R Notebook

0. Packages

```r
library(ElemStatLearn)
```

```
## Warning: package 'ElemStatLearn' was built under R version 3.5.3
```

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```r
library(RColorBrewer)
```

```
## Warning: package 'RColorBrewer' was built under R version 3.5.2
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.5.3
```

```r
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 3.5.3
```

```
##
## Attaching package: 'cowplot'
```

```
## The following object is masked from 'package:ggplot2':
##
##     ggsave
```

```r
library(e1071) # Naie Bayes
```

```
## Warning: package 'e1071' was built under R version 3.5.3
```

```r
library(caret) # Matrice de confusion
```

```
## Warning: package 'caret' was built under R version 3.5.3
```

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

```r
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.5.3
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

1. Données

```r
data(zip.train)
data(zip.test)
```

```r
dim(zip.train)
```

```
## [1] 7291  257
```

```r
dim(zip.test)
```

```
## [1] 2007  257
```

```r
zip.train[1:10,1:10]
```

```
##       [,1] [,2] [,3] [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]
##  [1,]    6   -1   -1   -1 -1.000 -1.000 -1.000 -1.000 -0.631  0.862
##  [2,]    5   -1   -1   -1 -0.813 -0.671 -0.809 -0.887 -0.671 -0.853
##  [3,]    4   -1   -1   -1 -1.000 -1.000 -1.000 -1.000 -1.000 -1.000
##  [4,]    7   -1   -1   -1 -1.000 -1.000 -0.273  0.684  0.960  0.450
##  [5,]    3   -1   -1   -1 -1.000 -1.000 -0.928 -0.204  0.751  0.466
##  [6,]    6   -1   -1   -1 -1.000 -1.000 -0.397  0.983 -0.535 -1.000
##  [7,]    3   -1   -1   -1 -0.830  0.442  1.000  1.000  0.479 -0.328
##  [8,]    1   -1   -1   -1 -1.000 -1.000 -1.000 -1.000  0.510 -0.213
##  [9,]    0   -1   -1   -1 -1.000 -1.000 -0.454  0.879 -0.745 -1.000
## [10,]    1   -1   -1   -1 -1.000 -1.000 -1.000 -1.000 -0.909  0.801
```

```r
zip.test[1:10,1:10]
```

```
##       [,1] [,2] [,3] [,4]   [,5] [,6]   [,7]   [,8]   [,9]  [,10]
##  [1,]    9   -1   -1   -1 -1.000 -1.0 -0.948 -0.561  0.148  0.384
##  [2,]    6   -1   -1   -1 -1.000 -1.0 -1.000 -1.000 -1.000 -1.000
##  [3,]    3   -1   -1   -1 -0.593  0.7  1.000  1.000  1.000  1.000
##  [4,]    6   -1   -1   -1 -1.000 -1.0 -1.000 -1.000 -1.000 -1.000
##  [5,]    6   -1   -1   -1 -1.000 -1.0 -1.000 -1.000 -0.858 -0.106
##  [6,]    0   -1   -1   -1 -1.000 -1.0 -1.000  0.195  1.000  0.054
##  [7,]    0   -1   -1   -1 -1.000 -1.0 -0.785  0.775  0.268 -1.000
##  [8,]    0   -1   -1   -1 -1.000 -1.0 -0.914 -0.688 -0.736  0.956
##  [9,]    6   -1   -1   -1 -1.000 -1.0 -1.000 -1.000 -0.761  0.438
## [10,]    9   -1   -1   -1 -1.000 -1.0 -0.904 -0.060  0.638  0.678
```

Première colonne: les digit 256 autres colonnes = valeures des pixels (imgage 16x16 = 256 pixels)

