STA 588 Final Project

By Smita Sukhadeve and Ttiana Romanchishina

Appendix C: Code and Output

In [144]:
```
library(randomForest)
library(RColorBrewer)
```

In [2]:
```
lon <- read.csv("/Users/tatiana/Desktop/DATA_ANALYTICS/all_meps_fil
es/2012 - longitudinal.csv", na.strings = c(-1,-9))
head(lon)
```

Out[2]:

	X	DUID	PID	DUPERSID	PANEL	YEARIND	ALL5RDS	DIED	INST	MILITARY
1	1	20004	101	20004101	17	1	1	0	0	0
2	2	20004	102	20004102	17	1	1	0	0	0
3	3	20004	103	20004103	17	1	1	0	0	0
4	4	20005	101	20005101	17	1	1	0	0	0
5	5	20005	102	20005102	17	1	1	0	0	0
6	6	20005	103	20005103	17	1	0	0	0	1

In [3]:
```
dim(lon)
```

Out[3]: 17923 3496

In [74]:
```
lon_few_missing <- lon[ , colSums(is.na(lon)) <= 500 ]
dim(lon_few_missing)
```

Out[74]: 17923 2388

In [78]:
```
lon_no_missing <- na.omit(lon_few_missing)
dim(lon_no_missing)
lon_wt <- lon_no_missing[,c(4, 14:ncol(lon_no_missing))]
dim(lon_wt)
```

Out[78]: 16274 2388

Out[78]: 16274 2376

In [81]:
```
lon_all_num <- lon_wt[,sapply(lon_wt,is.numeric)]
dim(lon_all_num)

lon_wt_y1 <- lon_all_num[,grepl("Y1",names(lon_all_num))]
dim(lon_wt_y1)
```

Out[81]: 16274 2356

Out[81]: 16274 989

In [6]:
```
Response <- lon_wt$IPNGTDY2
Response[Response > 0] <- 1
#Response <- factor(Response)
```

In [7]:
```
wt <- lon_wt_y1[,sapply(lon_wt_y1,
                        function(x) sum(length(unique(x))) > 100)]

wt[,sapply(wt,is.numeric)] <- scale(wt[,sapply(wt,is.numeric)])
head(wt)
```

Out[7]:

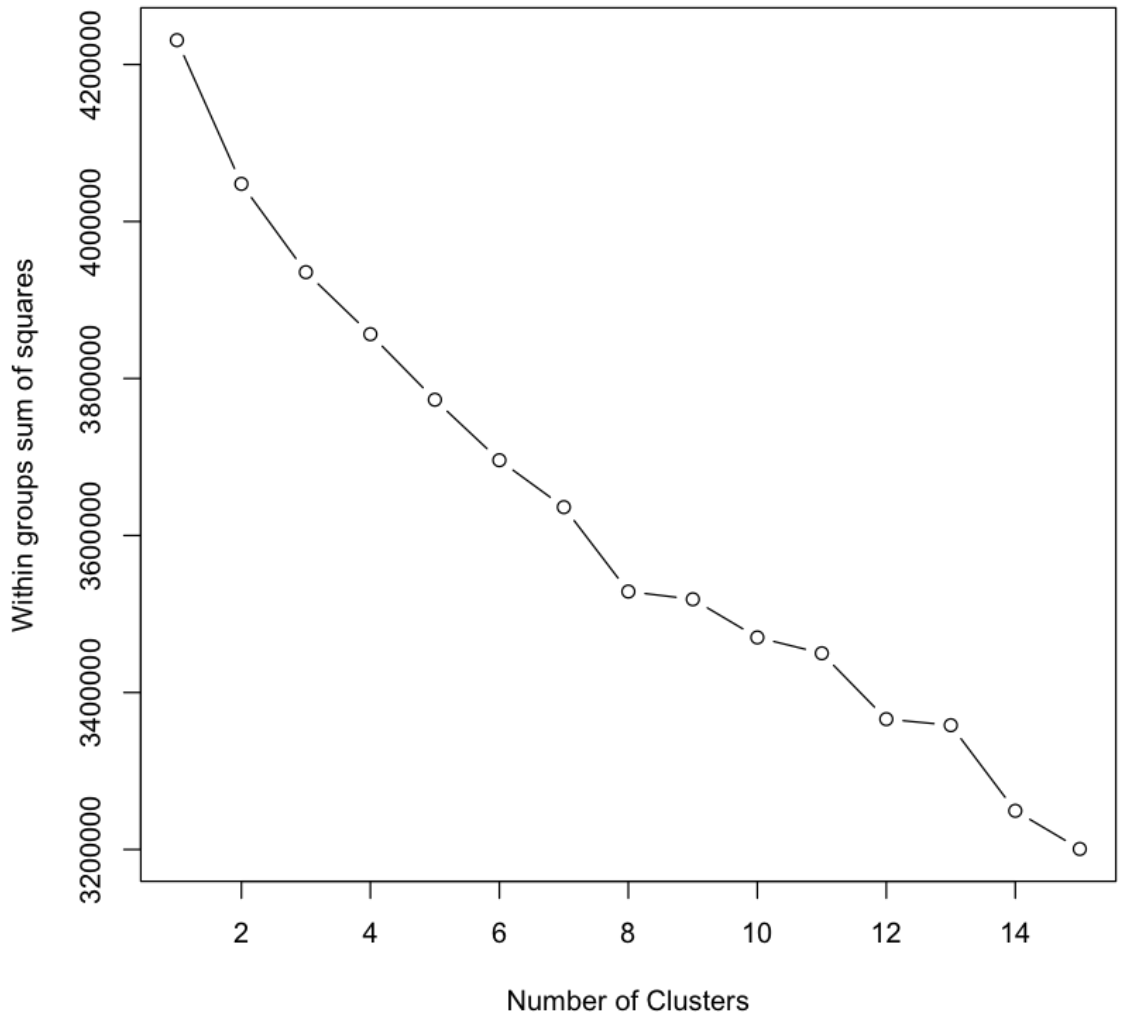
	TTLPY1X	FAMINCY1	POVLEVY1	WAGEPY1X	BUSNPY1X	UNEMPY1X	INTR
1	1.057738	0.4468941	0.5362892	1.241203	-0.0601531	-0.1266105	-0.10
2	0.2195184	0.4468941	0.5362892	0.3533358	-0.0601531	-0.1266105	-0.10
3	-0.6775425	0.4468941	0.5362892	-0.5968581	-0.0601531	-0.1266105	-0.10
4	0.07614411	-0.6287251	-0.5647706	0.2014695	-0.0601531	-0.1266105	-0.10
5	-0.6775425	-0.6287251	-0.5647706	-0.5968581	-0.0601531	-0.1266105	-0.10
7	0.3084123	0.06152826	-0.1326233	0.447495	-0.0601531	-0.1266105	-0.10

In [94]:
```
dim(wt)
```

Out[94]: 16274 261

In [8]:
```
wss <- (nrow(wt)-1)*sum(apply(wt,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(wt,centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
```

```
Warning message:
: Quick-TRANSfer stage steps exceeded maximum (= 813700)
```



In [9]:
```
wt <- cbind.data.frame(wt, Response)

#wt$Response <- as.numeric(wt$Response)
```

In [10]:
```
null1 <- glm(Response ~ 1, data = wt)

full1 <- glm(Response ~ ., data = wt)

both1 <- step(null1, scope = list(upper=full1),
              data=wt, direction="both", trace=F)

summary(both1)
```

Out[10]:

```
Call:
glm(formula = Response ~ RXTOTY1 + TOTEXPY1 + SSECPY1X + ERTMCRY1
+
    OPOTCHY1 + IPDMCDY1 + ERDTCHY1 + OBDMCRY1 + OBDWCPY1 + OBCPRVY
1 +
    OTHSLFY1 + ERTOSRY1 + PUBPY1X + AMTTCHY1 + IPFPTRY1 + TOTSLFY1
+
    RXSTLY1 + TOTOSRY1 + OPFSLFY1 + ERDMCDY1 + HHATCHY1 + HHAEXPY1
+
    VISEXPY1 + OTHTCHY1 + TRSTPY1X + FAMINCY1 + OBTOTVY1 + OBCSLFY
1 +
    OBOEXPY1 + AMNMCRY1 + OBTTCHY1 + OBVVAY1 + TOTWCPY1 + OPPEXPY1
+
    OPOPRVY1 + SSIPY1X + TOTOPUY1 + IPDMCRY1 + IPFMCRY1 + DVOTCHY1
+
    OBTPTRY1 + AMTPRVY1 + OPVMCDY1 + OPSPTRY1 + TOTVAY1 + OPDEXPY1
+
    OPTPTRY1 + OPTPRVY1 + OPOPTRY1 + OBVTRIY1 + OTHEXPY1 + OBDOPRY
1 +
    OBVOPRY1 + DIVDPY1X + OPFTCHY1 + OBCTCHY1 + OBCEXPY1 + OBNSLFY
1 +
    AMNEXPY1, data = wt)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.84502  -0.04534  -0.02959  -0.02620   1.06048

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.052476   0.001681  31.221  < 2e-16 ***
RXTOTY1      0.020652   0.002109   9.792  < 2e-16 ***
TOTEXPY1     0.023608   0.004070   5.801 6.71e-09 ***
SSECPY1X     0.016317   0.001873   8.711  < 2e-16 ***
ERTMCRY1     0.007759   0.001784   4.350 1.37e-05 ***
OPOTCHY1     0.017696   0.002994   5.909 3.50e-09 ***
IPDMCDY1     0.007806   0.001859   4.200 2.68e-05 ***
ERDTCHY1     0.007847   0.001810   4.337 1.46e-05 ***
OBDMCRY1     0.006705   0.002009   3.338 0.000845 ***
OBDWCPY1     0.011775   0.002756   4.273 1.94e-05 ***
OBCPRVY1     0.006451   0.002401   2.687 0.007218 **
OTHSLFY1     0.009651   0.002180   4.427 9.60e-06 ***
ERTOSRY1     0.007789   0.001825   4.268 1.99e-05 ***
PUBPY1X      0.005489   0.001691   3.246 0.001175 **
AMTTCHY1    -0.010357   0.003169  -3.268 0.001084 **
IPFPTRY1    -0.007661   0.002867  -2.672 0.007555 **
TOTSLFY1    -0.004655   0.002053  -2.268 0.023371 *
RXSTLY1     -0.004999   0.001705  -2.931 0.003379 **
TOTOSRY1    -0.005471   0.001864  -2.936 0.003335 **
OPFSLFY1    -0.007138   0.001898  -3.762 0.000169 ***
ERDMCDY1     0.004005   0.001801   2.223 0.026221 *
HHATCHY1     0.034629   0.008887   3.897 9.79e-05 ***
HHAEXPY1    -0.031820   0.008897  -3.576 0.000350 ***
VISEXPY1    -0.003563   0.001700  -2.096 0.036122 *
OTHTCHY1    -0.015983   0.007225  -2.212 0.026956 *
```

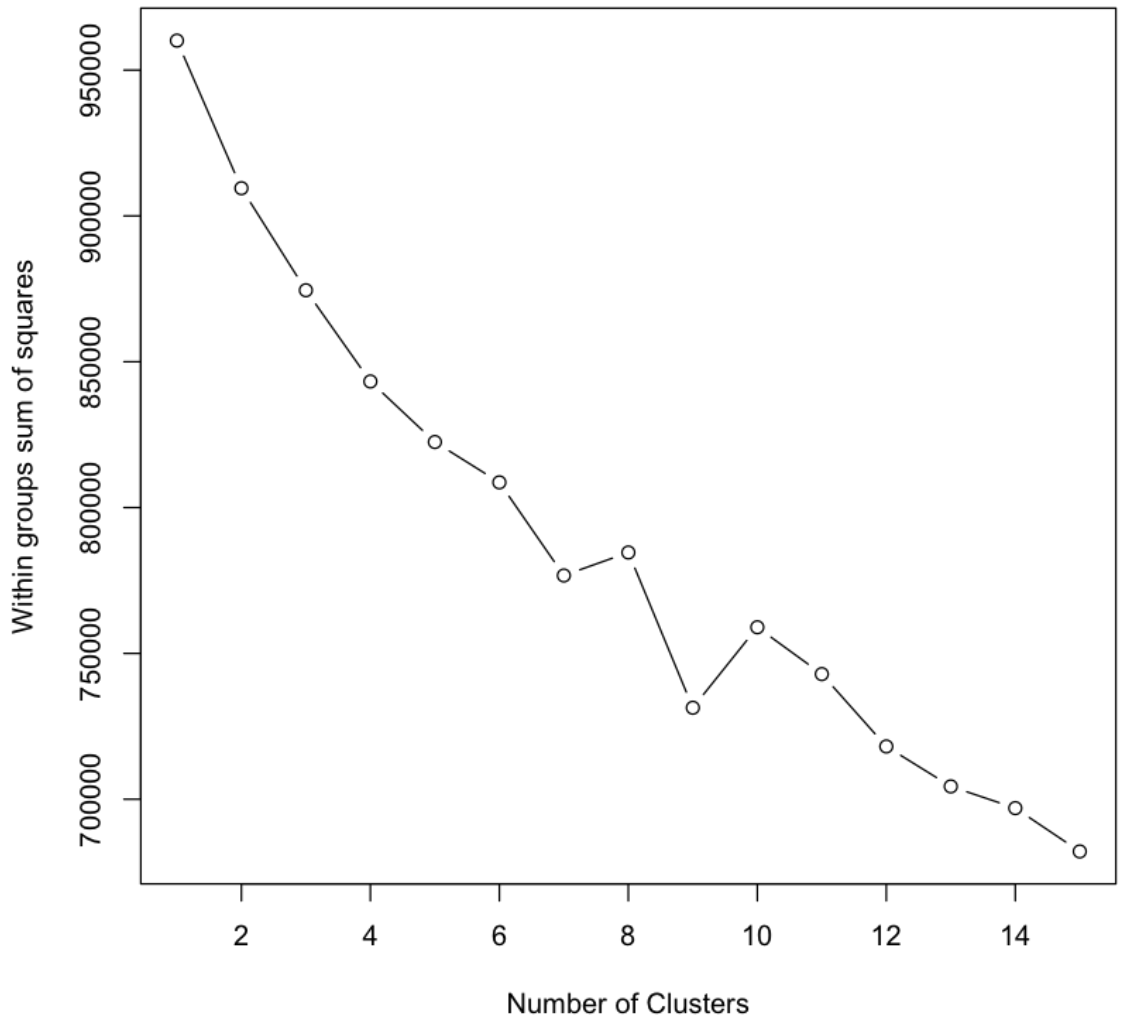
```
TRSTPY1X     0.004022   0.001712   2.350 0.018810 *
FAMINCY1    -0.003465   0.001751  -1.979 0.047812 *
OBTOTVY1     0.005643   0.002599   2.171 0.029910 *
OBCSLFY1    -0.003633   0.002310  -1.573 0.115760
OBOEXPY1    -0.008426   0.003062  -2.752 0.005930 **
AMNMCRY1     0.009175   0.002722   3.371 0.000751 ***
OBTTCHY1     0.010785   0.003410   3.163 0.001567 **
OBVVAY1      0.004875   0.001877   2.597 0.009417 **
TOTWCPY1    -0.006817   0.002807  -2.429 0.015163 *
OPPEXPY1    -0.013776   0.002824  -4.878 1.08e-06 ***
OPOPRVY1     0.056365   0.015185   3.712 0.000206 ***
SSIPY1X      0.003018   0.001731   1.743 0.081281 .
TOTOPUY1    -0.003005   0.001724  -1.743 0.081356 .
IPDMCRY1    -0.005602   0.002290  -2.447 0.014431 *
IPFMCRY1     0.004424   0.002573   1.719 0.085599 .
DVOTCHY1    -0.002768   0.001725  -1.604 0.108689
OBTPTRY1    -0.015503   0.007496  -2.068 0.038636 *
AMTPRVY1     0.010960   0.007212   1.520 0.128612
OPVMCDY1    -0.002596   0.001823  -1.424 0.154390
OPSPTRY1    -0.014433   0.003965  -3.640 0.000273 ***
TOTVAY1     -0.003796   0.001893  -2.005 0.045000 *
OPDEXPY1     0.020047   0.004524   4.432 9.42e-06 ***
OPTPTRY1     0.254546   0.040759   6.245 4.34e-10 ***
OPTPRVY1    -0.247110   0.040470  -6.106 1.04e-09 ***
OPOPTRY1    -0.055681   0.015302  -3.639 0.000275 ***
OBVTRIY1    -0.003979   0.001780  -2.235 0.025423 *
OTHEXPY1     0.012295   0.007252   1.695 0.090025 .
OBDOPRY1    -0.012770   0.003302  -3.868 0.000110 ***
OBVOPRY1     0.014351   0.003624   3.960 7.51e-05 ***
DIVDPY1X    -0.002896   0.001723  -1.681 0.092787 .
OPFTCHY1    -0.006878   0.002719  -2.529 0.011438 *
OBCTCHY1     0.013739   0.006344   2.166 0.030333 *
OBCEXPY1    -0.011286   0.006697  -1.685 0.091966 .
OBNSLFY1     0.003279   0.001856   1.767 0.077301 .
AMNEXPY1    -0.005245   0.003343  -1.569 0.116695
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.0459765)

    Null deviance: 809.19  on 16273  degrees of freedom
Residual deviance: 745.46  on 16214  degrees of freedom
AIC: -3872.3

Number of Fisher Scoring iterations: 2
```

