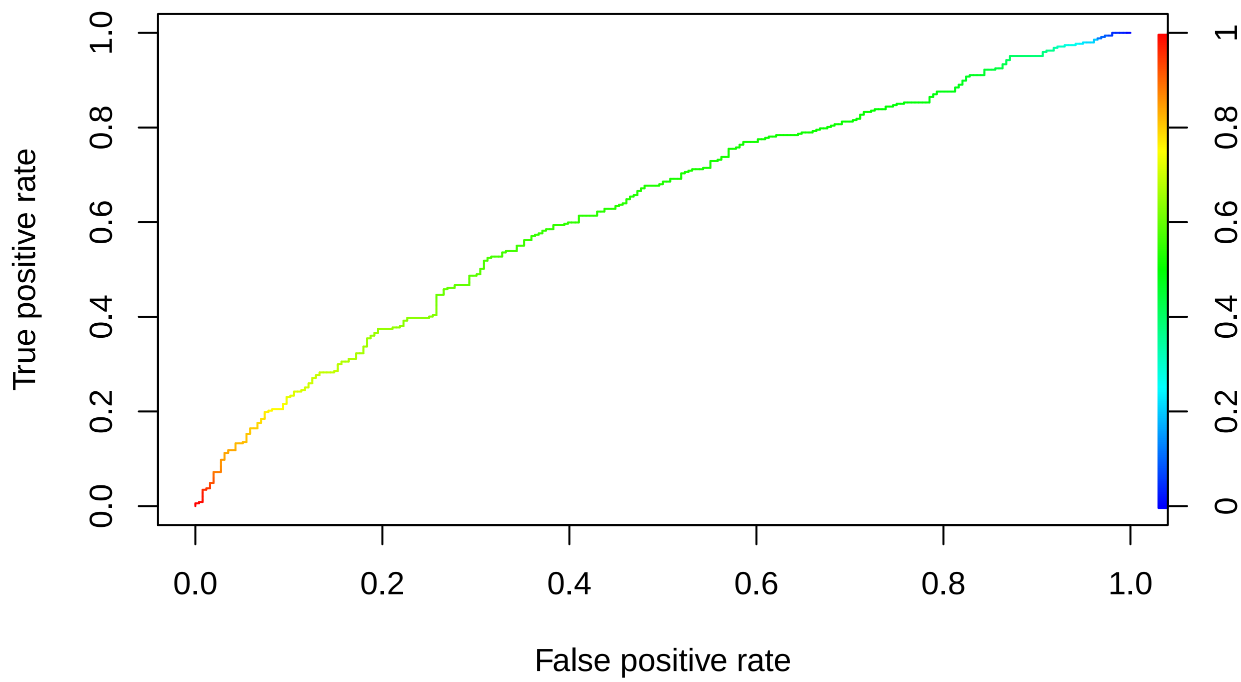
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  no yes
##        no   68  56
##        yes 188 291
##
##                Accuracy : 0.5954
##                  95% CI : (0.555, 0.6348)
##     No Information Rate : 0.5755
##     P-Value [Acc > NIR] : 0.1718
##
##                   Kappa : 0.1118
```

```
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.8386
##             Specificity : 0.2656
##          Pos Pred Value : 0.6075
##          Neg Pred Value : 0.5484
##              Prevalence : 0.5755
##          Detection Rate : 0.4826
##    Detection Prevalence : 0.7944
##       Balanced Accuracy : 0.5521
##
##        'Positive' Class : yes
##
```

```r
pred_test_prob<-predict(model,newdata = Test,type = "raw")
head(pred_test_prob)
```

```
##             no       yes
## [1,] 0.3483484 0.6516516
## [2,] 0.4083254 0.5916746
## [3,] 0.1969404 0.8030596
## [4,] 0.5693896 0.4306104
## [5,] 0.7994622 0.2005378
## [6,] 0.5922560 0.4077440
```

```r
P_Test<-prediction(pred_test_prob[,2],Test$switch)
perf<-performance(P_Test,"tpr","fpr")
plot(perf,colorize=T)
```




```r
performance(P_Test,"auc")@y.values
```

```
## [[1]]
## [1] 0.6303472
```

B) Probability of predicting Yes is 0,58 and for predicting No is 0,42.

C) Probability of switch-No given assosiation-Yes is 0.44. - for categorical

Probability of a new observation where arsenic is 2.5 it predicts No. - for numeric

For numeric variable we used Density function to get probabilies.

For categorical variables we use conditional probabilities.

D) Overall accuracy is 0.597, Sensitivity is 0.86 and as we interested in prediction 'Yes' it is good result as benchmark was 0,58 percent.

Yes, the Prior probability can influence the final result as it is used in function. AUC is 0.65.