2. Préparation des data.frame

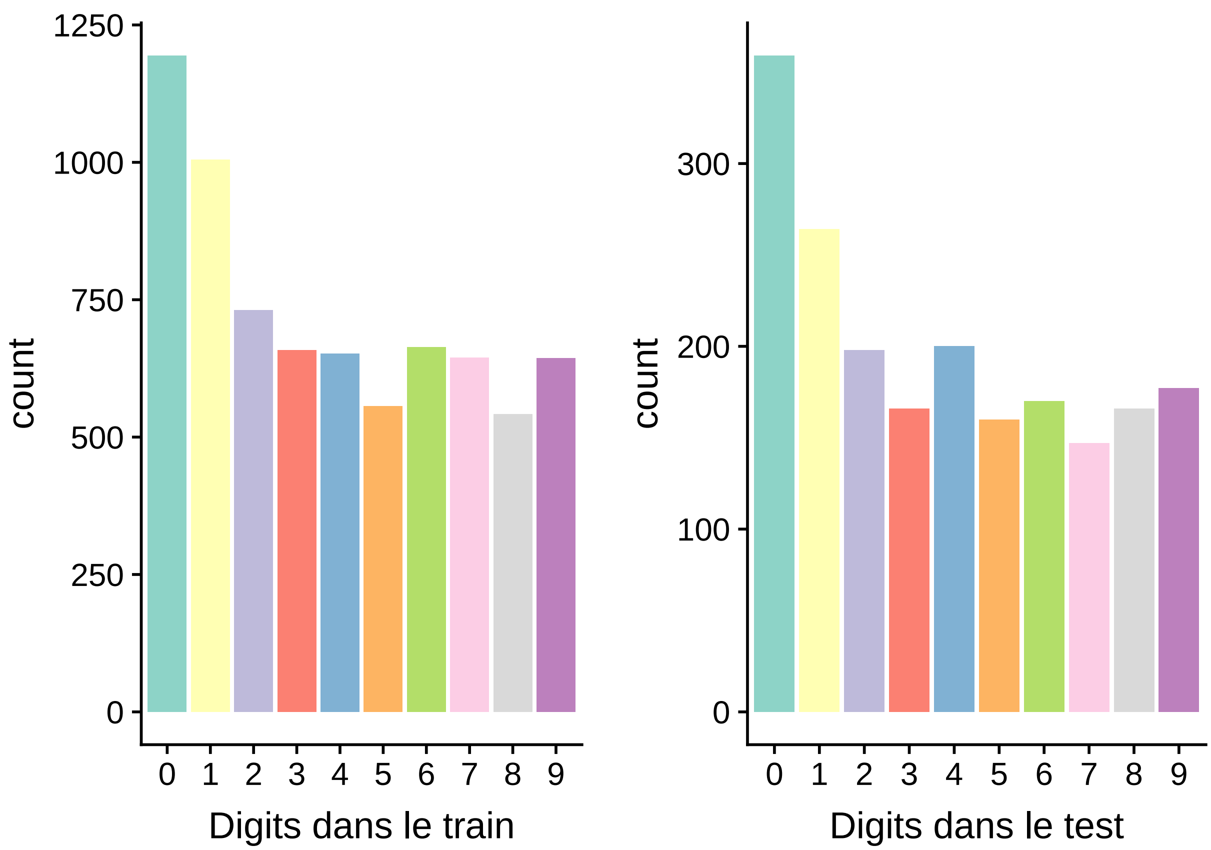
```r
train = as.data.frame(zip.train)
colnames(train)[1] = "Digit_train"
test = as.data.frame(zip.test)
colnames(test)[1] = "Digit_test"
```

```r
rm(zip.train)
rm(zip.test)
```

```r
#apply(train,2, class)
#apply(test,2, class)
```

```r
train$Digit_train = as.factor(train$Digit_train)
test$Digit_test = as.factor(test$Digit_test)
```

```r
plot_grid(
  ggplot(data = train, aes(x = train$Digit_train)) + geom_bar( fill = brewer.pal(n = 10, name = "Set3"))
  ggplot(data = test, aes(x = test$Digit_test)) + geom_bar( fill = brewer.pal(n = 10, name = "Set3")) +
          )
```