In [95]:
```
selected = attr(terms(formula(both1)), "term.labels")
length(selected)
wt_step = wt[selected]
dim(wt_step)
```

Out[95]: 59

Out[95]: 16274 59

In [12]:
```
wss <- (nrow(wt_step)-1)*sum(apply(wt_step,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(wt_step,centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
```



In [13]:
```
k1 <- kmeans(wt_step, 7, iter.max = 500, nstart = 25)
```

In [31]:
```
final <- cbind.data.frame(ClustID = k1$clust, wt_step, Response)
head(final)
```

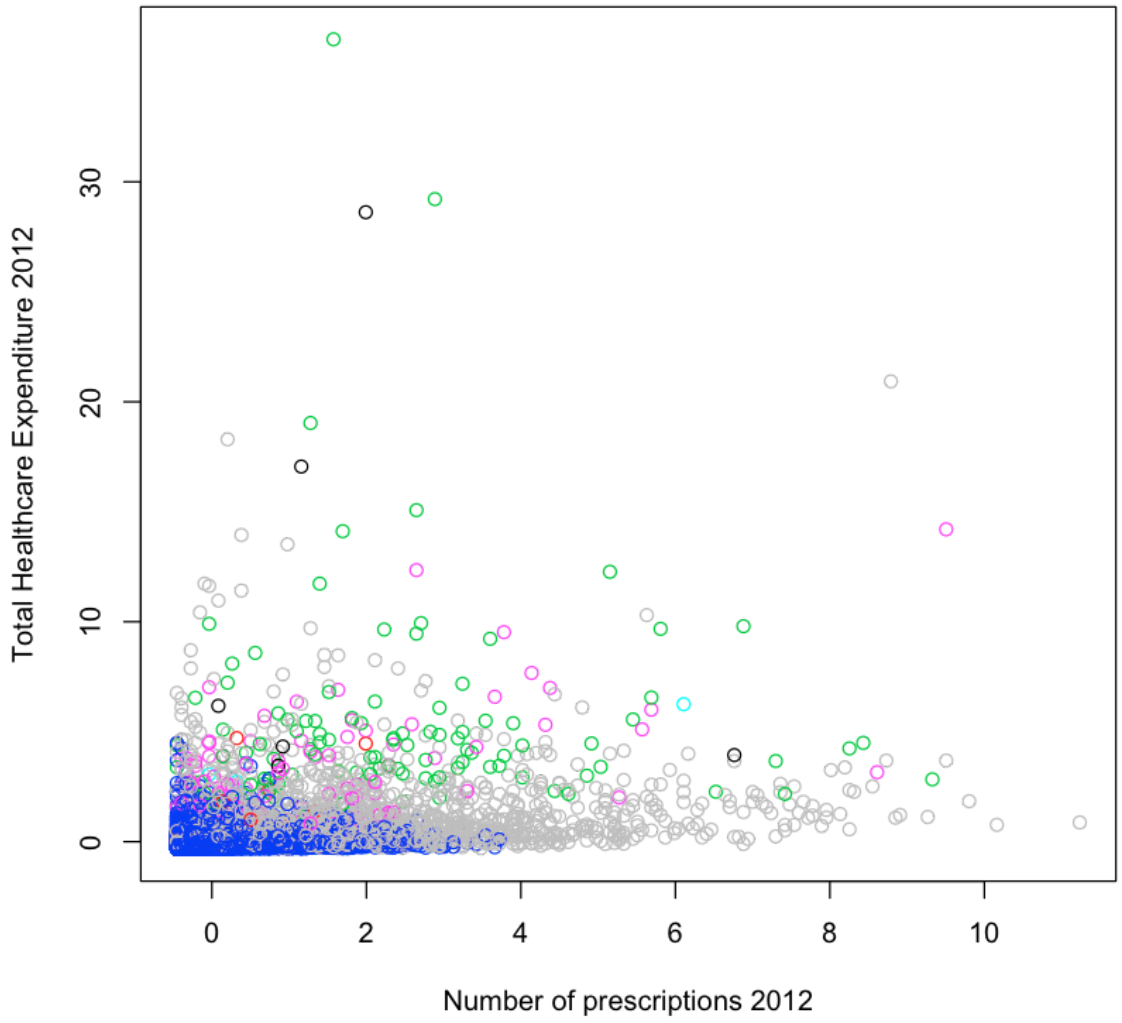
Out[31]:

	ClustID	RXTOTY1	TOTEXPY1	SSECPY1X	ERTMCRY1	OPOTCHY1	IPDMCDY
1	5	-0.452942	-0.3232491	-0.2890601	-0.07032514	-0.06893971	-0.078193
2	5	-0.452942	-0.3232491	-0.2890601	-0.07032514	-0.06893971	-0.078193
3	5	-0.452942	-0.3146469	-0.2890601	-0.07032514	-0.06893971	-0.078193
4	5	1.634197	-0.2378013	-0.2890601	-0.07032514	-0.06893971	-0.078193
5	5	-0.452942	-0.3086828	-0.2890601	-0.07032514	-0.06893971	-0.078193
7	5	-0.452942	-0.3232491	-0.2890601	-0.07032514	-0.06893971	-0.078193

In [73]:
```
dim(final)
```

Out[73]: 16274 61

In [196]:
```
cols <- as.numeric(levels(final$ClustID))[final$ClustID]

plot(final[,'RXTOTY1'], final[,'TOTEXPY1'], col=cols+23,

     xlab = 'Number of prescriptions 2012', ylab = 'Total Healthcar
e Expenditure 2012')
```



In [15]:
```
table(final$ClustID)
```

Out[15]:
```
    1     2     3     4     5     6     7
 1856     6    63   105 14123     3   118
```

In [31]:
```
table(final$ClustID)
```

Out[31]:
```
    1     2     3     4     5     6     7     8
13953   118    64  1773    99   258     3     6
```

In [37]:
```
table(final$ClustID)
```

Out[37]:
```
    1     2     3     4     5     6     7     8     9 
  563    63 13751     6   100   118  1545     3   125 
```

In [43]:
```
table(final$ClustID)
```

Out[43]:
```
    1     2     3     4     5     6     7     8     9    10 
   10   118     6   164    64 11650     3    75  1432  2752 
```

In [40]:
```
table(final$ClustID)
```

Out[40]:
```
    1     2     3     4     5     6     7     8     9    10    11 
12    13 
 1253    79    86    28    14   152     3   128    61     6  2884 
71 10976 
   14    15 
  457    76 
```

In [16]:
```
table(final$ClustID, final$Response)
```

Out[16]:
```
     0     1
1 1567   289
2    4     2
3   55     8
4   73    32
5 13625   498
6    3     0
7   93    25
```

In [32]:
```
final$Response <- as.factor(final$Response)
levels(final$Response) <- c("NotAdmittedY2", "AdmittedY2")
final$Response <- relevel(final$Response, "NotAdmittedY2")
```

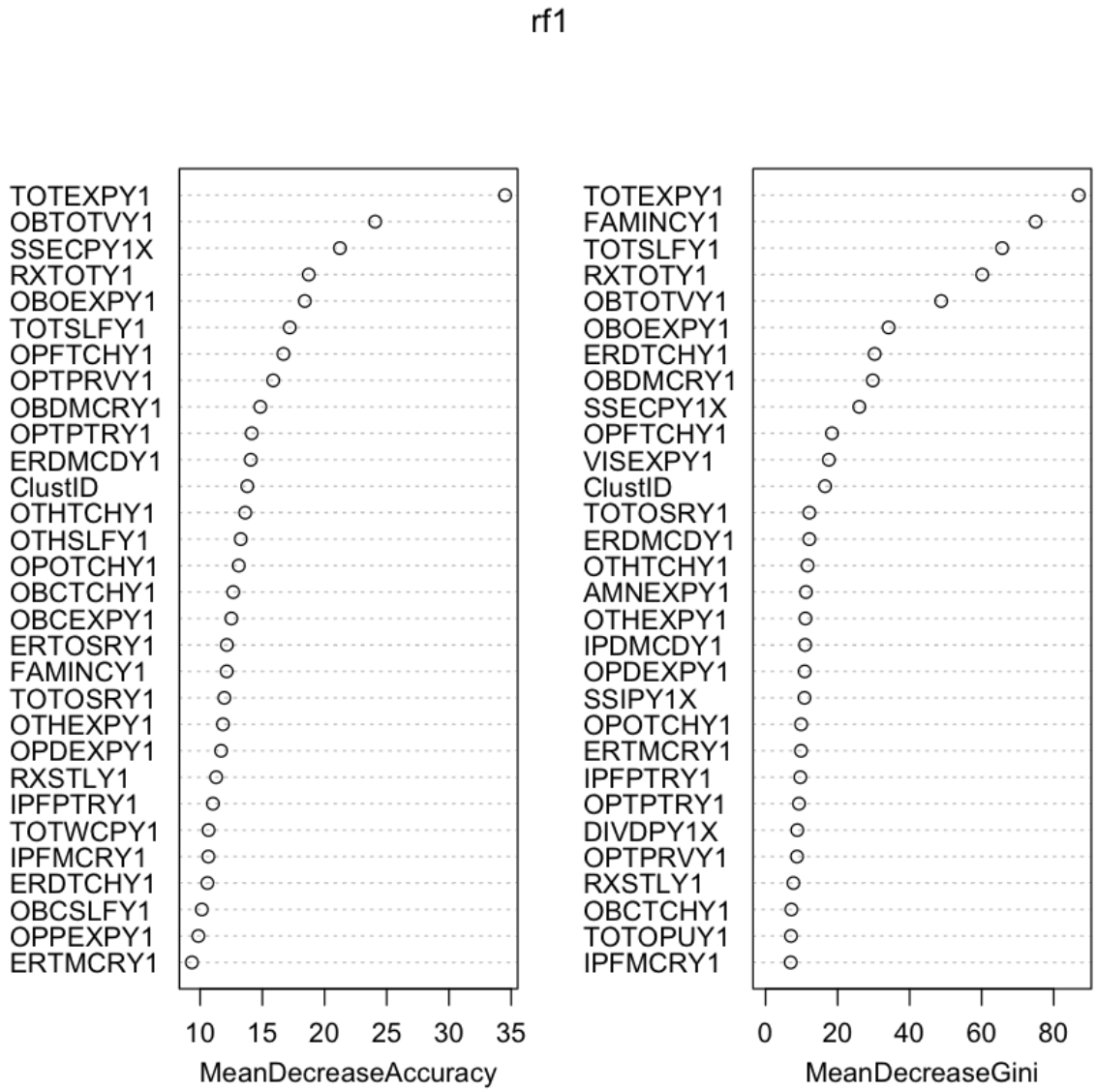
In [33]:
```
final$ClustID <- as.factor(final$ClustID)
train <- sample(1:nrow(final), 2*nrow(final)/3)
test <- final[-train, ]
train <- final[train, ]
```

In [37]:
```
table(train$Response)
```

Out[37]:
```
NotAdmittedY2    AdmittedY2 
        10274           575 
```

In [34]:
```
rf1 <- randomForest(Response ~ ., data = train,
                    mtry = floor(sqrt(ncol(train))), ntree = 1001,
                    do.trace = 100, importance = T)
varImpPlot(rf1)
```

```
ntree      OOB      1      2
  100:   5.33%  0.11% 98.61%
  200:   5.32%  0.09% 98.78%
  300:   5.30%  0.06% 98.96%
  400:   5.30%  0.05% 99.13%
  500:   5.31%  0.06% 99.13%
  600:   5.31%  0.06% 99.13%
  700:   5.31%  0.06% 99.13%
  800:   5.32%  0.06% 99.30%
  900:   5.31%  0.06% 99.13%
 1000:   5.30%  0.06% 98.96%
```

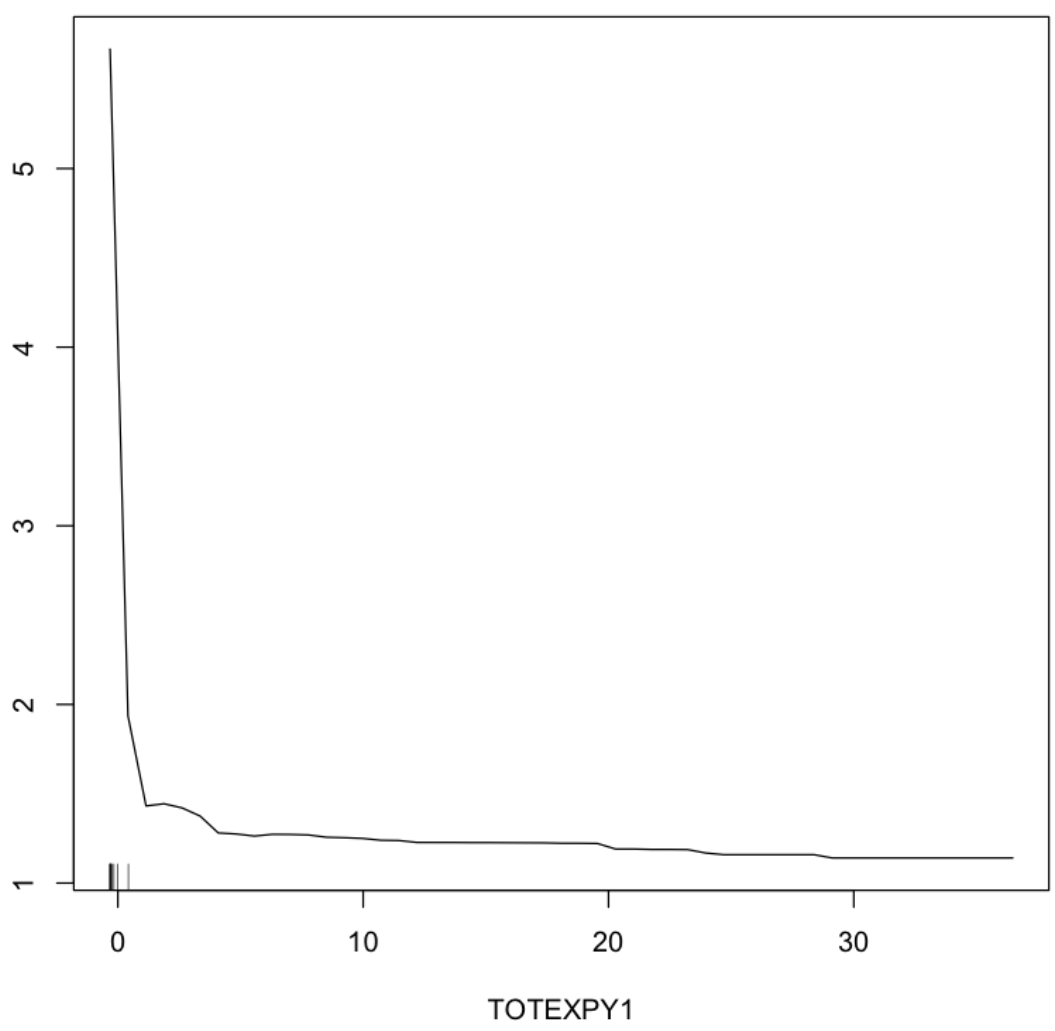


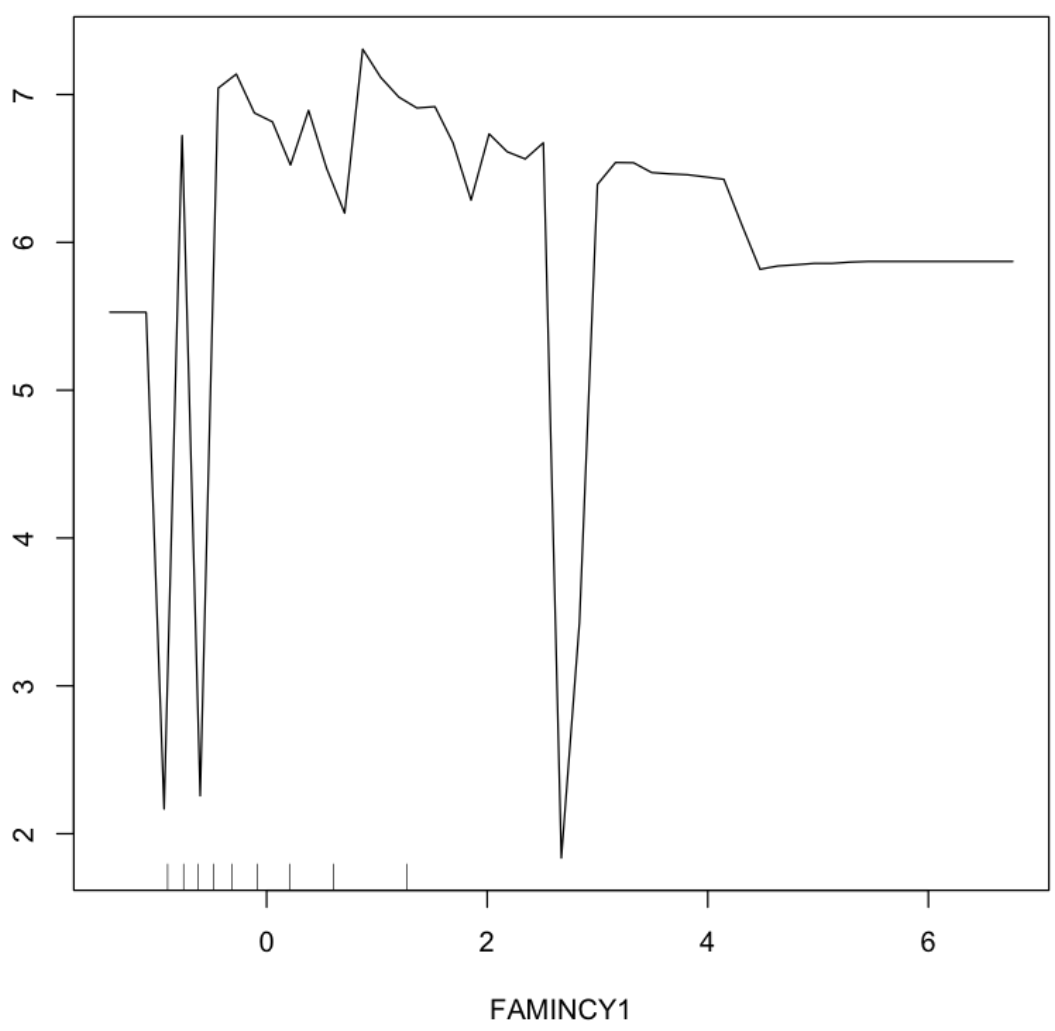
In [130]:
```
importanceOrder=order(-rf1$importance[,4])
names=names(rf1$importance[,4])[importanceOrder][1:15]
par(mfrow=c(5, 3), xpd=NA)
```

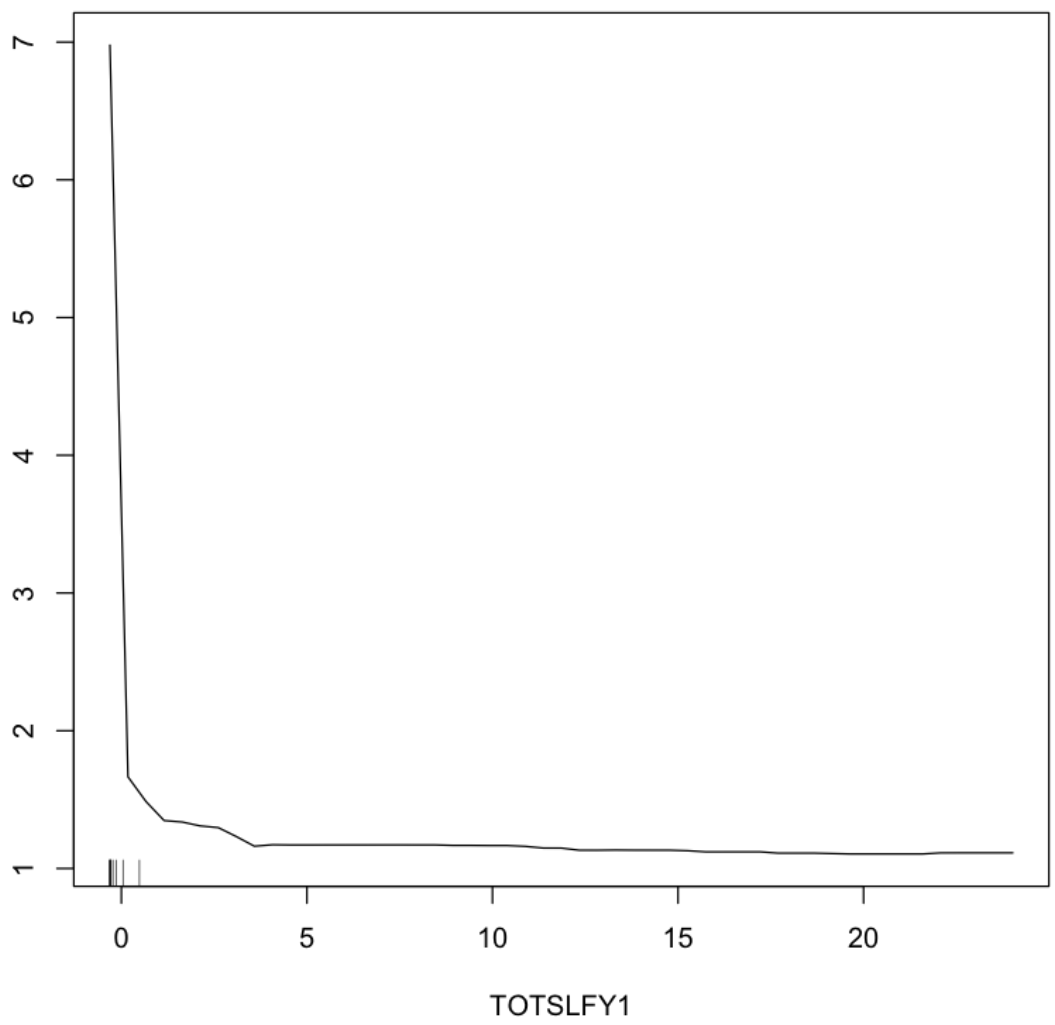
In [134]:
```
names
```

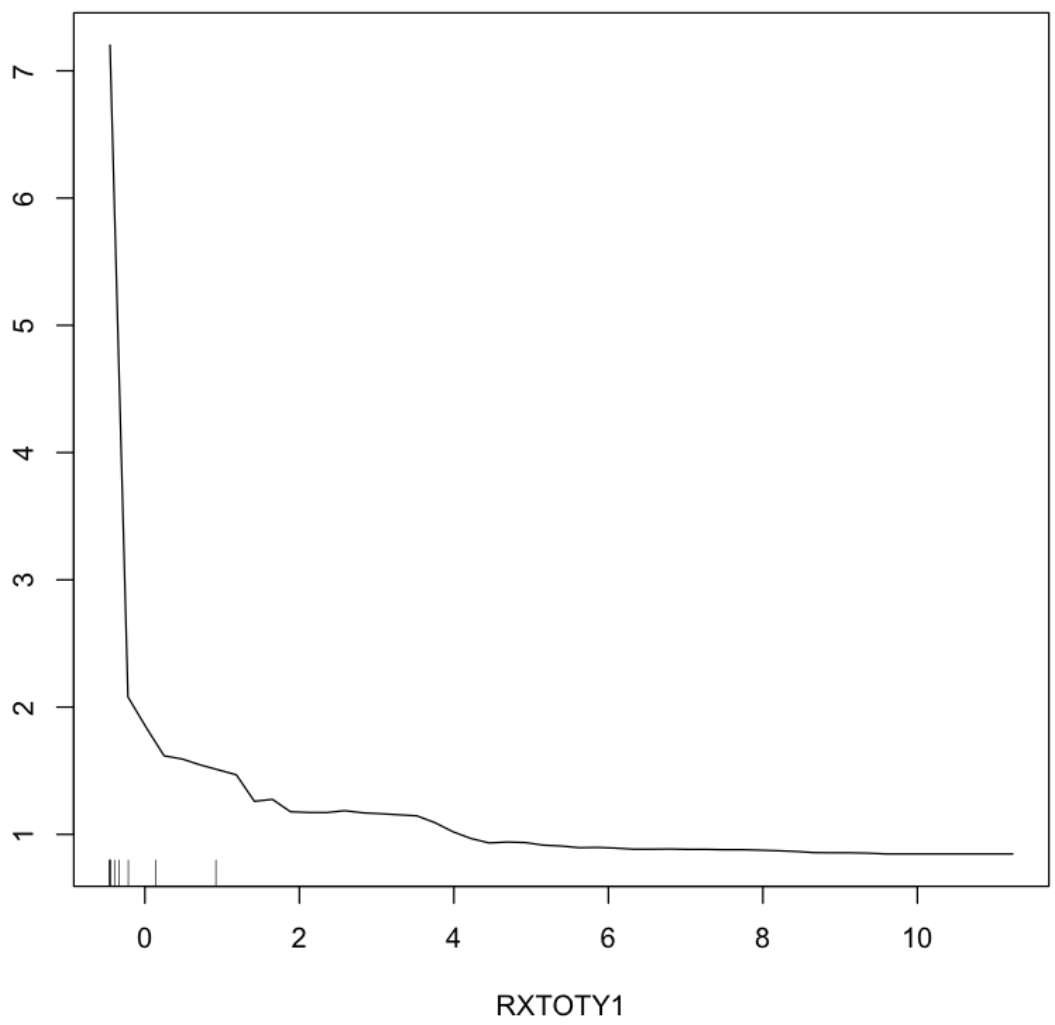
Out[134]:
'TOTEXPY1' 'FAMINCY1' 'TOTSLFY1' 'RXTOTY1' 'OBTOTVY1'
'OBOEXPY1' 'ERDTCHY1' 'OBDMCRY1' 'SSECPY1X' 'OPFTCHY1'
'VISEXPY1' 'ClustID' 'TOTOSRY1' 'ERDMCDY1' 'OTHTCHY1'

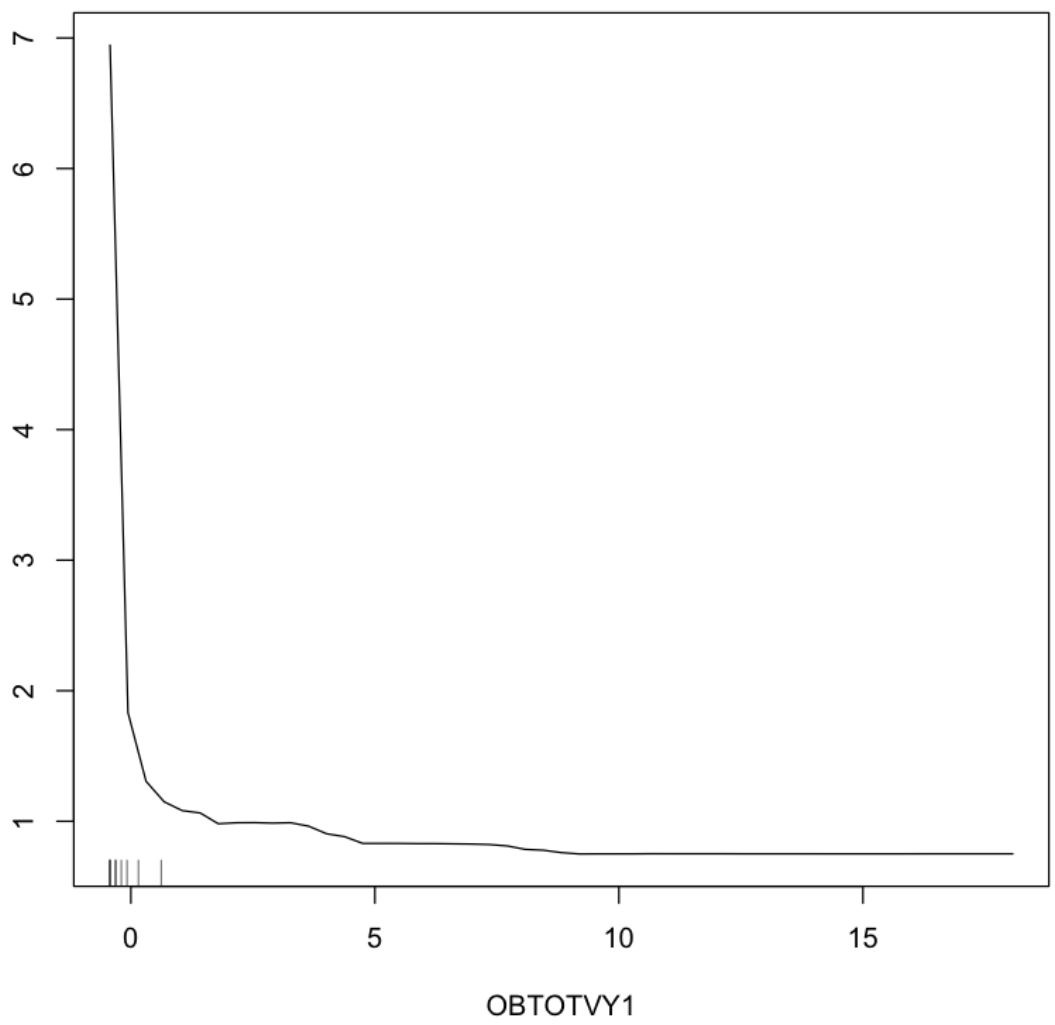
In [131]:
```
for (name in names){
    partialPlot(rf1, train, eval(name), main=name, xlab=name)
    }
```