2.1 Choix aléatoire des deux digits à prédire

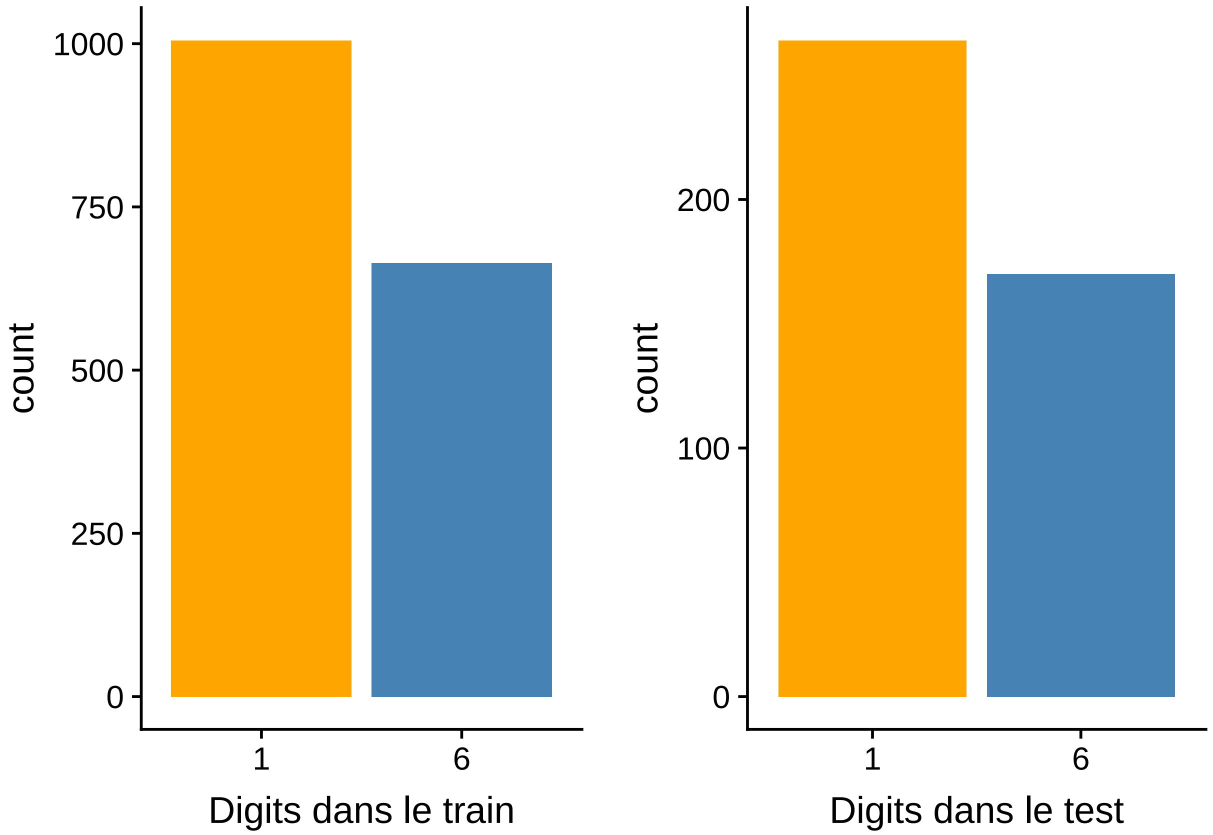
```r
set.seed(2) # POur que les valeurs aléatoires ne changent pas au moins le temps de commencer le code
digits_chosen  = sample(seq(from=0, to= 9, by=1), 2, replace = F)
```

2.2 Data frame final pour entrainer les modèles

```r
train.2d = subset(train, (Digit_train %in% c(digits_chosen)))
test.2d = subset(test, (Digit_test %in% c(digits_chosen)))
train.2d$Digit_train = as.factor(as.character(train.2d$Digit_train))
test.2d$Digit_test = as.factor(as.character(test.2d$Digit_test))
rm(train)
```

```r
rm(test)
```

```r
plot_grid(
  ggplot(data = train.2d, aes(x = train.2d$Digit_train)) + geom_bar(fill = c("orange", "steelblue")) + x
  ggplot(data = test.2d, aes(x = test.2d$Digit_test)) + geom_bar(fill = c("orange", "steelblue")) + xlab
          )
```



3. Classifiers

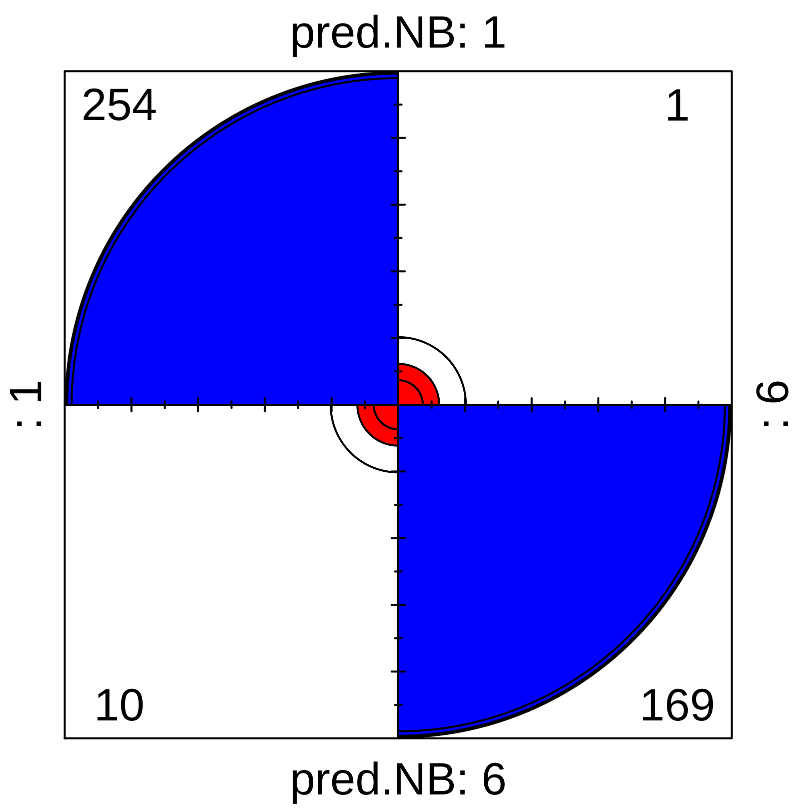
3.1 Naive Bayes

```r
# Entrainement du modèle
mod.NB = naiveBayes(Digit_train ~ ., data = train.2d)

# On prédit les données de test avec le modèle obtenu
pred.NB = predict(mod.NB, subset(test.2d, select = -Digit_test))

# Matrice de confusion
confusionMatrix(pred.NB, test.2d$Digit_test)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   1   6
##            1 254   1
##            6  10 169
```

```
##
##                Accuracy : 0.9747
##                  95% CI : (0.9551, 0.9873)
##     No Information Rate : 0.6083
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.9473
##
##  Mcnemar's Test P-Value : 0.01586
##
##             Sensitivity : 0.9621
##             Specificity : 0.9941
##          Pos Pred Value : 0.9961
##          Neg Pred Value : 0.9441
##              Prevalence : 0.6083
##          Detection Rate : 0.5853
##    Detection Prevalence : 0.5876
##       Balanced Accuracy : 0.9781
##
##        'Positive' Class : 1
##
```