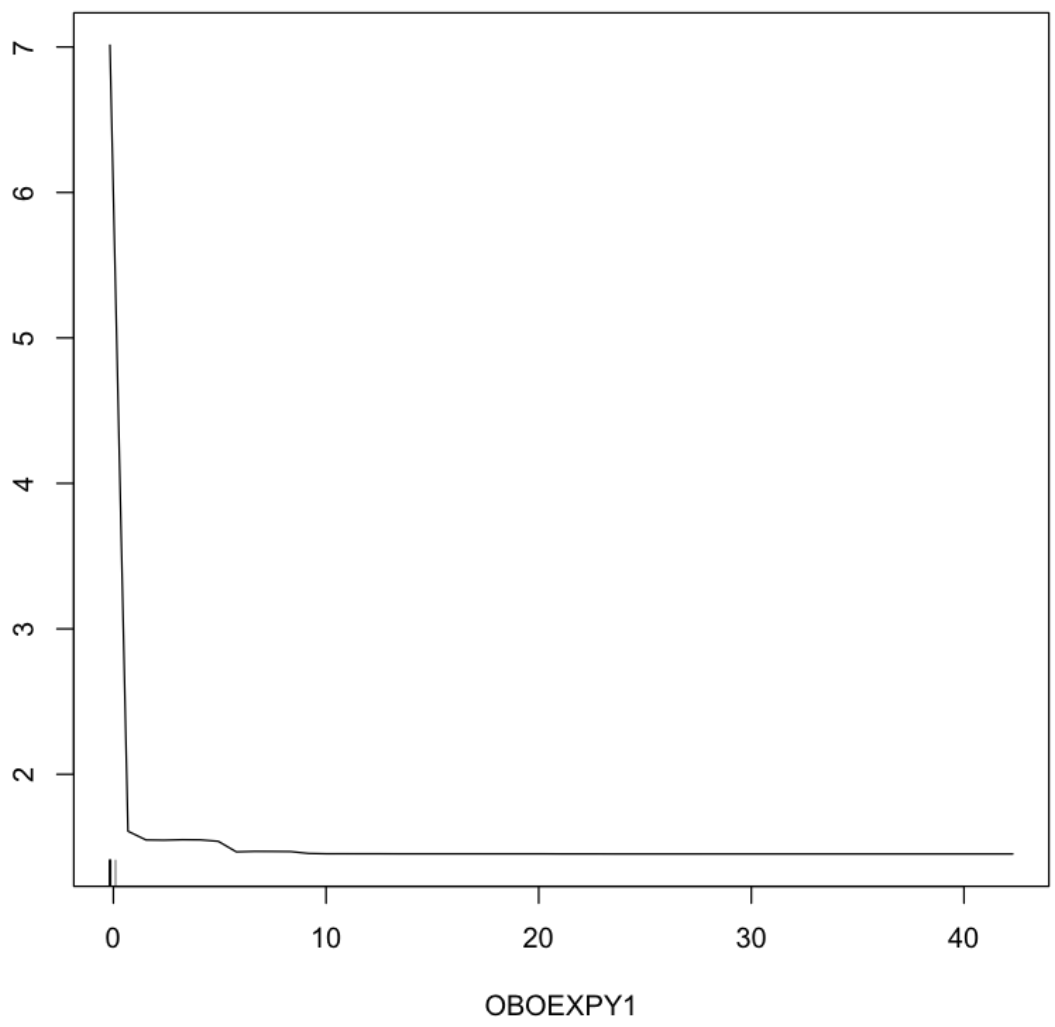

TOTEXPY1
1
2
3
4
5
0
10
20
30
TOTEXPY1

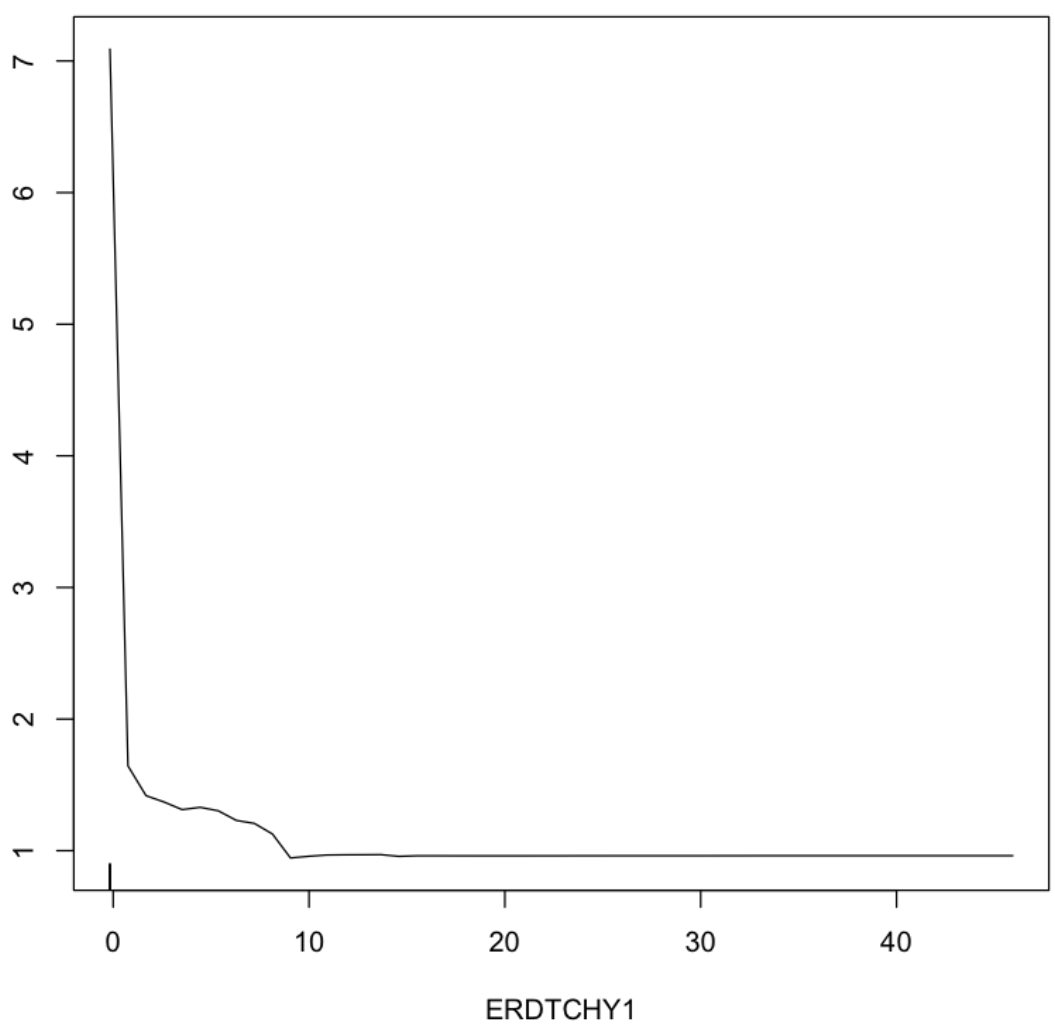

FAMINCY1
7
6
5
4
3
2
0
2
4
6
FAMINCY1

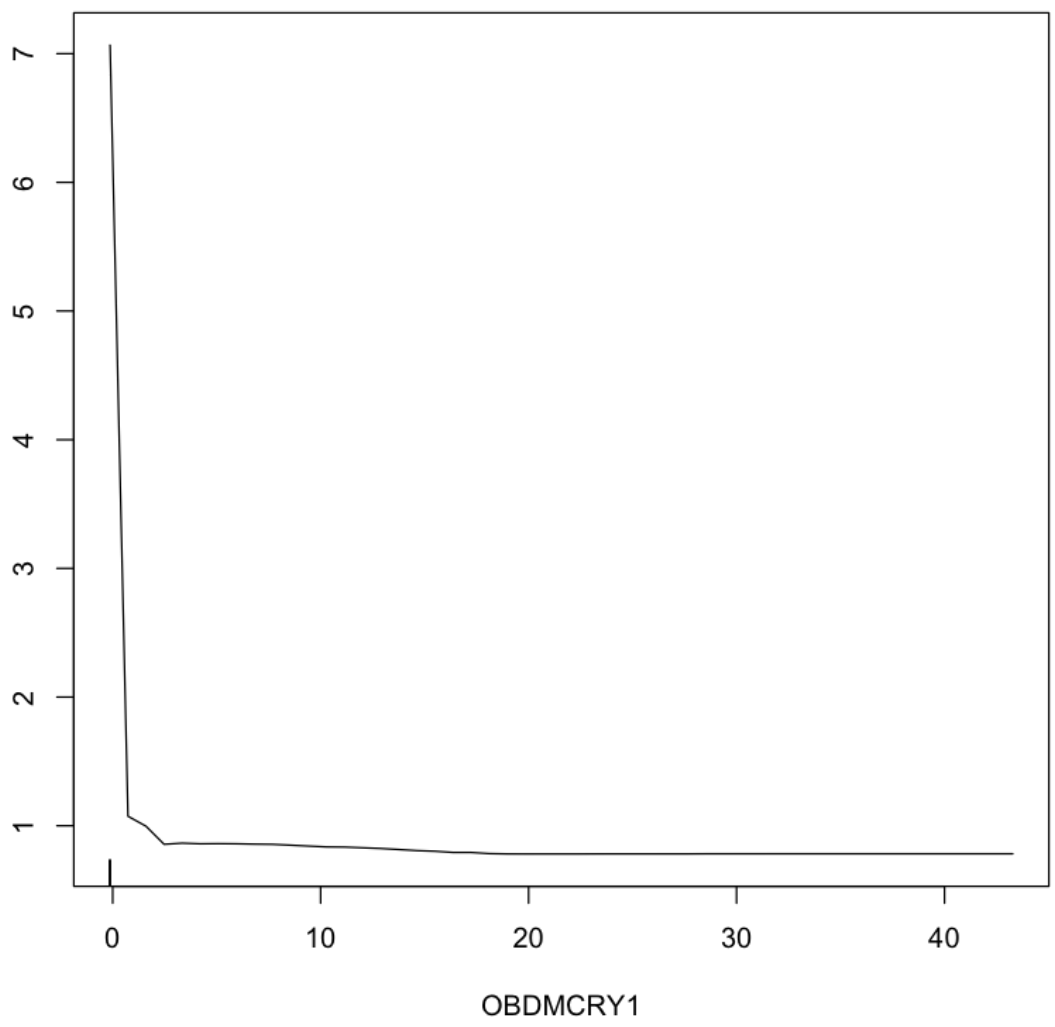

TOTSLFY1
1
2
3
4
5
6
7
0
5
10
15
20
TOTSLFY1

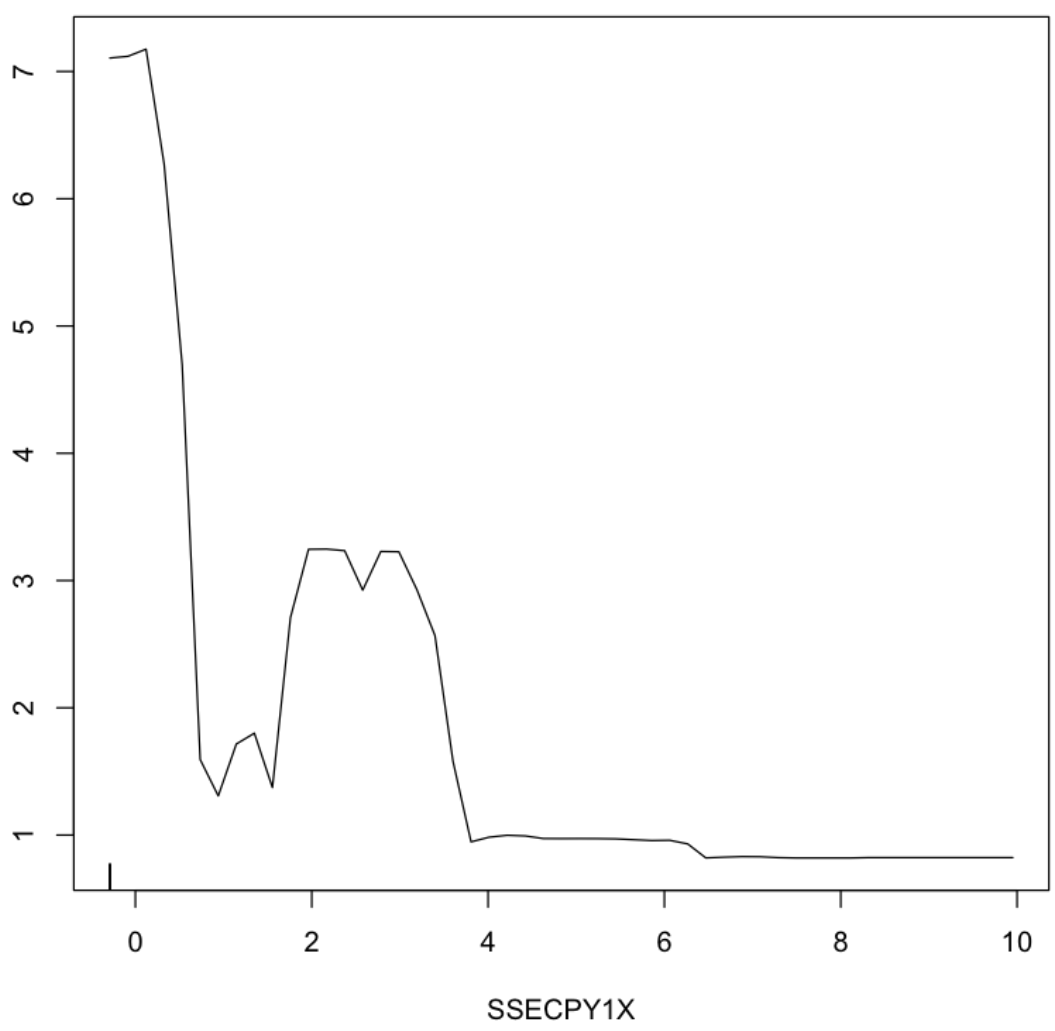

RXTOTY1
7
6
5
4
3
2
1
0
2
4
6
8
10
RXTOTY1

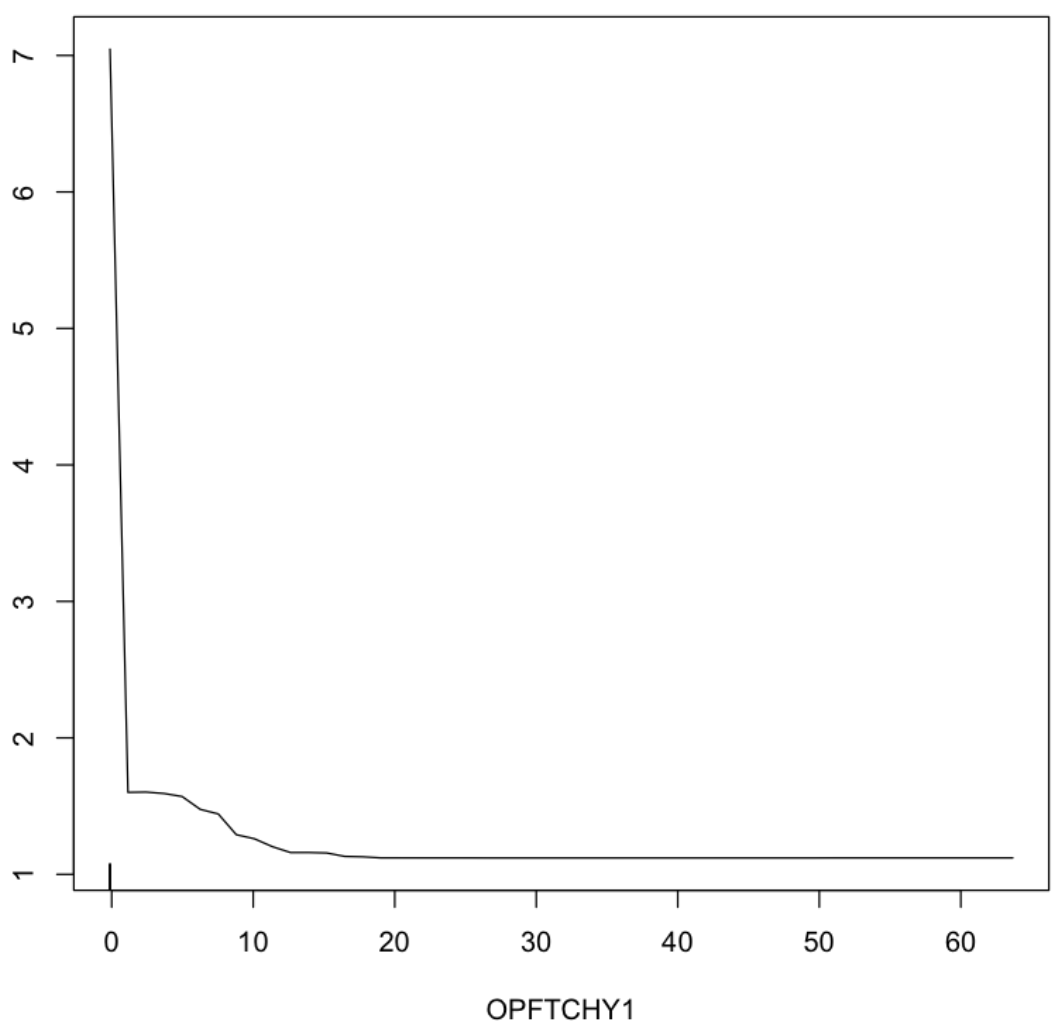

OBTOTVY1
7
6
5
4
3
2
1
0
5
10
15
OBTOTVY1

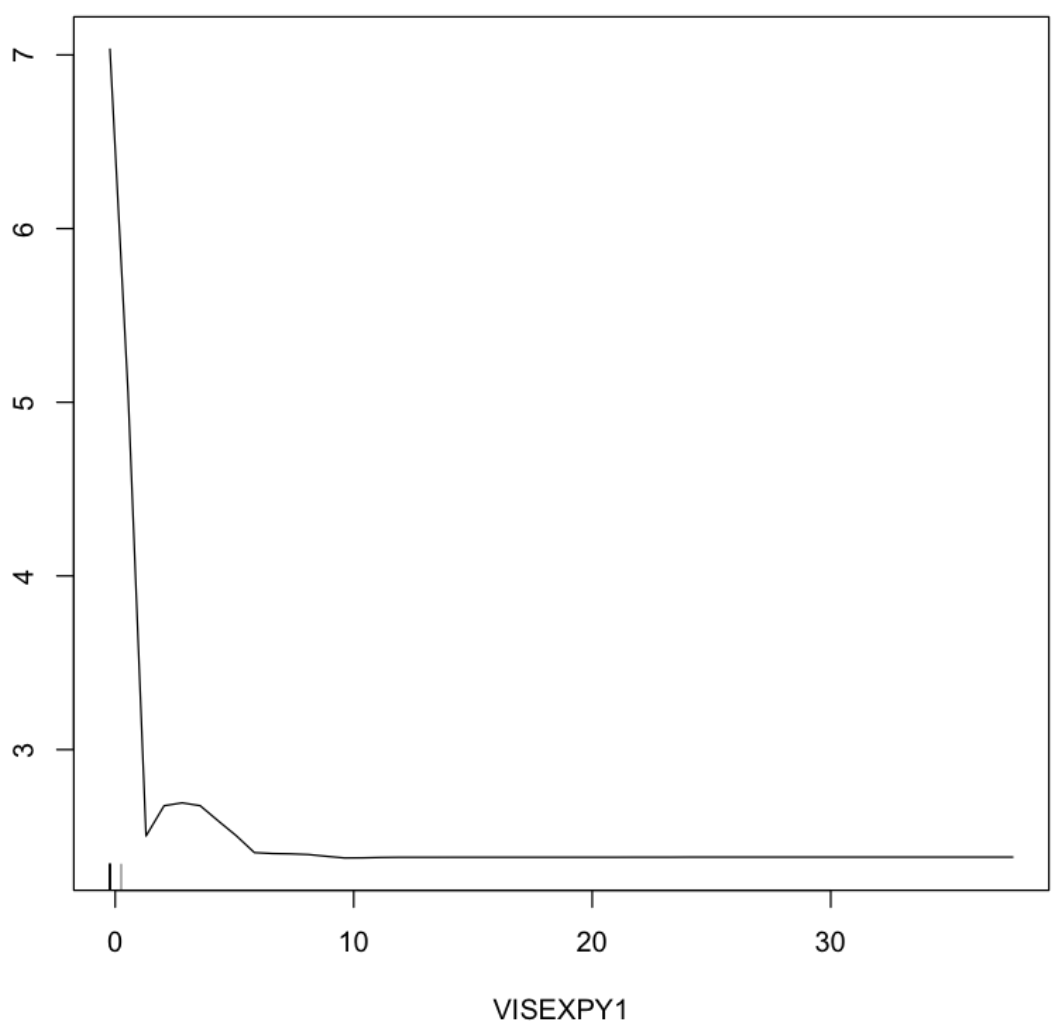

OBOEXPY1
7
6
5
4
3
2
0
10
20
30
40
OBOEXPY1

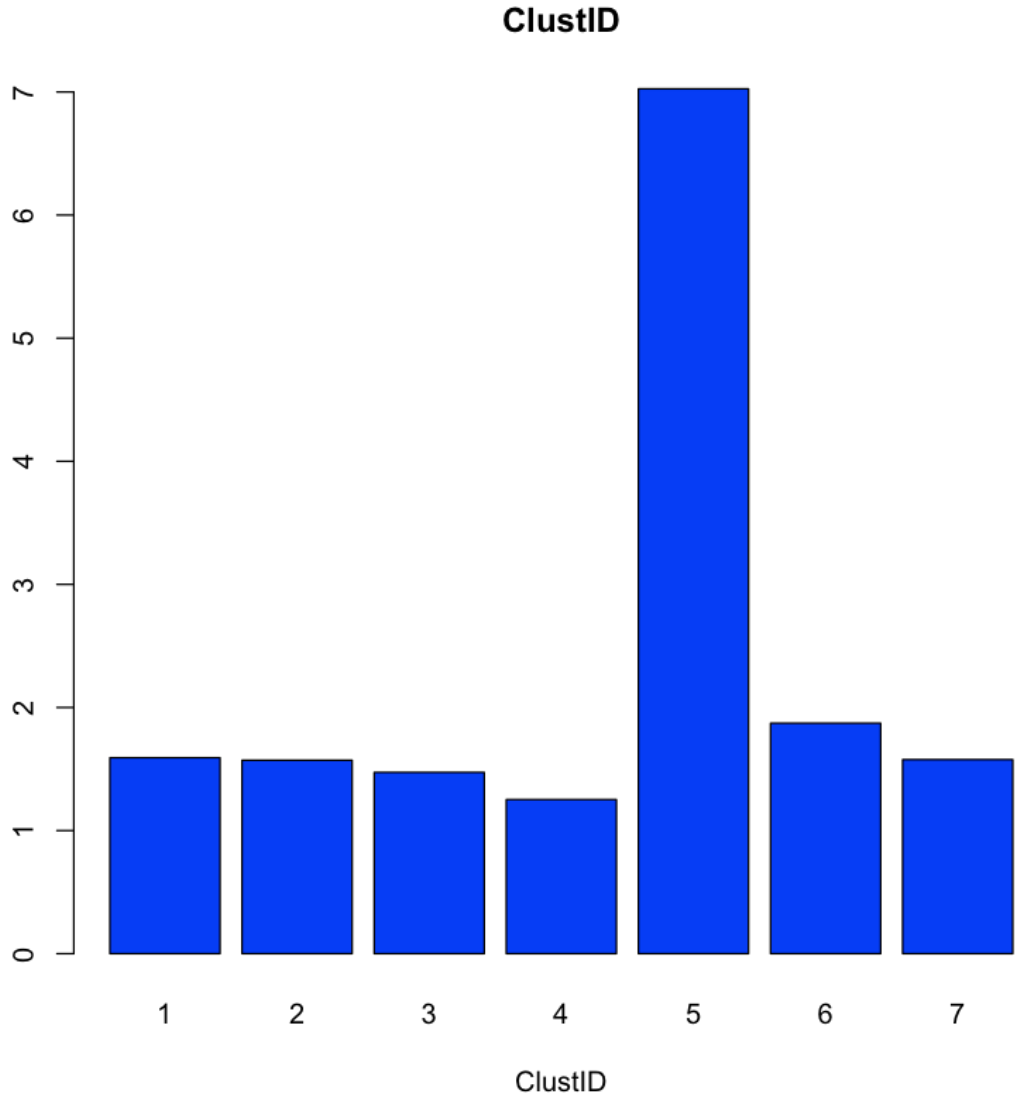

ERDTCHY1
1
2
3
4
5
6
7
0
10
20
30
40
ERDTCHY1

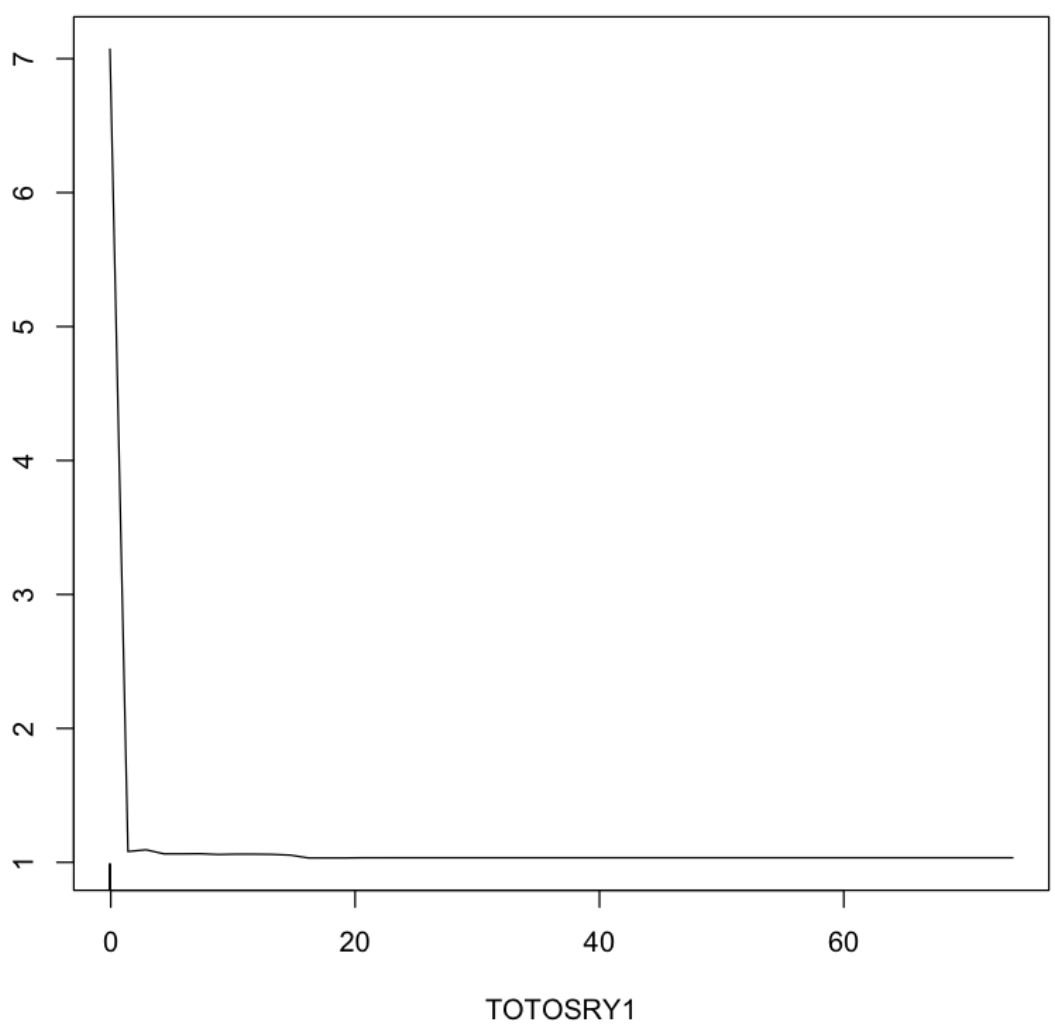

OBDMCRY1
7
6
5
4
3
2
1
0
10
20
30
40
OBDMCRY1

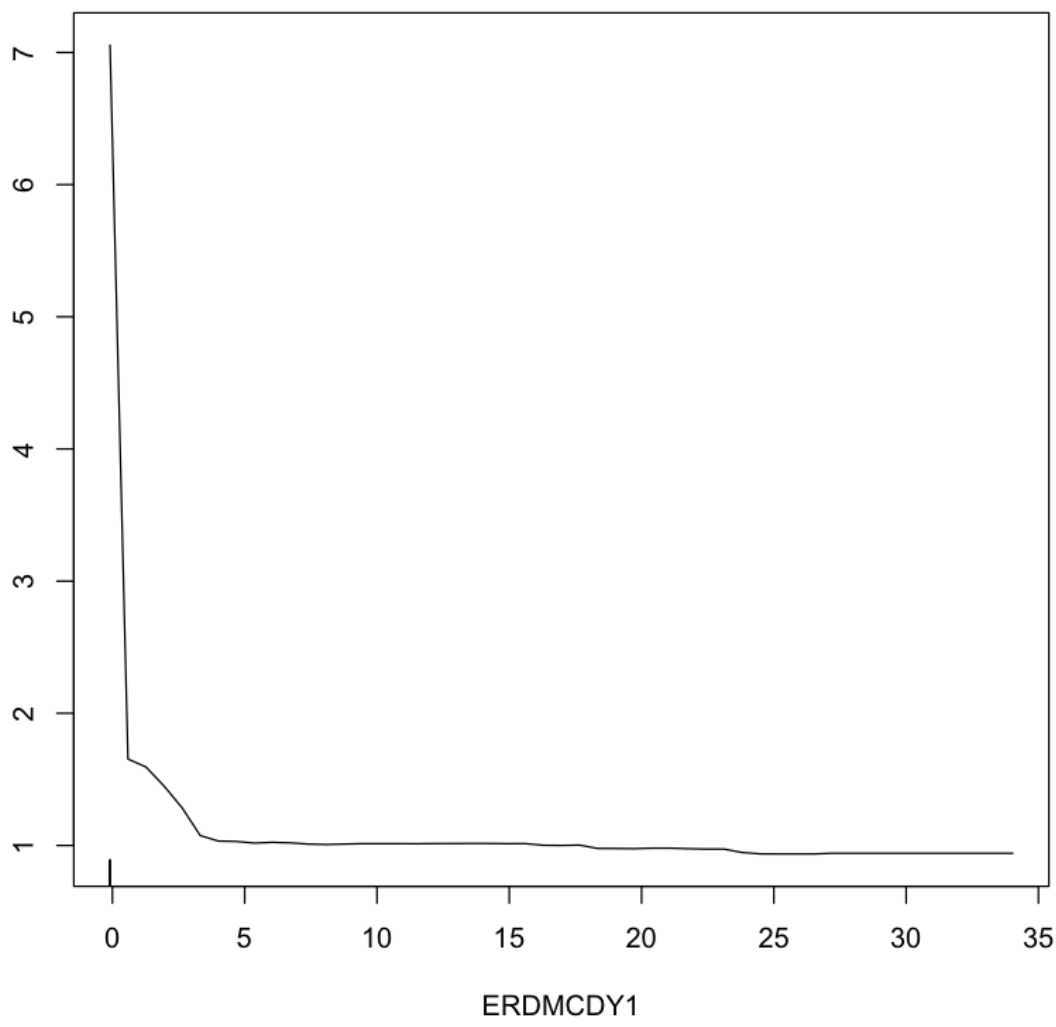

SSECPY1X
1
2
3
4
5
6
7
0
2
4
6
8
10
SSECPY1X

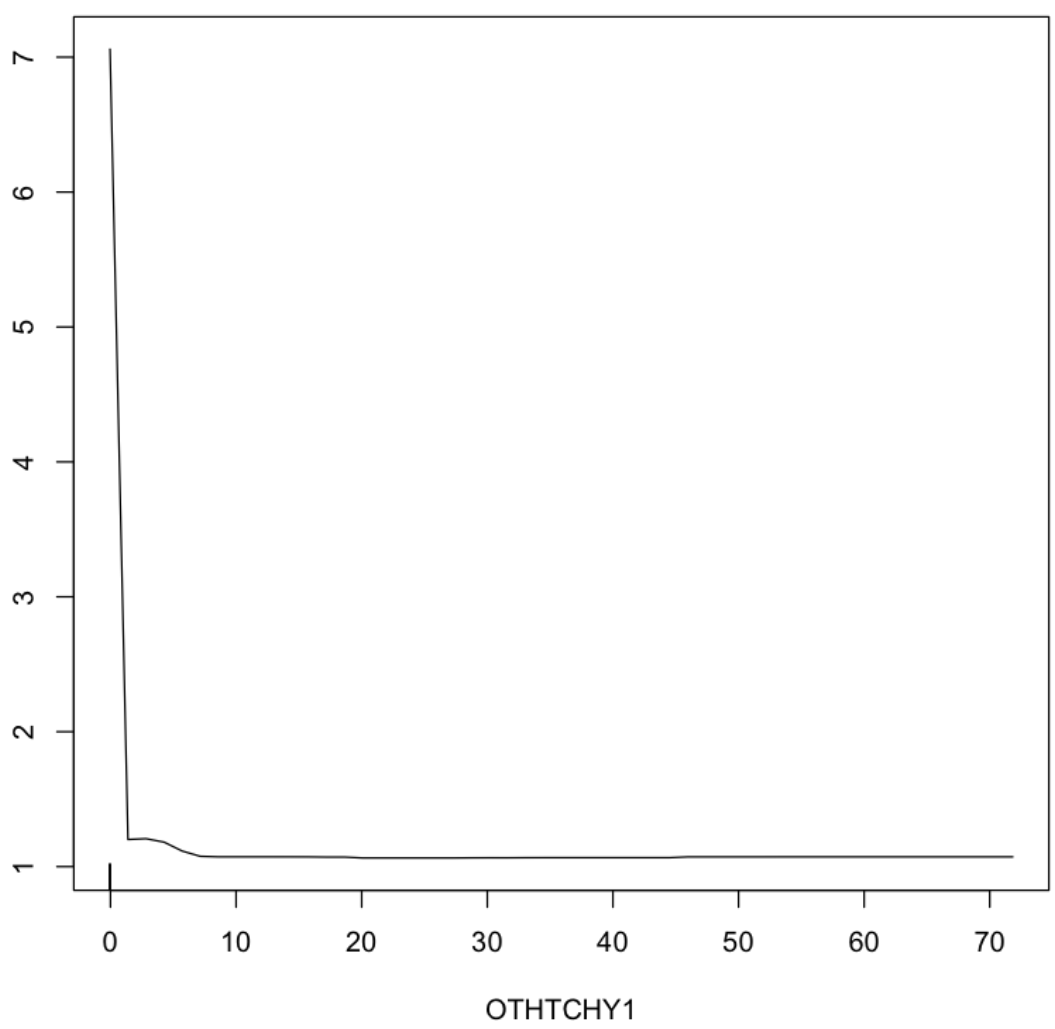

OPFTCHY1
7
6
5
4
3
2
1
0
10
20
30
40
50
60
OPFTCHY1


VISEXPY1
7
6
5
4
3
0
10
20
30
VISEXPY1


ClustID
0
1
2
3
4
5
6
7
1
2
3
4
5
6
7
ClustID


TOTOSRY1
7
6
5
4
3
2
1
0
20
40
60
TOTOSRY1


ERDMCDY1
1
2
3
4
5
6
7
0