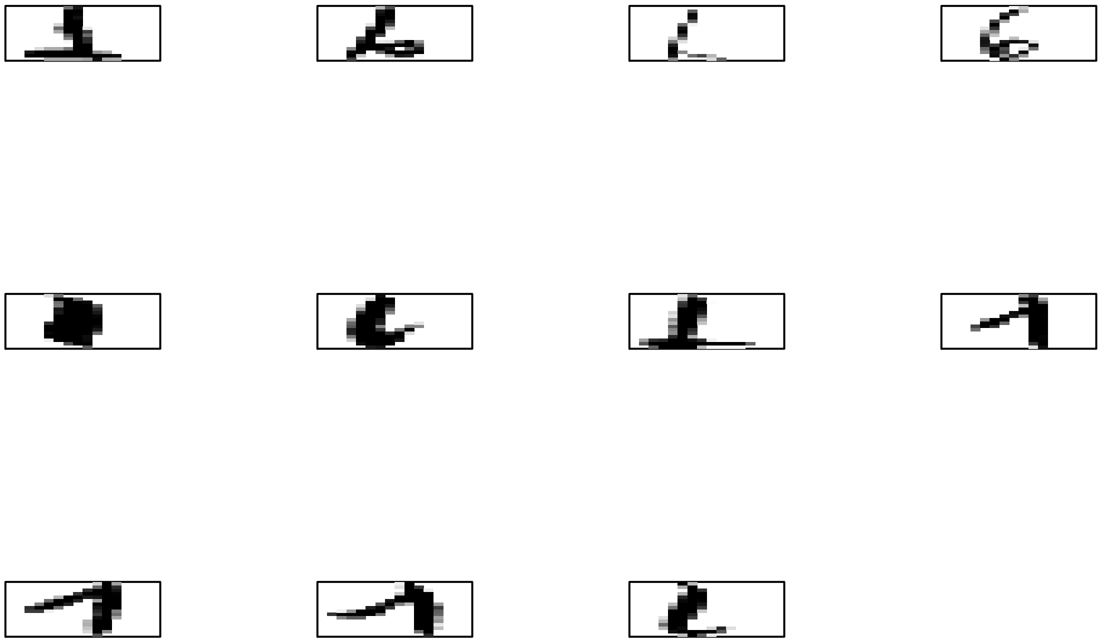
```r
fourfoldplot(table(pred.NB, test.2d$Digit_test), color = c("red","blue"))
```




```r
# Visualisation des mauvaises predictions
NB.errors = cbind(test.2d[which(test.2d$Digit_test != pred.NB),], pred.NB[which(test.2d$Digit_test != pr
colnames(NB.errors)[c(1,258)] = c("Label","Pred")
```

```r
par(mfrow=c(3,4))
for(i in 1:dim(NB.errors)[1]){
image(zip2image(as.matrix(NB.errors[i,2:257])), col=gray(256:0/256), zlim=c(0,1), xlab="", ylab="",xaxt=
}
```

```
## [1] "digit  -1  taken"

## Warning in matrix(im, 16, 16, byrow = TRUE): la longueur des données [255]
## n'est pas un diviseur ni un multiple du nombre de lignes [16]

## [1] "digit  -1  taken"

## Warning in matrix(im, 16, 16, byrow = TRUE): la longueur des données [255]
## n'est pas un diviseur ni un multiple du nombre de lignes [16]

## [1] "digit  -1  taken"

## Warning in matrix(im, 16, 16, byrow = TRUE): la longueur des données [255]
## n'est pas un diviseur ni un multiple du nombre de lignes [16]

## [1] "digit  -1  taken"

## Warning in matrix(im, 16, 16, byrow = TRUE): la longueur des données [255]
## n'est pas un diviseur ni un multiple du nombre de lignes [16]

## [1] "digit  -1  taken"

## Warning in matrix(im, 16, 16, byrow = TRUE): la longueur des données [255]
## n'est pas un diviseur ni un multiple du nombre de lignes [16]

## [1] "digit  -1  taken"

## Warning in matrix(im, 16, 16, byrow = TRUE): la longueur des données [255]
## n'est pas un diviseur ni un multiple du nombre de lignes [16]

## [1] "digit  -1  taken"

## Warning in matrix(im, 16, 16, byrow = TRUE): la longueur des données [255]
## n'est pas un diviseur ni un multiple du nombre de lignes [16]

## [1] "digit  -1  taken"

## Warning in matrix(im, 16, 16, byrow = TRUE): la longueur des données [255]
## n'est pas un diviseur ni un multiple du nombre de lignes [16]

## [1] "digit  -1  taken"

## Warning in matrix(im, 16, 16, byrow = TRUE): la longueur des données [255]
## n'est pas un diviseur ni un multiple du nombre de lignes [16]

## [1] "digit  -1  taken"

## Warning in matrix(im, 16, 16, byrow = TRUE): la longueur des données [255]
## n'est pas un diviseur ni un multiple du nombre de lignes [16]

## [1] "digit  -1  taken"

## Warning in matrix(im, 16, 16, byrow = TRUE): la longueur des données [255]
## n'est pas un diviseur ni un multiple du nombre de lignes [16]
```



```r
NB.errors[,c(1,258)]
```

```
##      Label Pred
## 53       1    6
## 234      1    6
## 437      1    6
## 1223     6    1
## 1332     1    6
## 1335     1    6
## 1631     1    6
## 1814     1    6
## 1815     1    6
## 1816     1    6
## 1885     1    6
```