5
10
15
20
25
30
35
ERDMCDY1


OTHTCHY1
7
6
5
4
3
2
1
0
10
20
30
40
50
60
70
OTHTCHY1

In [97]:
```
library(caret)
varImp(rf1)
```

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

Out[97]:

	NotAdmittedY2	AdmittedY2
ClustID	1.751955	1.751955
RXTOTY1	13.38969	13.38969
TOTEXPY1	5.081527	5.081527
SSECPY1X	13.63035	13.63035
ERTMCRY1	6.812783	6.812783
OPOTCHY1	4.09112	4.09112
IPDMCDY1	4.864932	4.864932
ERDTCHY1	8.918273	8.918273
OBDMCRY1	12.52303	12.52303
OBDWCPY1	4.335661	4.335661
OBCPRVY1	4.728115	4.728115
OTHSLFY1	4.358331	4.358331
ERTOSRY1	8.510591	8.510591
PUBPY1X	-1.00111	-1.00111
AMTTCHY1	-0.586257	-0.586257
IPFPTRY1	5.969305	5.969305
TOTSLFY1	-2.980633	-2.980633
RXSTLY1	5.523363	5.523363
TOTOSRY1	6.624293	6.624293
OPFSLFY1	2.883687	2.883687
ERDMCDY1	10.57081	10.57081
HHATCHY1	2.983612	2.983612
HHAEXPY1	2.650526	2.650526
VISEXPY1	1.853175	1.853175
OTHTCHY1	7.42618	7.42618
TRSTPY1X	0.6001318	0.6001318
FAMINCY1	6.580865	6.580865
OBTOTVY1	6.690174	6.690174
OBCSLFY1	2.318861	2.318861
OBOEXPY1	3.42409	3.42409