3.2 Logistic regression

```r
mod.logit = glm(Digit_train ~., data = train.2d, family = "binomial")
```

```
## Warning: glm.fit: algorithm did not converge
```

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

```r
pred.logit = predict(mod.logit, subset(test.2d, select = -Digit_test))
```

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

```r
table(pred.logit, test.2d$Digit_test)
```

```
##
## pred.logit           1 6
##   -28162889317535.2  0 1
##   -9074.05101570798  0 1
##   -1312.98165503331  0 1
##   -1147.11586264867  0 1
##   -144.706738687884  0 1
##   -118.008797020753  0 1
##   -80.8666990133061  1 0
##   -67.3734892686916  1 0
##   -67.0965380856942  1 0
##   -59.619118452676   0 1
##   -58.2396253977349  1 0
##   -57.6069971998731  1 0
##   -55.7070799149151  1 0
##   -54.8626903913391  1 0
##   -51.6625934247859  1 0
##   -51.3248258381427  1 0
##   -51.0756909019619  1 0
##   -50.963569720232   1 0
##   -50.6461538111253  1 0
##   -50.4816544839559  1 0
##   -50.4541839364174  1 0
##   -49.9909293022283  1 0
##   -49.055484928409   1 0
##   -49.0433359296876  1 0
##   -48.7214403222824  1 0
##   -48.6056677212109  1 0
##   -48.3862420318255  1 0
##   -48.365895642266   1 0
##   -48.2774268915309  1 0
##   -47.8829337670977  1 0
##   -47.7887756326018  1 0
##   -47.5350550279982  1 0
##   -47.2759131313287  1 0
##   -46.9935454538572  1 0
##   -46.9520934758766  1 0
##   -46.7099690030591  1 0
##   -46.6606859922467  1 0
##   -46.4703096298399  1 0
##   -46.3841326133843  1 0
##   -46.3036329142633  1 0
##   -46.279187792672   1 0
##   -46.1861014728638  1 0
##   -46.1278015483549  1 0
##   -45.9410216688921  1 0
##   -45.8693328551526  1 0
##   -45.5420816409096  1 0
##   -45.4633611220343  1 0
##   -45.2683868763343  1 0
##   -45.1195904349806  1 0
##   -45.0957930460208  1 0
```

```
##   -45.0437479055108  1 0
##   -44.5838589448904  1 0
##   -44.5064564768836  1 0
##   -44.4636021629631  1 0
##   -44.4266640295318  1 0
##   -44.4041826035027  1 0
##   -44.339299461506   1 0
##   -44.2824220601542  1 0
##   -44.2564357110241  1 0
##   -44.2370879446971  1 0
##   -44.229433531058   1 0
##   -44.1913392463612  1 0
##   -43.9996053062496  1 0
##   -43.8578180529221  1 0
##   -43.8470880727036  1 0
##   -43.691695657275   1 0
##   -43.6756466018123  1 0
##   -43.6520776768593  1 0
##   -43.4854016090248  1 0
##   -43.4723997824913  1 0
##   -43.4219013929105  1 0
##   -43.2816321602004  1 0
##   -43.1053085153762  1 0
##   -43.088535169496   1 0
##   -43.0124585197336  1 0
##   -42.9678551888283  1 0
##   -42.9440632275364  1 0
##   -42.9055229131263  1 0
##   -42.8657818200081  1 0
##   -42.8310344432248  1 0
##   -42.8101878325397  1 0
##   -42.5618803359903  1 0
##   -42.5086526644591  1 0
##   -42.4700446682764  1 0
##   -42.4086649972232  1 0
##   -42.4080309577257  1 0
##   -42.3262978139828  1 0
##   -42.2148247273071  1 0
##   -42.1974957774     1 0
##   -42.1521946784778  1 0
##   -42.0909279840125  1 0
##   -42.0546980815634  1 0
##   -42.0311526119258  1 0
##   -41.9203191162233  1 0
##   -41.884793086625   1 0
##   -41.6771413535753  1 0
##   -41.662110462923   1 0
##   -41.6581339284239  1 0
##   -41.6332987002752  1 0
##   -41.6107477787373  1 0
##   -41.6017779186586  1 0
##   -41.5407767450452  1 0
##   -41.5040435206101  1 0
##   -41.4295475911858  1 0
```

```
##   -41.2912988245225  1 0
##   -41.2893873423382  1 0
##   -41.1830373250414  1 0
##   -41.1298318323752  1 0
##   -41.0794150107904  1 0
##   -41.0081965951467  1 0
##   -40.9605404791946  1 0
##   -40.858339443017   1 0
##   -40.7254820577145  1 0
##   -40.6420812191864  1 0
##   -40.503783820066   1 0
##   -40.3697910220799  1 0
##   -40.3473324096121  1 0
##   -40.3321924229385  1 0
##   -40.2424716635796  1 0
##   -40.2192374304723  1 0
##   -40.0600480421563  1 0
##   -39.9929365691642  1 0
##   -39.977293619937   1 0
##   -39.9113987145975  1 0
##   -39.8041727516975  1 0
##   -39.7674375145725  1 0
##   -39.757226001333   1 0
##   -39.7424190911988  1 0
##   -39.7314650214321  1 0
##   -39.6169289212776  1 0
##   -39.5055713052061  1 0
##   -39.4859048789222  1 0
##   -39.4737121809812  1 0
##   -39.4438744136714  1 0
##   -39.3656881801653  1 0
##   -39.232826255713   1 0
##   -39.0719870252942  1 0
##   -38.9936579254645  1 0
##   -38.9426813568352  1 0
##   -38.941174631138   1 0
##   -38.9133003456882  1 0
##   -38.8988796018457  1 0
##   -38.8739620298948  1 0
##   -38.8728734273682  1 0
##   -38.8718829758     1 0
##   -38.8064046512154  1 0
##   -38.7801925476961  1 0
##   -38.7748627256951  1 0
##   -38.771135009607   1 0
##   -38.770421258072   1 0
##   -38.7057838065084  1 0
##   -38.6466471995809  1 0
##   -38.5794392821845  1 0
##   -38.5267798099085  1 0
##   -38.4383065593138  1 0
##   -38.2285278609706  1 0
##   -38.193904080319   1 0
##   -38.1487653849981  1 0
```

```
## -38.0673582222953 1 0
## -38.0463807123087 1 0
## -38.0276677216025 1 0
## -38.0213146883179 1 0
## -37.8787494997814 1 0
## -37.7839633431868 1 0
## -37.660428144336  1 0
## -37.62749201111   1 0
## -37.5586859571267 1 0
## -37.4803307737675 1 0
## -37.4720152954687 1 0
## -37.2787128891359 1 0
## -37.0159243298986 1 0
## -36.8879381428778 1 0
## -36.6548213000351 1 0
## -36.6401702712756 1 0
## -36.5025947211325 1 0
## -36.4471558380392 1 0
## -36.3767391705333 1 0
## -36.3630861576967 1 0
## -36.2322111207832 1 0
## -36.0894218621688 1 0
## -36.0004699464043 1 0
## -35.9974259035807 1 0
## -35.8531831147047 1 0
## -35.7499803078026 1 0
## -35.6436387736758 1 0
## -35.6061578677836 1 0
## -35.5266947022101 1 0
## -35.4662017621013 1 0
## -35.4567820650409 1 0
## -35.4153975881563 1 0
## -35.2424535411119 1 0
## -35.213661804577  1 0
## -35.1627336243691 1 0
## -35.0789953461426 1 0
## -35.0686954862977 1 0
## -35.038090103284  1 0
## -34.9377149406719 1 0
## -34.9350414471555 1 0
## -34.7949068958405 1 0
## -34.6498119098396 1 0
## -34.6286237152963 1 0
## -34.6171372943136 1 0
## -34.4521817810746 1 0
## -34.3791847061802 1 0
## -34.3578396007943 1 0
## -34.2806189104886 1 0
## -34.2778531834556 1 0
## -33.9624607616424 1 0
## -33.9352550596959 1 0
## -33.6739313710495 1 0
## -32.9023646132409 1 0
## -32.8640373599555 1 0
```

```
##   -32.7288282087538  1 0
##   -32.5703380835548  1 0
##   -32.2352628032022  1 0
##   -32.1634481535802  1 0
##   -32.0076127741559  1 0
##   -31.6968315874255  1 0
##   -31.6877306851093  1 0
##   -31.4045305119434  1 0
##   -31.1854186901983  1 0
##   -31.0724713660675  1 0
##   -30.8267842972928  1 0
##   -30.7145072227431  1 0
##   -30.3257107938407  1 0
##   -30.2955418970814  1 0
##   -29.8096709506426  1 0
##   -29.4061379229097  1 0
##   -28.0263990609092  1 0
##   -27.9990740346839  1 0
##   -27.9346269440503  1 0
##   -27.8342950716396  1 0
##   -27.8240889466688  1 0
##   -27.7295634948896  1 0
##   -27.6442547700935  1 0
##   -27.5721733660903  1 0
##   -27.5219470463489  1 0
##   -27.389969996133   1 0
##   -26.9245404096655  1 0
##   -26.8698827849003  1 0
##   -26.8088737367798  1 0
##   -26.0598068933905  1 0
##   -25.7790401391176  0 1
##   -25.5759036612944  1 0
##   -24.7356109514803  1 0
##   -24.4688840938034  1 0
##   -23.6886194342369  1 0
##   -22.0075507403089  1 0
##   -21.9715768675815  1 0
##   -21.1576342319095  1 0
##   -20.8228240331446  1 0
##   -20.792107641646   1 0
##   -17.5817185501219  1 0
##   -17.2442982289722  1 0
##   -16.468812937841   1 0
##   -16.0040736443916  0 1
##   -13.44051342483    1 0
##   -12.9415895952843  1 0
##   -12.4686531988846  1 0
##   -11.1222475320101  1 0
##   -10.614899127766   1 0
##   -7.89337029173475  1 0
##   -7.52570824095164  0 1
##   -7.15308502347034  0 1
##   -5.73086793912807  0 1
##   -5.41968686612381  1 0
```

```
##   -0.900070755604247 0 1
##   -0.382969073085405 1 0
##   0.585641687852331  1 0
##   0.737277382322645  0 1
##   2.13462156430614   0 1
##   2.7150450832487    0 1
##   5.7061978215861    0 1
##   8.53443835358485   0 1
##   9.53522882224206   0 1
##   10.4974349822078   1 0
##   11.589121691024    0 1
##   11.6726024037125   0 1
##   12.440083863672    0 1
##   12.6783044992699   0 1
##   13.6065822934543   1 0
##   14.0424936531781   0 1
##   15.2474717968435   0 1
##   15.6751869674408   0 1
##   15.9657453735563   0 1
##   18.0579998269677   0 1
##   18.4317817846822   0 1
##   18.8760084103124   0 1
##   20.705454080904    0 1
##   