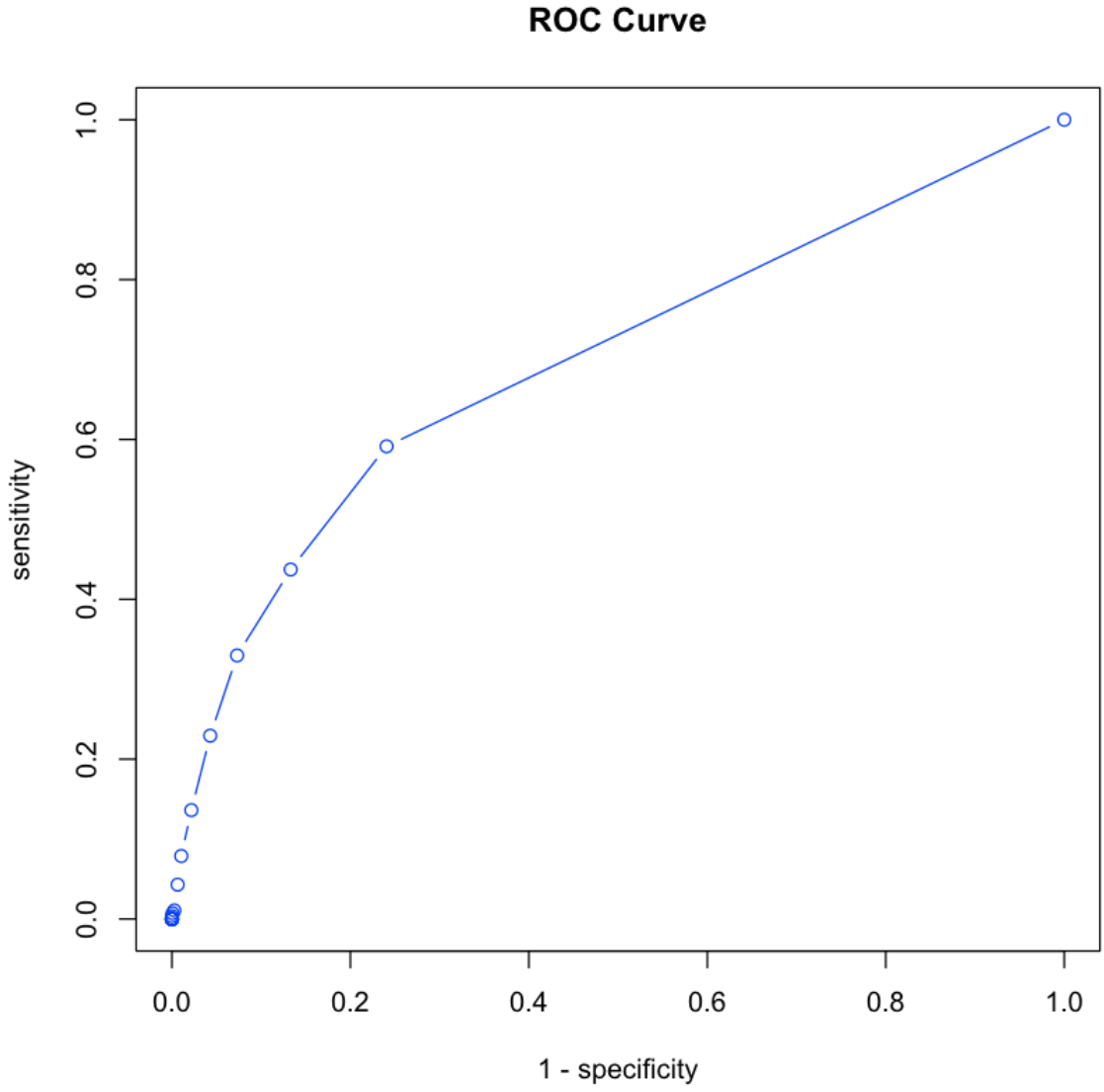
AMNMCRY1	0.9449618	0.9449618
OBTTCHY1	0.3447606	0.3447606
OBVVAY1	1.642447	1.642447
TOTWCPY1	5.043234	5.043234
OPPEXPY1	1.630925	1.630925
OPOPRVY1	1.556419	1.556419
SSIPY1X	3.675315	3.675315
TOTOPUY1	-3.176803	-3.176803
IPDMCRY1	3.822196	3.822196
IPFMCRY1	6.217809	6.217809
DVOTCHY1	2.790948	2.790948
OBTPTRY1	1.361655	1.361655
AMTPRVY1	1.202196	1.202196
OPVMCDY1	0.7097594	0.7097594
OPSPTRY1	1.490514	1.490514
TOTVAY1	3.602503	3.602503
OPDEXPY1	1.936316	1.936316
OPTPTRY1	3.344695	3.344695
OPTPRVY1	4.799932	4.799932
OPOPTRY1	1.571828	1.571828
OBVTRIY1	-1.149885	-1.149885
OTHEXPY1	8.076125	8.076125
OBDOPRY1	1.052252	1.052252
OBVOPRY1	1.837337	1.837337
DIVDPY1X	-1.082222	-1.082222
OPFTCHY1	4.542997	4.542997
OBCTCHY1	3.792725	3.792725
OBCEXPY1	2.979564	2.979564
OBNSLFY1	1.598339	1.598339
AMNEXPY1	1.888197	1.888197

In [39]:
```
SenSpec=function(model.name, dataset, cut.off.prob.seq){
  cut.off.prob.seq = as.vector(cut.off.prob.seq)
  np=length(cut.off.prob.seq)
  prob.good=predict(model.name, newdata=dataset, type="prob")
  ###### Here the appropriate response column should be specified
  true.class=dataset$Response
  perf.matrix=matrix(rep(0,2*np), byrow=TRUE, ncol=2)
  for(i in 1:np){
    pred.class=true.class
    ##### adding indexing for second column --> prob of Admitted
    pred.class[which(prob.good[,2] >= cut.off.prob.seq[i])] = "Admi
ttedY2"
    pred.class[which(prob.good[,2] < cut.off.prob.seq[i])] = "NotAd
mittedY2"
    confusion=ftable(pred.class, true.class)
    specificity=confusion[1,1]/sum(confusion[,1])
    sensitivity=confusion[2,2]/sum(confusion[,2])
    perf.matrix[i,1] = sensitivity
    perf.matrix[i,2] = specificity
  }
  colnames(perf.matrix)=c("sensitivity", "specificity")
  perf.matrix
}
```

In [40]:
```
SenSpec1 <- SenSpec(model.name=rf1, dataset=test,
                    cut.off.prob.seq=seq(0,1,length=20))
SenSpec1
TPF.1 = SenSpec1[,1]
FNF.1 = 1 - SenSpec1[,2]

##### ROC curve
plot(FNF.1, TPF.1, xlim=c(0,1), ylim=c(0,1), type="b", lty=1, col
="blue",
     xlab="1 - specificity", ylab="sensitivity", main="ROC Curve")
```

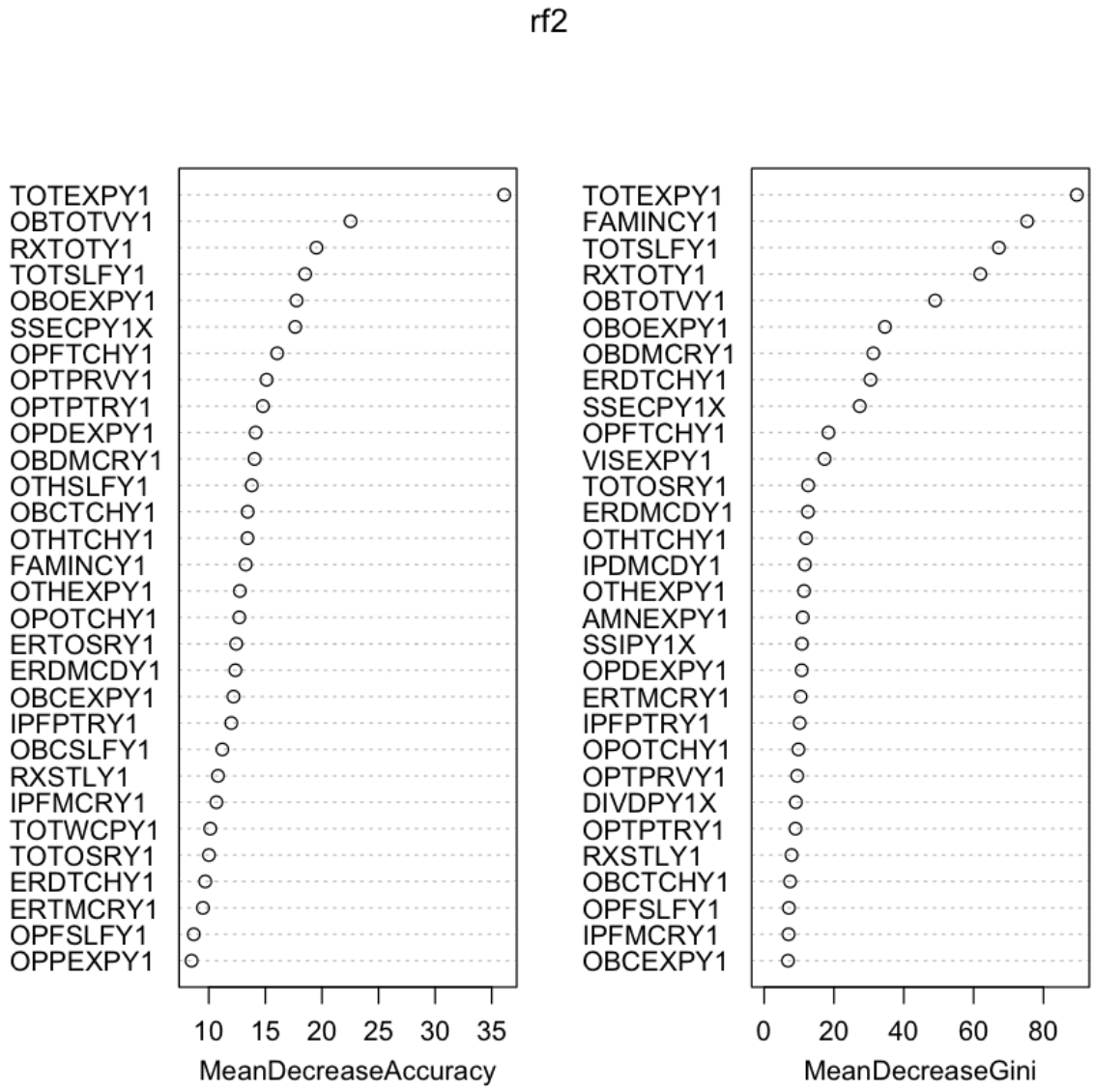
Out[40]:

sensitivity	specificity
1	0
0.5913978	0.7594248
0.4372760	0.8668869
0.3297491	0.9269335
0.2293907	0.9570540
0.1362007	0.9782355
0.07885305	0.98950641
0.04301075	0.99358725
0.01075269	0.99727944
0.007168459	0.999222697
0.003584229	1.000000000
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1


ROC Curve
sensitivity
1 - specificity
0.0
0.2
0.4
0.6
0.8
1.0

In [49]:
```
rf2 <- randomForest(Response ~ ., data = train[,-1],
                    mtry = floor(sqrt(ncol(train[,-1]))), ntree = 1
001,
                    do.trace = 100, importance = T)
varImpPlot(rf2)
```

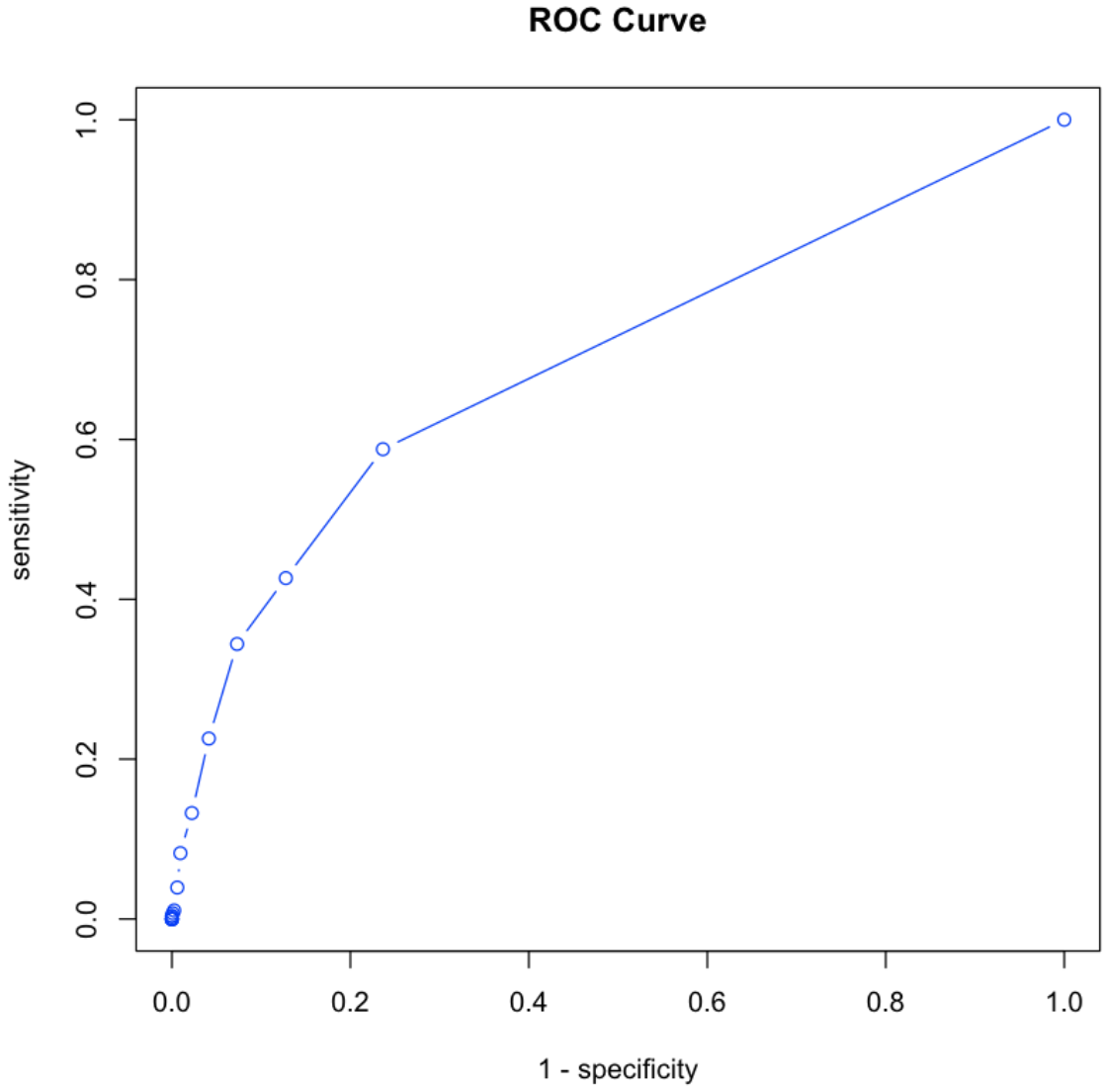
```
ntree      OOB      1      2
  100:   5.27%  0.07% 98.26%
  200:   5.29%  0.07% 98.61%
  300:   5.28%  0.05% 98.78%
  400:   5.24%  0.05% 98.09%
  500:   5.23%  0.04% 97.91%
  600:   5.25%  0.04% 98.43%
  700:   5.27%  0.05% 98.61%
  800:   5.27%  0.04% 98.78%
  900:   5.26%  0.03% 98.78%
 1000:   5.28%  0.05% 98.78%
```



In [51]:
```
SenSpec2 <- SenSpec(model.name=rf2, dataset=test[,-1],
                    cut.off.prob.seq=seq(0,1,length=20))
SenSpec1
TPF.2 = SenSpec2[,1]
FNF.2 = 1 - SenSpec2[,2]

##### ROC curve
plot(FNF.2, TPF.2, xlim=c(0,1), ylim=c(0,1), type="b", lty=1, col
="blue",
     xlab="1 - specificity", ylab="sensitivity", main="ROC Curve")
```

Out[51]:

sensitivity	specificity
1	0
0.5913978	0.7594248
0.4372760	0.8668869
0.3297491	0.9269335
0.2293907	0.9570540
0.1362007	0.9782355
0.07885305	0.98950641
0.04301075	0.99358725
0.01075269	0.99727944
0.007168459	0.999222697
0.003584229	1.000000000
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1
0	1


ROC Curve
sensitivity
1 - specificity
0.0
0.2
0.4
0.6
0.8
1.0

In [53]:
```
glm1 <- glm(Response ~ ., data = train, family = binomial)
summary(glm1)
```

```
Warning message:
: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Out[53]:

```
Call:
glm(formula = Response ~ ., family = binomial, data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9224  -0.2814  -0.2650  -0.2487   3.2368