21.2769510896178   0 1
##   22.3565650015225   0 1
##   22.8572642002619   0 1
##   23.0105720697175   0 1
##   23.609141323228    0 1
##   23.7242667869941   0 1
##   24.0440822788296   0 1
##   24.7197182990785   0 1
##   24.8395783916712   0 1
##   25.0487453298192   0 1
##   25.1705707481742   0 1
##   25.5903985860568   0 1
##   25.92956753006     0 1
##   26.61789756699     0 1
##   26.766546808045    0 1
##   27.0988397210822   0 1
##   27.4756758569929   0 1
##   27.6285354909414   0 1
##   27.7745611966602   0 1
##   27.9848090623927   0 1
##   28.0341136482675   0 1
##   28.2283674138671   0 1
##   28.5456069820211   0 1
##   28.7358814348336   0 1
##   28.8555357696896   0 1
##   28.8580486520732   0 1
##   28.9639945488816   1 0
##   29.1778241158754   0 1
##   29.4848320806414   0 1
##   29.8492945835096   0 1
##   30.3146450492713   0 1
```

```
##    30.7840544239953   0 1
##    30.8108140774712   0 1
##    31.1643364240153   0 1
##    31.746310926319    0 1
##    32.0620785355932   0 1
##    32.2522492239368   1 0
##    32.3593146573039   0 1
##    32.7074562697671   0 1
##    32.8014854299763   0 1
##    32.9143502713487   0 1
##    33.2232456774436   0 1
##    33.3436995912707   0 1
##    33.5680300729509   0 1
##    33.6764172037656   0 1
##    33.8422394882218   0 1
##    33.8730847999323   0 1
##    34.7836650598765   0 1
##    34.9493017034329   0 1
##    34.9941197291191   0 1
##    35.045537237078    0 1
##    35.1157488540921   0 1
##    35.5537790339804   0 1
##    35.7435926859034   0 1
##    35.9214504435949   0 1
##    36.4404720563034   0 1
##    36.5458815003221   0 1
##    37.023187114668    0 1
##    37.0987285577648   0 1
##    37.5876649688362   0 1
##    37.981656938071    0 1
##    38.2202785717018   0 1
##    38.446000689939    0 1
##    38.4782426600723   0 1
##    39.3723327962216   0 1
##    40.4239530694176   0 1
##    40.5082101372391   0 1
##    40.5395607790415   0 1
##    40.876699168999    0 1
##    41.0113285800326   0 1
##    41.231720320342    0 1
##    41.2401678718888   0 1
##    41.3104689932661   0 1
##    41.473950893349    0 1
##    41.4926525829578   0 1
##    41.8958153086278   0 1
##    42.0290364908797   0 1
##    42.0950680450114   0 1
##    42.2907601540137   0 1
##    42.4023584384995   0 1
##    42.4400640603126   0 1
##    42.7922788643118   0 1
##    42.8792186977735   0 1
##    42.9301893121592   0 1
##    43.0392873578676   0 1
```

```
##    44.2356809777411   0 1
##    44.6710235301871   0 1
##    44.7221316004507   0 1
##    45.1883931350603   0 1
##    45.5718695730175   0 1
##    46.337232098922    0 1
##    46.3963131377968   0 1
##    47.0214895963509   0 1
##    47.0293209470547   0 1
##    47.2165686511871   0 1
##    47.5783411184311   0 1
##    47.5875950066911   0 1
##    47.6064345345731   0 1
##    47.7314300509024   0 1
##    48.1531107591145   0 1
##    48.1996479409281   0 1
##    48.4375994865841   0 1
##    48.5576076309153   0 1
##    49.0945122806079   0 1
##    49.259310531379    0 1
##    49.4692739788879   0 1
##    49.5149440609166   0 1
##    49.518819479621    0 1
##    53.9594497331927   0 1
##    54.1413323265078   0 1
##    54.1484980866298   0 1
##    54.2929814373201   0 1
##    54.3455358361316   0 1
##    54.6337535489729   0 1
##    54.7202010345718   0 1
##    55.1263498814224   0 1
##    55.1768675396379   0 1
##    56.483252334896    0 1
##    56.6484293464018   0 1
##    59.005842134633    0 1
##    61.0793016911339   0 1
##    61.3187027005988   0 1
##    61.4951911478202   0 1
##    62.441995531939    0 1
##    64.2239156040523   0 1
##    64.3816133762521   0 1
##    64.4829125399774   0 1
##    65.858709349166    0 1
##    66.5231143485726   0 1
##    67.5925993443889   0 1
##    68.9370173430507   0 1
##    69.2075313114256   0 1
##    69.3404988030888   0 1
##    72.1043765409486   1 0
##    74.0932762949742   0 1
##    81.7428261724635   0 1
##    87.9749989521079   0 1
##    90.8266421497683   1 0
##    101.865596737654   0 1
```

```
##    116.896305084738    0 1
##    169.411422753881    0 1
##    191.006399706435    1 0
##    274.245302398725    1 0
##    4681.42472753445    0 1
##    46488.689831588     0 1
```