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.68446    0.16786 -15.992  < 2e-16 ***
ClustID2     -3.81945    2.75516  -1.386 0.165658
ClustID3      1.16586    0.98921   1.179 0.238565
ClustID4      0.28387    0.61766   0.460 0.645807
ClustID5     -0.60676    0.18446  -3.289 0.001004 **
ClustID6    -10.56046  186.76016  -0.057 0.954907
ClustID7      1.11896    0.71673   1.561 0.118474
RXTOTY1       0.17438    0.03904   4.466 7.96e-06 ***
TOTEXPY1      0.26209    0.08693   3.015 0.002570 **
SSECPY1X      0.14589    0.04066   3.588 0.000333 ***
ERTMCRY1      0.02128    0.02774   0.767 0.443007
OPOTCHY1      0.15288    0.07091   2.156 0.031089 *
IPDMCDY1      0.06253    0.02792   2.240 0.025115 *
ERDTCHY1      0.08406    0.02912   2.886 0.003896 **
OBDMCRY1     -0.02164    0.03164  -0.684 0.493966
OBDWCPY1      0.11242    0.10777   1.043 0.296892
OBCPRVY1      0.07543    0.05047   1.494 0.135050
OTHSLFY1      0.06119    0.03169   1.931 0.053507 .
ERTOSRY1      0.10380    0.04335   2.394 0.016657 *
PUBPY1X       0.07341    0.03125   2.349 0.018811 *
AMTTCHY1     -0.10223    0.07016  -1.457 0.145077
IPFPTRY1     -0.11415    0.05335  -2.140 0.032395 *
TOTSLFY1     -0.01086    0.04423  -0.246 0.805984
RXSTLY1      -0.14999    0.09693  -1.547 0.121765
TOTOSRY1     -0.16466    0.10998  -1.497 0.134350
OPFSLFY1     -0.14377    0.07565  -1.901 0.057348 .
ERDMCDY1      0.02823    0.02556   1.104 0.269417
HHATCHY1      0.25992    0.14181   1.833 0.066822 .
HHAEXPY1     -0.28087    0.14606  -1.923 0.054484 .
VISEXPY1     -0.07461    0.05518  -1.352 0.176291
OTHTCHY1     -0.07048    0.09883  -0.713 0.475734
TRSTPY1X      0.07550    0.02687   2.810 0.004949 **
FAMINCY1     -0.12518    0.05397  -2.319 0.020369 *
OBTOTVY1      0.09543    0.05379   1.774 0.076055 .
OBCSLFY1     -0.16783    0.09830  -1.707 0.087758 .
OBOEXPY1     -0.12483    0.08096  -1.542 0.123123
AMNMCRY1     -0.04893    0.09756  -0.501 0.616035
OBTTCHY1      0.06532    0.06999   0.933 0.350670
OBVVAY1       0.06159    0.03384   1.820 0.068765 .
TOTWCPY1     -0.11123    0.09348  -1.190 0.234081
OPPEXPY1     -0.17553    0.06533  -2.687 0.007210 **
OPOPRVY1      5.81802   10.94870   0.531 0.595149
SSIPY1X       0.04193    0.03424   1.224 0.220780
TOTOPUY1     -0.08882    0.12845  -0.691 0.489279
IPDMCRY1     -0.03242    0.03895  -0.832 0.405171
```

```
IPFMCRY1     -0.04179    0.04575  -0.913 0.360993
DVOTCHY1     -0.49748    0.38117  -1.305 0.191853
OBTPTRY1     -0.00796    0.17256  -0.046 0.963208
AMTPRVY1     -0.01285    0.15897  -0.081 0.935586
OPVMCDY1     -0.11247    0.07890  -1.425 0.154031
OPSPTRY1     -0.18522    0.07475  -2.478 0.013216 *
TOTVAY1      -0.06159    0.05644  -1.091 0.275145
OPDEXPY1      0.18359    0.07787   2.357 0.018401 *
OPTPTRY1      2.75531    1.15066   2.395 0.016641 *
OPTPRVY1     -2.68657    1.14781  -2.341 0.019253 *
OPOPTRY1     -5.92008   11.06128  -0.535 0.592506
OBVTRIY1     -0.19183    0.15335  -1.251 0.210959
OTHEXPY1      0.11043    0.10981   1.006 0.314599
OBDOPRY1     -0.18305    0.17436  -1.050 0.293801
OBVOPRY1      0.25349    0.20738   1.222 0.221595
DIVDPY1X     -0.08923    0.06687  -1.334 0.182049
OPFTCHY1     -0.05373    0.04291  -1.252 0.210460
OBCTCHY1      0.12240    0.09714   1.260 0.207658
OBCEXPY1     -0.12354    0.14406  -0.858 0.391117
OBNSLFY1      0.05530    0.03033   1.823 0.068230 .
AMNEXPY1     -0.09299    0.11050  -0.842 0.400065
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4497.0  on 10848  degrees of freedom
Residual deviance: 3968.9  on 10783  degrees of freedom
AIC: 4100.9

Number of Fisher Scoring iterations: 11
```

In [ ]:

In [62]:
```
SenSpec=function(model.name, dataset, cut.off.prob.seq){
  cut.off.prob.seq = as.vector(cut.off.prob.seq)
  np=length(cut.off.prob.seq)
  prob.good=predict(model.name, newdata=dataset, type="response")
  true.class=dataset$Response
  perf.matrix=matrix(rep(0,2*np), byrow=TRUE, ncol=2)
  for(i in 1:np){
    pred.class=true.class
    pred.class[which(prob.good >= cut.off.prob.seq[i])] = "Admitted
Y2"
    pred.class[which(prob.good< cut.off.prob.seq[i])] = "NotAdmitte
dY2"
    confusion=ftable(pred.class, true.class)
    specificity=confusion[1,1]/sum(confusion[,1])
    sensitivity=confusion[2,2]/sum(confusion[,2])
    perf.matrix[i,1] = sensitivity
    perf.matrix[i,2] = specificity
  }
  colnames(perf.matrix)=c("sensitivity", "specificity")
  perf.matrix
}
```

In [63]:
```
SenSpec3 <- SenSpec(model.name=glm1, dataset=test,
                    cut.off.prob.seq=seq(0,1,length=20))
SenSpec3
TPF.3 = SenSpec3[,1]
FNF.3 = 1 - SenSpec3[,2]
```

Out[63]:

sensitivity	specificity
1	0
0.4731183	0.8538671
0.3548387	0.9197435
0.2580645	0.9562767
0.1756272	0.9762923
0.1182796	0.9848426
0.09318996	0.99008939
0.07526882	0.99222697
0.05017921	0.99475321
0.04301075	0.99591916
0.03584229	0.99650214
0.02508961	0.99747377
0.02150538	0.99747377
0.01792115	0.99863972
0.01433692	0.99883405
0.01075269	0.99922270
0.01075269	0.99961135
0.007168459	0.999611349
0.003584229	0.999611349
0	1

In [64]:
```
glm2 <- glm(Response ~ ., data = train[,-1], family = binomial)
summary(glm2)
```

```
Warning message:
: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Out[64]:

```
Call:
glm(formula = Response ~ ., family = binomial, data = train[,
    -1])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0972  -0.2890  -0.2663  -0.2506   3.0877

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.181717   0.065048 -48.913  < 2e-16 ***
RXTOTY1      0.226287   0.035622   6.352 2.12e-10 ***
TOTEXPY1     0.304227   0.088294   3.446 0.000570 ***
SSECPY1X     0.219959   0.033623   6.542 6.07e-11 ***
ERTMCRY1     0.022077   0.028105   0.786 0.432155
OPOTCHY1     0.172737   0.074895   2.306 0.021089 *
IPDMCDY1     0.077958   0.027882   2.796 0.005174 **
ERDTCHY1     0.102051   0.030300   3.368 0.000757 ***
OBDMCRY1    -0.023350   0.031420  -0.743 0.457383
OBDWCPY1     0.114377   0.113400   1.009 0.313161
OBCPRVY1     0.108192   0.050556   2.140 0.032351 *
OTHSLFY1     0.051433   0.037519   1.371 0.170428
ERTOSRY1     0.108530   0.043537   2.493 0.012672 *
PUBPY1X      0.077326   0.030728   2.516 0.011855 *
AMTTCHY1    -0.128822   0.071818  -1.794 0.072856 .
IPFPTRY1    -0.123520   0.054694  -2.258 0.023921 *
TOTSLFY1     0.004257   0.044216   0.096 0.923292
RXSTLY1     -0.143777   0.094754  -1.517 0.129176
TOTOSRY1    -0.154800   0.111003  -1.395 0.163150
OPFSLFY1    -0.129732   0.077791  -1.668 0.095375 .
ERDMCDY1     0.036465   0.026366   1.383 0.166662
HHATCHY1     0.308062   0.139397   2.210 0.027108 *
HHAEXPY1    -0.316767   0.148455  -2.134 0.032863 *
VISEXPY1    -0.064112   0.054165  -1.184 0.236550
OTHTCHY1    -0.093346   0.098072  -0.952 0.341191
TRSTPY1X     0.081980   0.026990   3.037 0.002386 **
FAMINCY1    -0.128413   0.053827  -2.386 0.017048 *
OBTOTVY1     0.128811   0.054078   2.382 0.017222 *
OBCSLFY1    -0.149653   0.101562  -1.474 0.140612
OBOEXPY1    -0.148916   0.087108  -1.710 0.087350 .
AMNMCRY1    -0.054988   0.105826  -0.520 0.603337
OBTTCHY1     0.092112   0.070350   1.309 0.190420
OBVVAY1      0.067137   0.035454   1.894 0.058277 .
TOTWCPY1    -0.115819   0.097946  -1.182 0.237018
OPPEXPY1    -0.169351   0.064102  -2.642 0.008244 **
OPOPRVY1     6.272178  10.768922   0.582 0.560275
SSIPY1X      0.066424   0.033593   1.977 0.048004 *
TOTOPUY1    -0.105200   0.130637  -0.805 0.420656
IPDMCRY1    -0.028482   0.037504  -0.759 0.447583
IPFMCRY1    -0.037995   0.044079  -0.862 0.388706
DVOTCHY1    -0.497365   0.382450  -1.300 0.193440
OBTPTRY1    -0.042146   0.161677  -0.261 0.794338
AMTPRVY1    -0.009451   0.157044  -0.060 0.952010
OPVMCDY1    -0.105894   0.080638  -1.313 0.189115
```

```
OPSPTRY1    -0.163640   0.071681  -2.283 0.022438 *
TOTVAY1     -0.064447   0.058376  -1.104 0.269595
OPDEXPY1     0.198914   0.077584   2.564 0.010352 *
OPTPTRY1     2.599962   1.148543   2.264 0.023592 *
OPTPRVY1    -2.483270   1.144294  -2.170 0.029997 *
OPOPTRY1    -6.346969  10.879675  -0.583 0.559639
OBVTRIY1    -0.179328   0.152874  -1.173 0.240779
OTHEXPY1     0.064229   0.099775   0.644 0.519742
OBDOPRY1    -0.198763   0.177298  -1.121 0.262260
OBVOPRY1     0.281211   0.211361   1.330 0.183362
DIVDPY1X    -0.074730   0.067412  -1.109 0.267617
OPFTCHY1    -0.038319   0.040072  -0.956 0.338944
OBCTCHY1     0.146973   0.092739   1.585 0.113010
OBCEXPY1    -0.085625   0.134717  -0.636 0.525042
OBNSLFY1     0.062493   0.030037   2.081 0.037477 *
AMNEXPY1    -0.069223   0.115658  -0.599 0.549500
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4497.0  on 10848  degrees of freedom
Residual deviance: 3987.1  on 10789  degrees of freedom
AIC: 4107.1

Number of Fisher Scoring iterations: 10
```

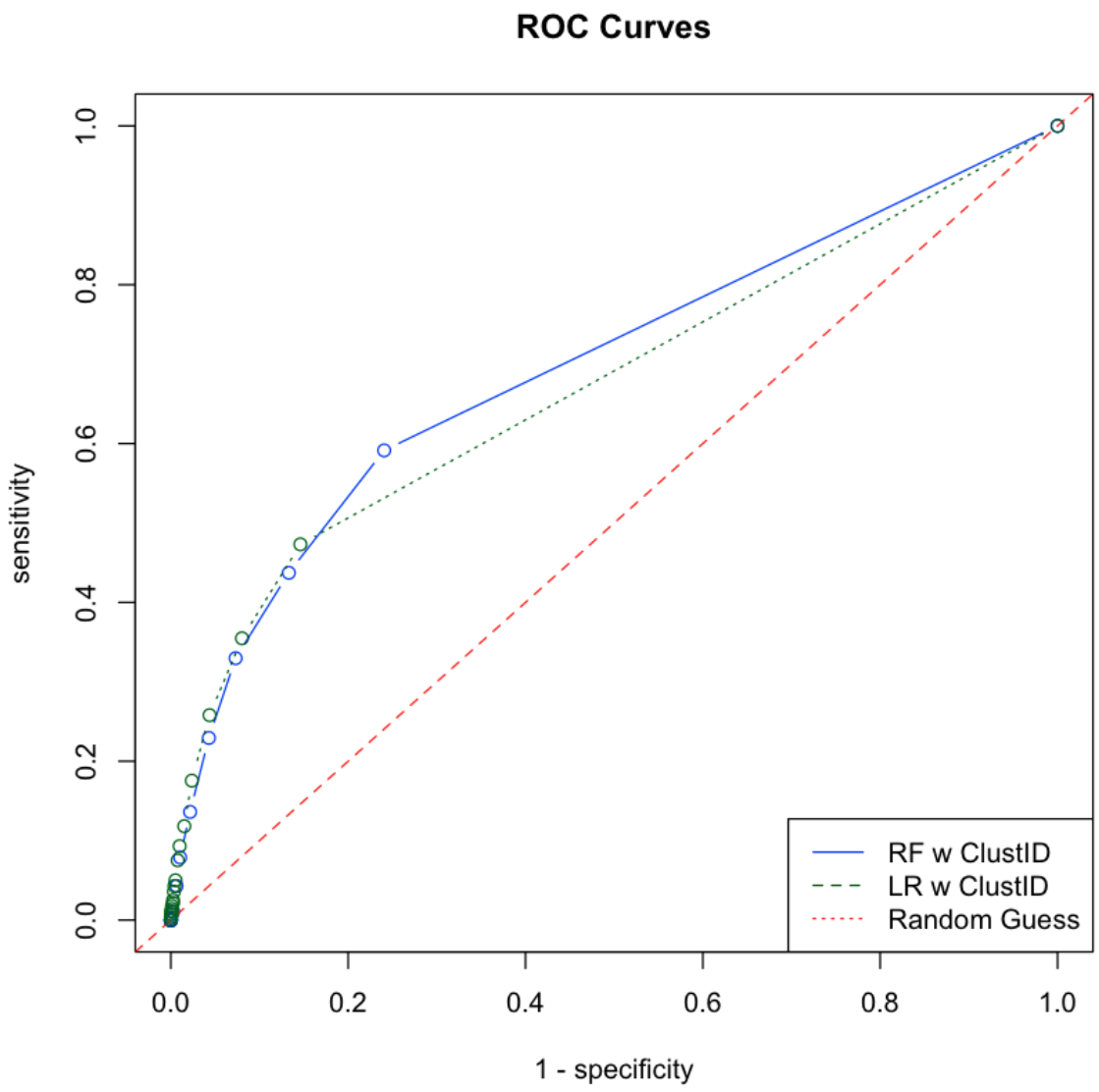
In [65]:
```
SenSpec4 <- SenSpec(model.name=glm2, dataset=test[,-1],
                    cut.off.prob.seq=seq(0,1,length=20))
SenSpec4
TPF.4 = SenSpec4[,1]
FNF.4 = 1 - SenSpec4[,2]
```

Out[65]:

sensitivity	specificity
1	0
0.5161290	0.8260785
0.2974910	0.9350952
0.2150538	0.9654100
0.1541219	0.9792072
0.1075269	0.9873688
0.09318996	0.98989506
0.05734767	0.99183832
0.05017921	0.99397590
0.04301075	0.99514186
0.04301075	0.99591916
0.03942652	0.99611349
0.03225806	0.99650214
0.02508961	0.99786242
0.01792115	0.99825107
0.01792115	0.99883405
0.01075269	0.99902837
0.007168459	0.999611349
0.003584229	0.999611349
0	1

In [ ]:

In [72]:
```
plot(FNF.1, TPF.1, xlim=c(0,1), ylim=c(0,1), type="b", lty=1, col
="blue",
     xlab="1 - specificity", ylab="sensitivity", main="ROC Curves")
abline(0,1, lty=2, col="red")

#lines(FNF.2, TPF.2, type="b", lty=3, col="purple")
lines(FNF.3, TPF.3, type="b", lty=3, col="darkgreen")
#lines(FNF.4, TPF.4, type="b", lty=3, col="darkorange")


legend("bottomright", c("RF w ClustID", "LR w ClustID",
                        "Random Guess"),
       col=c("blue", "darkgreen", "red"), lty=c(1,2,3))
```



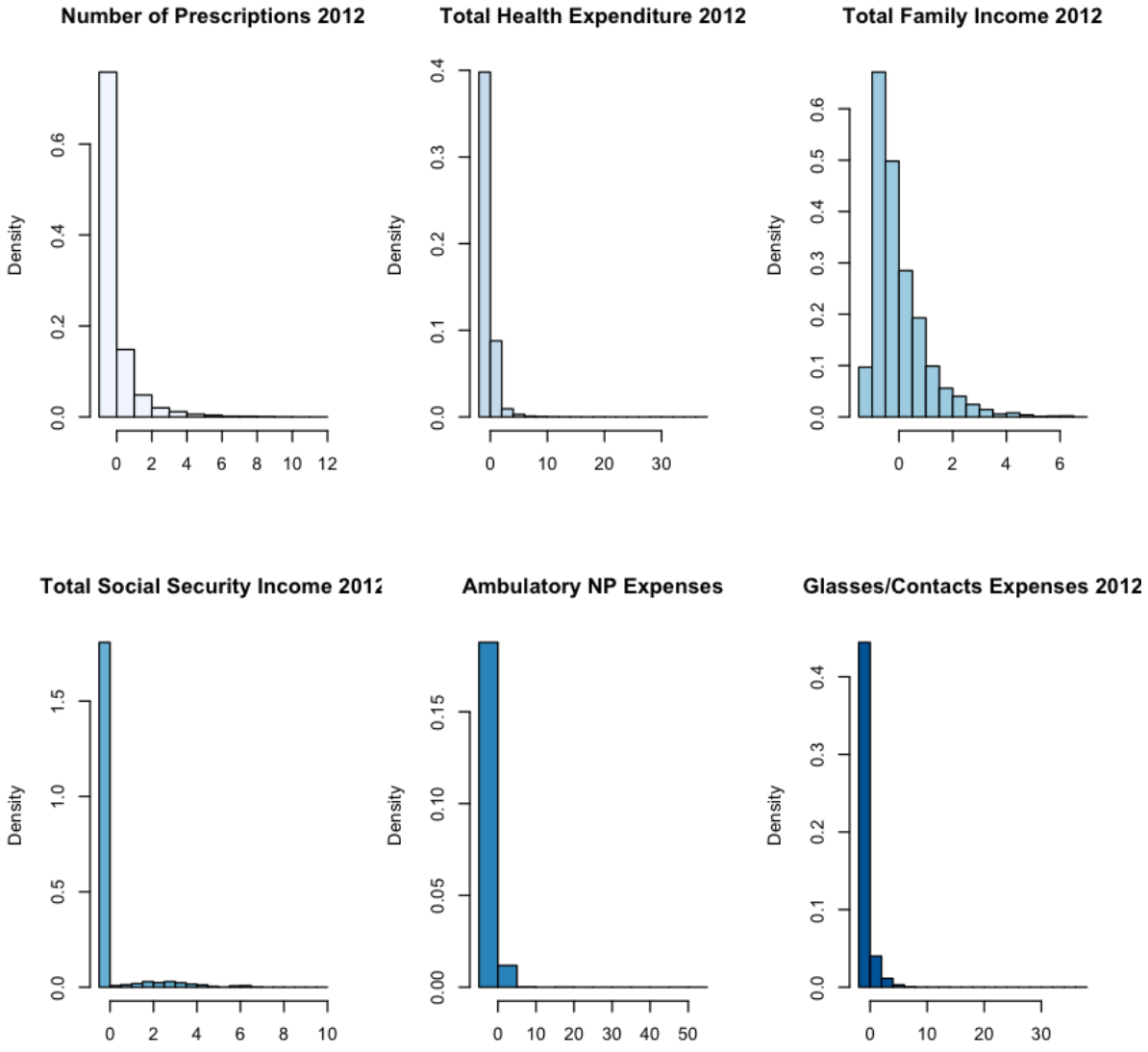
In [132]:
```
seq(0,1,length=20)
```

Out[132]:
```
 0   0.0526315789473684   0.105263157894737   0.157894736842105
 0.210526315789474   0.263157894736842   0.315789473684211
 0.368421052631579   0.421052631578947   0.473684210526316
 0.526315789473684   0.578947368421053   0.631578947368421
 0.684210526315789   0.736842105263158   0.789473684210526
 0.842105263157895   0.894736842105263   0.947368421052632   1
```

In [140]:
```
names(final)
```

Out[140]:
```
'ClustID'  'RXTOTY1'  'TOTEXPY1'  'SSECPY1X'  'ERTMCRY1'
'OPOTCHY1'  'IPDMCDY1'  'ERDTCHY1'  'OBDMCRY1'  'OBDWCPY1'
'OBCPRVY1'  'OTHSLFY1'  'ERTOSRY1'  'PUBPY1X'  'AMTTCHY1'
'IPFPTRY1'  'TOTSLFY1'  'RXSTLY1'  'TOTOSRY1'  'OPFSLFY1'
'ERDMCDY1'  'HHATCHY1'  'HHAEXPY1'  'VISEXPY1'  'OTHTCHY1'
'TRSTPY1X'  'FAMINCY1'  'OBTOTVY1'  'OBCSLFY1'  'OBOEXPY1'
'AMNMCRY1'  'OBTTCHY1'  'OBVVAY1'  'TOTWCPY1'  'OPPEXPY1'
'OPOPRVY1'  'SSIPY1X'  'TOTOPUY1'  'IPDMCRY1'  'IPFMCRY1'
'DVOTCHY1'  'OBTPTRY1'  'AMTPRVY1'  'OPVMCDY1'  'OPSPTRY1'
'TOTVAY1'  'OPDEXPY1'  'OPTPTRY1'  'OPTPRVY1'  'OPOPTRY1'
'OBVTRIY1'  'OTHEXPY1'  'OBDOPRY1'  'OBVOPRY1'  'DIVDPY1X'
'OPFTCHY1'  'OBCTCHY1'  'OBCEXPY1'  'OBNSLFY1'  'AMNEXPY1'
'Response'
```

In [174]:
```
par(mfrow=c(2,3))
cols = brewer.pal(6, "Blues")

hist(final[,'RXTOTY1'], main = "Number of Prescriptions 2012", freq
=F, col = cols[1], xlab = "")
hist(final[,'TOTEXPY1'], main = "Total Health Expenditure 2012", fr
eq=F, col = cols[2], xlab = "")
hist(final[,'FAMINCY1'], main = "Total Family Income 2012", freq=F,
col = cols[3], xlab = "")
hist(final[,'SSECPY1X'], main = "Total Social Security Income 201
2", freq=F, col = cols[4], xlab = "")
hist(final[,'AMNEXPY1'], main = "Ambulatory NP Expenses", freq=F, c
ol = cols[5], xlab = "")
hist(final[,'VISEXPY1'], main = "Glasses/Contacts Expenses 2012", f
req=F, col = cols[6], xlab = "")

par(mfrow=c(1,1))
```



In [146]:
```
par(mfrow=c(1,1))
col = brewer.pal(6, "Greys")
col[1]
```

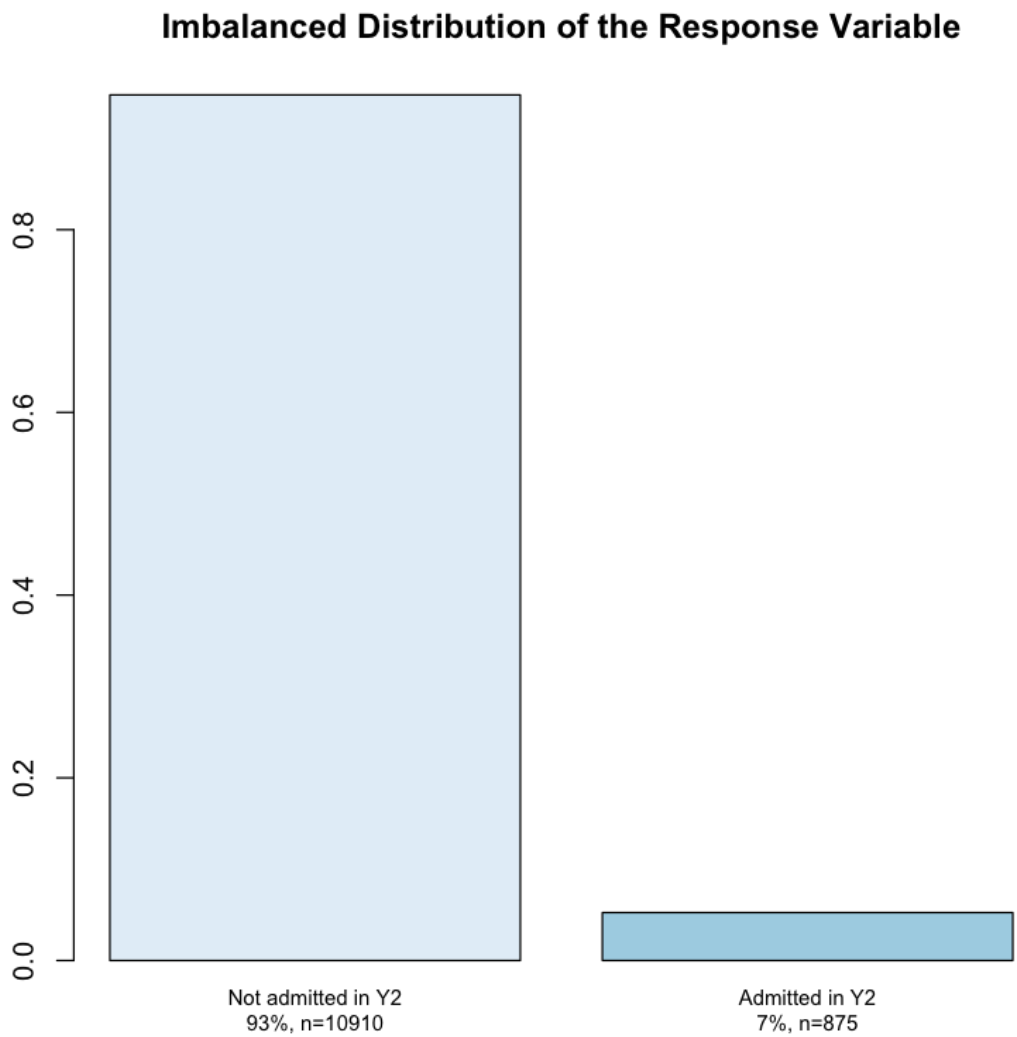
Out[146]:
```
'#F7F7F7'
```

In [150]:
```
polygon(density(final[,'RXTOTY1']), col = brewer.pal(6,"Blues"))
```

```
Error in polygon(density(final[, "RXTOTY1"]), col = brewer.pal(6,
"Blues")): plot.new has not been called yet
```

In [177]:
```
outcome <- prop.table(table(final$Response))
bp <- barplot(outcome, col = brewer.pal(2, "Blues"),
        main = "Imbalanced Distribution of the Response Variable",
        xlab = "",
        ylab = "",
        names.arg = c("Not admitted in Y2\n93%, n=10910","Admitted
in Y2\n7%, n=875"),
        cex.names = .75,
        legend.text = F
)
#title("Imbalanced distribution of the response", line = +3)
#labels <- c("93%\n(n=10910)", "7%\n(n = 875)")
#text(bp, outcome, labels, cex=.6, pos=3)
```

```
Warning message:
In brewer.pal(2, "Blues"): minimal value for n is 3, returning req
uested palette with 3 different levels
```



In [197]:
```
str(final)
```

```
'data.frame':   16274 obs. of  61 variables:
 $ ClustID : Factor w/ 7 levels "1","2","3","4",..: 5 5 5 5 5 5 5
5 5 5 ...
 $ RXTOTY1 : num  -0.453 -0.453 -0.453 1.634 -0.453 ...
 $ TOTEXPY1: num  -0.323 -0.323 -0.315 -0.238 -0.309 ...
 $ SSECPY1X: num  -0.289 -0.289 -0.289 -0.289 -0.289 ...
 $ ERTMCRY1: num  -0.0703 -0.0703 -0.0703 -0.0703 -0.0703 ...
 $ OPOTCHY1: num  -0.0689 -0.0689 -0.0689 -0.0689 -0.0689 ...
 $ IPDMCDY1: num  -0.0782 -0.0782 -0.0782 -0.0782 -0.0782 ...
 $ ERDTCHY1: num  -0.166 -0.166 -0.166 -0.166 -0.166 ...
 $ OBDMCRY1: num  -0.133 -0.133 -0.133 -0.133 -0.133 ...
 $ OBDWCPY1: num  -0.0284 -0.0284 -0.0284 -0.0284 -0.0284 ...
 $ OBCPRVY1: num  -0.0646 -0.0646 -0.0646 -0.0646 -0.0646 ...
 $ OTHSLFY1: num  -0.038 -0.038 -0.038 -0.038 -0.038 ...
 $ ERTOSRY1: num  -0.0498 -0.0498 -0.0498 -0.0498 -0.0498 ...
 $ PUBPY1X : num  -0.0853 -0.0853 -0.0853 -0.0853 -0.0853 ...
 $ AMTTCHY1: num  -0.058 -0.058 -0.058 -0.058 -0.058 ...
 $ IPFPTRY1: num  -0.0738 -0.0738 -0.0738 -0.0738 -0.0738 ...
 $ TOTSLFY1: num  -0.309 -0.309 -0.253 -0.273 -0.305 ...
 $ RXSTLY1 : num  -0.0413 -0.0413 -0.0413 0.1211 -0.0413 ...
 $ TOTOSRY1: num  -0.0694 -0.0694 -0.0694 -0.0694 -0.0694 ...
 $ OPFSLFY1: num  -0.0873 -0.0873 -0.0873 -0.0873 -0.0873 ...
 $ ERDMCDY1: num  -0.0938 -0.0938 -0.0938 -0.0938 -0.0938 ...
 $ HHATCHY1: num  -0.0651 -0.0651 -0.0651 -0.0651 -0.0651 ...
 $ HHAEXPY1: num  -0.062 -0.062 -0.062 -0.062 -0.062 ...
 $ VISEXPY1: num  -0.221 -0.221 -0.221 -0.221 -0.221 ...
 $ OTHTCHY1: num  -0.044 -0.044 -0.044 -0.044 -0.044 ...
 $ TRSTPY1X: num  -0.07 -0.07 -0.07 -0.07 -0.07 ...
 $ FAMINCY1: num  0.447 0.447 0.447 -0.629 -0.629 ...
 $ OBTOTVY1: num  -0.429 -0.429 -0.429 0.152 -0.313 ...
 $ OBCSLFY1: num  -0.087 -0.087 -0.087 -0.087 -0.087 ...
 $ OBOEXPY1: num  -0.162 -0.162 -0.162 -0.162 -0.162 ...
 $ AMNMCRY1: num  -0.0325 -0.0325 -0.0325 -0.0325 -0.0325 ...
 $ OBTTCHY1: num  -0.0741 -0.0741 -0.0741 -0.0741 -0.0741 ...
 $ OBVVAY1 : num  -0.0495 -0.0495 -0.0495 -0.0495 -0.0495 ...
 $ TOTWCPY1: num  -0.0473 -0.0473 -0.0473 -0.0473 -0.0473 ...
 $ OPPEXPY1: num  -0.069 -0.069 -0.069 -0.069 -0.069 ...
 $ OPOPRVY1: num  -0.0822 -0.0822 -0.0822 -0.0822 -0.0822 ...
 $ SSIPY1X : num  -0.156 -0.156 -0.156 -0.156 -0.156 ...
 $ TOTOPUY1: num  -0.0246 -0.0246 -0.0246 -0.0246 -0.0246 ...
 $ IPDMCRY1: num  -0.0736 -0.0736 -0.0736 -0.0736 -0.0736 ...
 $ IPFMCRY1: num  -0.0766 -0.0766 -0.0766 -0.0766 -0.0766 ...
 $ DVOTCHY1: num  -0.0949 -0.0949 -0.0949 -0.0949 -0.0949 ...
 $ OBTPTRY1: num  -0.0458 -0.0458 -0.0458 -0.0458 -0.0458 ...
 $ AMTPRVY1: num  -0.0508 -0.0508 -0.0508 -0.0508 -0.0508 ...
 $ OPVMCDY1: num  -0.064 -0.064 -0.064 -0.064 -0.064 ...
 $ OPSPTRY1: num  -0.0835 -0.0835 -0.0835 -0.0835 -0.0835 ...
 $ TOTVAY1 : num  -0.0419 -0.0419 -0.0419 -0.0419 -0.0419 ...
 $ OPDEXPY1: num  -0.128 -0.128 -0.128 -0.128 -0.128 ...
 $ OPTPTRY1: num  -0.0974 -0.0974 -0.0974 -0.0974 -0.0974 ...
 $ OPTPRVY1: num  -0.0961 -0.0961 -0.0961 -0.0961 -0.0961 ...
 $ OPOPTRY1: num  -0.0836 -0.0836 -0.0836 -0.0836 -0.0836 ...
 $ OBVTRIY1: num  -0.0449 -0.0449 -0.0449 -0.0449 -0.0449 ...
 $ OTHEXPY1: num  -0.0464 -0.0464 -0.0464 -0.0464 -0.0464 ...
```

```
 $ OBDOPRY1: num  -0.0428 -0.0428 -0.0428 -0.0428 -0.0428 ...
 $ OBVOPRY1: num  -0.0405 -0.0405 -0.0405 -0.0405 -0.0405 ...
 $ DIVDPY1X: num  -0.0823 -0.0823 -0.0823 -0.0823 -0.0823 ...
 $ OPFTCHY1: num  -0.125 -0.125 -0.125 -0.125 -0.125 ...
 $ OBCTCHY1: num  -0.095 -0.095 -0.095 -0.095 -0.095 ...
 $ OBCEXPY1: num  -0.0904 -0.0904 -0.0904 -0.0904 -0.0904 ...
 $ OBNSLFY1: num  -0.0733 -0.0733 -0.0733 -0.0733 -0.0733 ...
 $ AMNEXPY1: num  -0.0638 -0.0638 -0.0638 -0.0638 -0.0638 ...
 $ Response: Factor w/ 2 levels "NotAdmittedY2",..: 1 1 1 1 1 1 1
1 1 1 ...
```

In [ ]: