stat149_project_EDA

Zecai Liang

2/28/2017

OUTLINE

Part 1. Exploratory Analysis 1.1 - check: dimension of data 1.2 - check: missing value determine imputation method 1.3 - check: if unbalanced data (ratio of 0/1 labels) 1.4 - check: distribution of each variable (screwd? need transformation? feature engineering?) 1.5 - check: colinearity 1.6 - *research: possible interactions between variables (check domain knowledge/paper/reports/similar Kaggle competitions)*

Part 2. Visualize clustering/dimension reduction 2.1 PCA 2.2 t-SNE

Part 3. Baseline Classifier 3.1 GAM 3.1.1 full model (no interaction yet) 3.1.2 vairable selection: forward/backward/both (no interaction yet) 3.2 random forest 3.3. SVM

Part 4. Tune Model 4.1 add interaction 4.2 add feature engineering 4.3 tune parameters by cross validation 4.4 ensemble multiple models if necessary (popular vote)

```
## load libraries
options(warning = -1)

library("ggplot2")
library("gridExtra")

library("cluster")
library(factoextra)
```

```
## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ
```

```
library(corrplot)

library(e1071)
library(gam)
```

```
## Loading required package: splines

## Loading required package: foreach

## Loaded gam 1.14
```

```
library(randomForest)
```

```
## randomForest 4.6-12

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':
##
##     combine

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

Part 1. Exploratory Analysis

1.1 - check: dimension of data

```
str(train)
```

```
## 'data.frame':    118529 obs. of  17 variables:
##  $ voted          : Factor w/ 2 levels "N","Y": 2 2 2 2 2 1 2 2 2 1 ...
##  $ gender         : Factor w/ 3 levels "F","M","U": 2 1 1 1 2 2 2 2 1 2 ...
##  $ cd             : int  7 6 2 7 5 1 2 2 4 1 ...
##  $ hd             : int  31 38 53 30 19 7 13 52 39 5 ...
##  $ age            : int  36 55 24 25 22 22 27 33 22 26 ...
##  $ dbdistance     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ vccdistance    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ party          : Factor w/ 6 levels "D","G","L","O",..: 6 6 6 1 5 6 1 6 1 1 ...
##  $ racename       : Factor w/ 11 levels "African-American",..: 5 11 2 2 2 2 11 2 2 2 ...
##  $ hsonly         : num  25.4 7.9 50.2 38 30.5 32 36.7 10.7 30.6 16.4 ...
##  $ mrrg           : num  63.4 97.8 7.6 8.5 19.1 7.5 10.5 60.2 17.1 10.4 ...
##  $ chldprsnt      : num  54 59.8 49.5 47.4 23.1 29.4 37.2 35.7 26.4 14.8 ...
##  $ cath           : num  16.7 16.7 14.6 13.1 16 13.5 14.1 8.1 13.3 8.4 ...
##  $ evang          : num  16.5 15.5 24 22.3 10.5 21.6 21.1 11.3 11.7 8.9 ...
##  $ nonchrst       : num  39.6 30.9 29.6 33.3 39.1 34 34.4 52.4 43.5 57.2 ...
##  $ otherchrst     : num  27.3 36.9 31.7 31.4 34.5 30.9 30.4 28.3 31.5 25.6 ...
##  $ days.since.reg: int  420 307 292 316 392 333 300 540 474 531 ...
```

```
## turn categorical variable into factors
train$cd <- factor(train$cd)
train$hd <- factor(train$hd)

test$cd <- factor(test$cd)
test$hd <- factor(test$hd)
```

```
colnames(train)
```

```
##  [1] "voted"          "gender"         "cd"             "hd"
##  [5] "age"            "dbdistance"     "vccdistance"    "party"
##  [9] "racename"       "hsonly"         "mrrg"           "chldprsnt"
## [13] "cath"           "evang"          "nonchrst"       "otherchrst"
## [17] "days.since.reg"
```

```
colnames(test)
```

```
##  [1] "gender"         "cd"             "hd"             "age"
##  [5] "dbdistance"     "vccdistance"    "party"          "racename"
##  [9] "hsonly"         "mrrg"           "chldprsnt"      "cath"
## [13] "evang"          "nonchrst"       "otherchrst"     "days.since.reg"
## [17] "Id"
```

```
## pool data
df1 <- cbind(train[-1])
df1$source <- "train"

df2 <- cbind(test[1:16])
df2$source <- "test"

all <- rbind(df1, df2)
```

1.2 - check: missing value

Most rows of `dbdistance` and `vccdistance` are missing.

```
# number of NA for train data
print("Number of NA for train data:")
```

```
## [1] "Number of NA for train data:"
```

```
colSums(is.na(train))
```

```
##          voted         gender             cd             hd            age
##              0              0              2              2              0
##     dbdistance    vccdistance          party       racename         hsonly
##         113247         113247              0              0              0
##           mrrg       chldprsnt           cath          evang       nonchrst
##              0              0              0              0              0
##     otherchrst days.since.reg
##              0              0
```

```
# number of NA for test data
print("Number of NA for test data:")
```

```
## [1] "Number of NA for test data:"
```

```
colSums(is.na(test))
```

```
##         gender             cd             hd            age     dbdistance
##              0              1              1              0          37758
##    vccdistance          party       racename         hsonly           mrrg
##          37758              0              0              0              0
##      chldprsnt           cath          evang       nonchrst     otherchrst
##              0              0              0              0              0
## days.since.reg             Id
##              0              0
```

1.3 - check: if unbalanced data

Conclusion: mostly balanced data, more Y than N.

```
table(train['voted'])
```

```
##
##     N     Y
## 38172 80357
```

1.4 - check: distribution of each variable

Check the distribution in train, test and the pooled data.

```
## function to plot categorical variable
  # plot 1: the distribution of train and test data
  # plot 2: the pertange of train data that didn't vote

bar_plot <- function(val, alpha = 0.8, width = 0.5, ncol = 2, angle = 0){
    p1 = ggplot(data = all, aes_string(x = val)) +
        geom_bar(aes(fill = source), alpha = alpha, position = "dodge") +
```

```
        ggtitle("Train + Test") +
       theme(axis.text.x = element_text(angle = angle, hjust = 1))

   p2 = ggplot(data = train, aes_string(x = val)) +
        geom_bar(aes(color = voted),
                 width = width, alpha = alpha, position = "fill") +
        scale_color_manual(values = c("red","blue")) +
        ggtitle("Train (%)") +
       theme(axis.text.x = element_text(angle = angle, hjust = 1))

    grid.arrange(p1, p2, ncol = ncol)
}
```

```
## function to plot quantitative variable
  # plot 1: the distribution of train and test data
  # plot 2: the pertange of train data that didn't vote

his_plot <- function(val, alpha = 0.8, width = 0.5, ncol = 2, angle = 0){
    p1 = ggplot(data = all, aes_string(x = val)) +
        geom_histogram(aes(fill = source), alpha = alpha, position = "dodge") +
        ggtitle("Train + Test") +
        theme(axis.text.x = element_text(angle = angle, hjust = 1))

   p2 = ggplot(data = train, aes_string(x = val)) +
        geom_histogram(aes(color = voted),
                       alpha = alpha, position = "fill") +
        scale_color_manual(values = c("red","blue")) +
        ggtitle("Train (%)") +
       theme(axis.text.x = element_text(angle = angle, hjust = 1))

    grid.arrange(p1, p2, ncol = ncol)
}
```

Gender

```
table(train["gender"], exclude = NULL)
```

```
##
##     F     M     U  <NA>
## 58414 59916   199     0
```

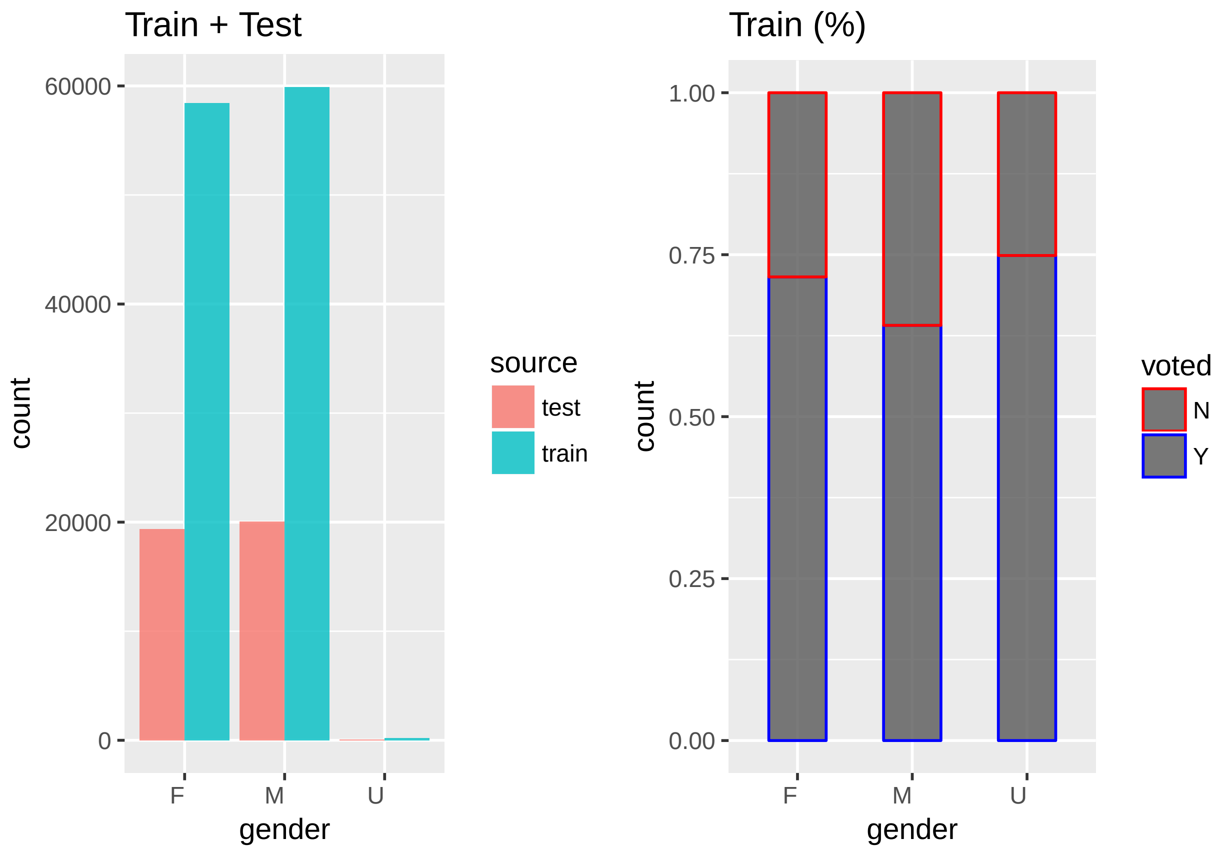
```
table(test["gender"], exclude = NULL)
```

```
##
##     F     M     U  <NA>
## 19385 20047    78     0
```

```
table(all["gender"], exclude = NULL)
```

```
##
##     F     M     U  <NA>
## 77799 79963   277     0
```

```
bar_plot("gender")
```



cd

congressional district

```
table(train["cd"], exclude = NULL)
```

```
##
##     1     2     3     4     5     6     7  <NA>
## 20229 20883 14459 14916 20156 14474 13410     2
```

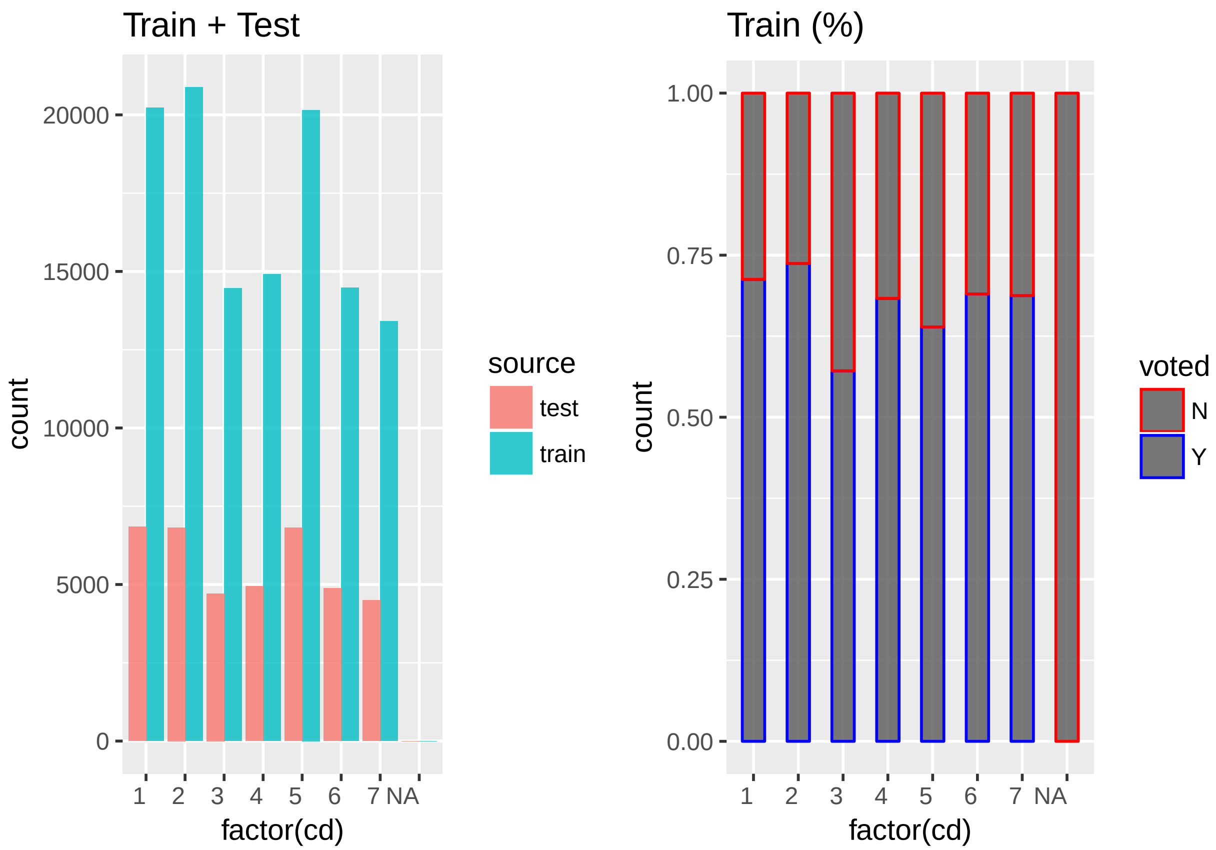
```
table(test["cd"], exclude = NULL)
```

```
##
##    1    2    3    4    5    6    7 <NA>
## 6846 6821 4712 4941 6815 4876 4498    1
```

```
table(all["cd"], exclude = NULL)
```

```
##
##     1     2     3     4     5     6     7  <NA>
## 27075 27704 19171 19857 26971 19350 17908     3
```

```r
bar_plot("factor(cd)")
```

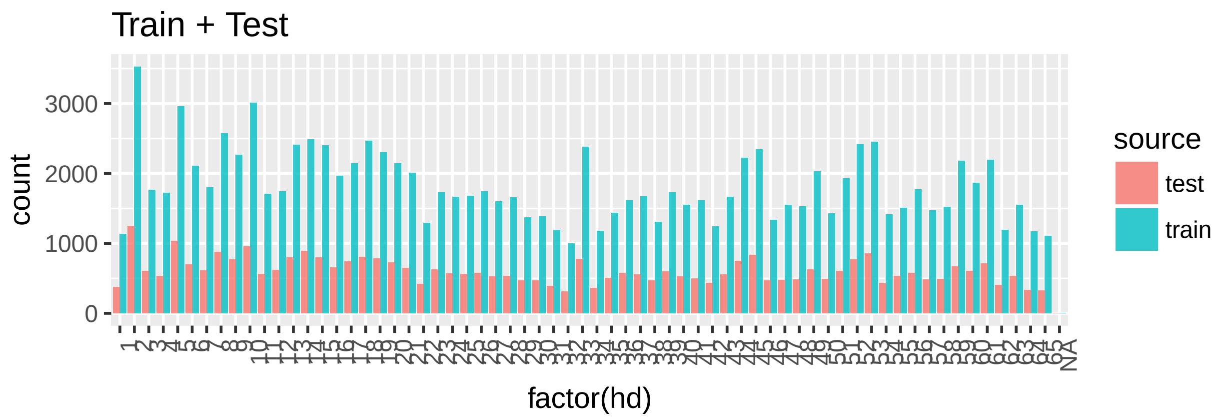


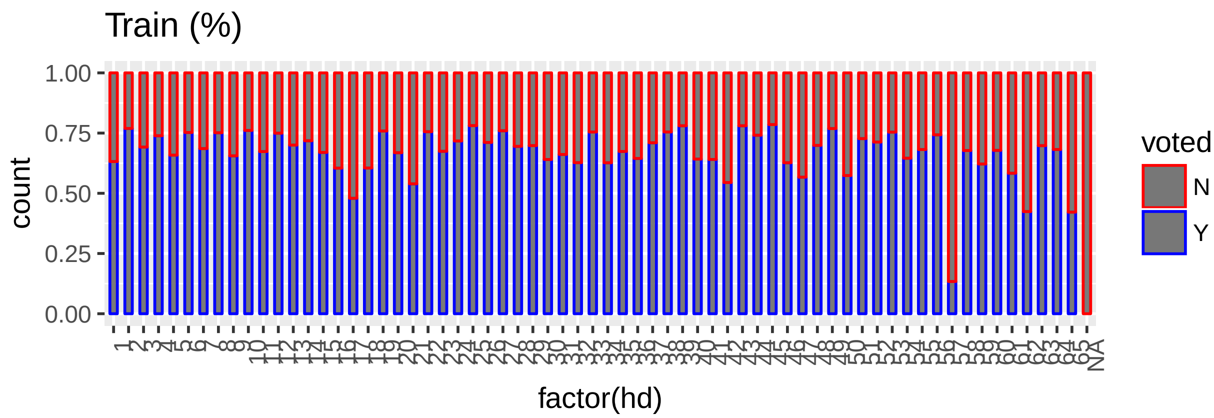
hd

state house district * some district (57) have high prpbability of not voting

```r
#table(train["hd"], exclude = NULL)
#table(test["hd"], exclude = NULL)
#table(all["hd"], exclude = NULL)
```

```r
bar_plot("factor(hd)", ncol = 1, angle = 90)
```

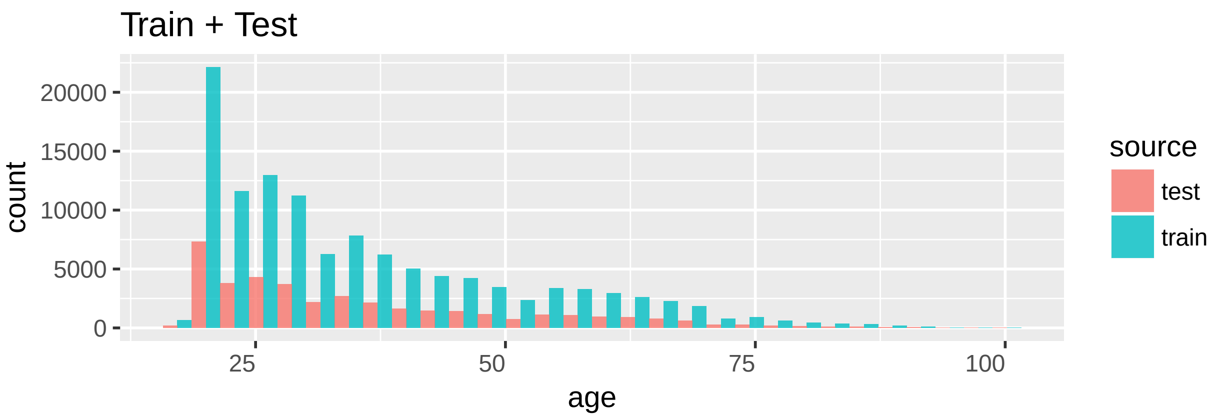


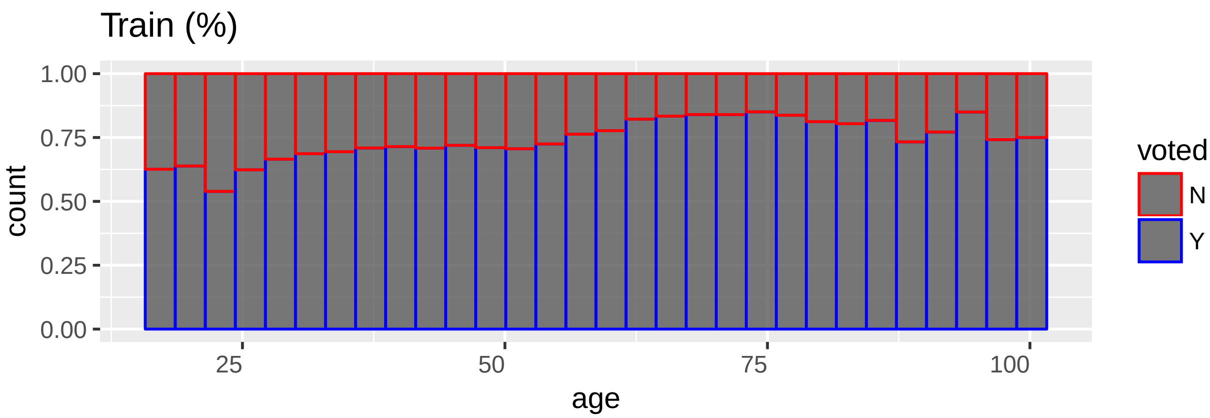


age

```r
his_plot("age", ncol = 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

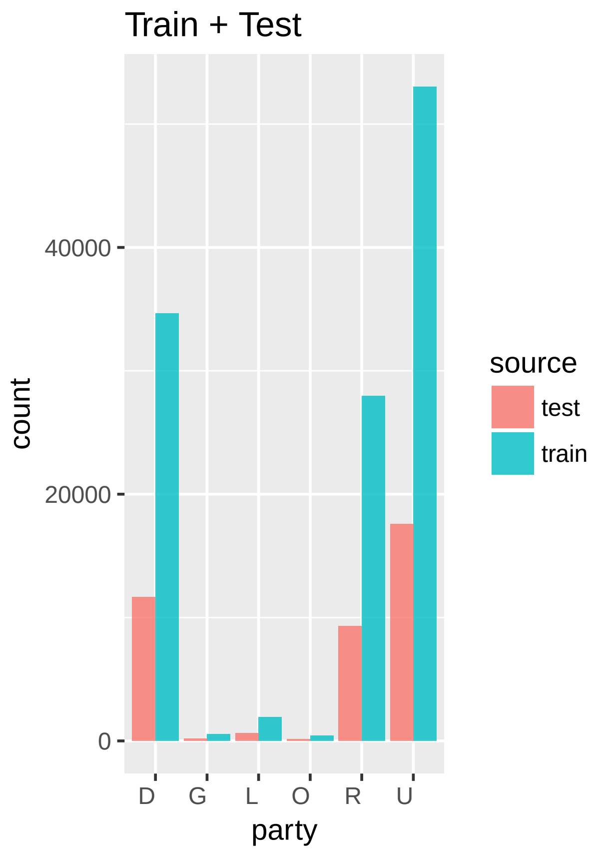


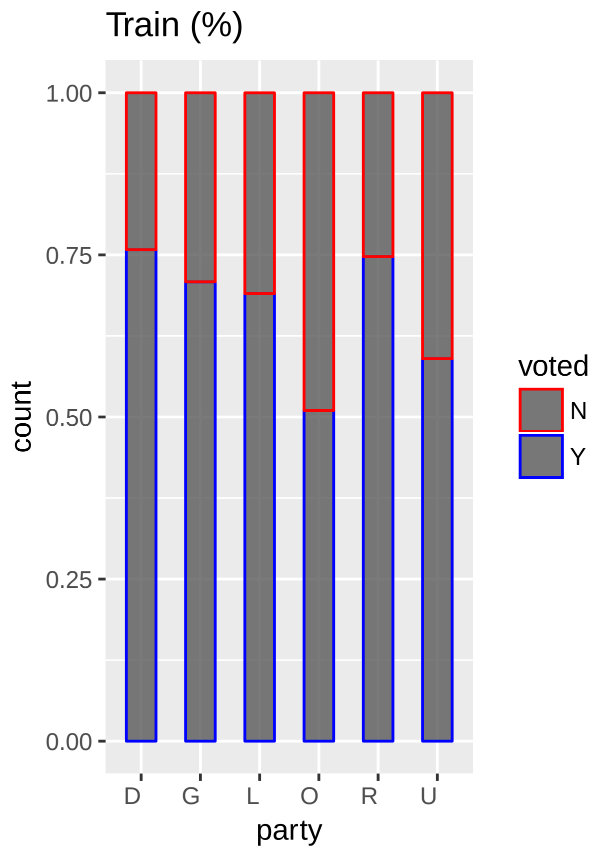


party

- Check with the reported distribution. Unaliated? (D=Democrat, R=Republican, L=Libertarian, G=Green, O=American Constitutional Party, U=Unaliated)

```
bar_plot("party")
```



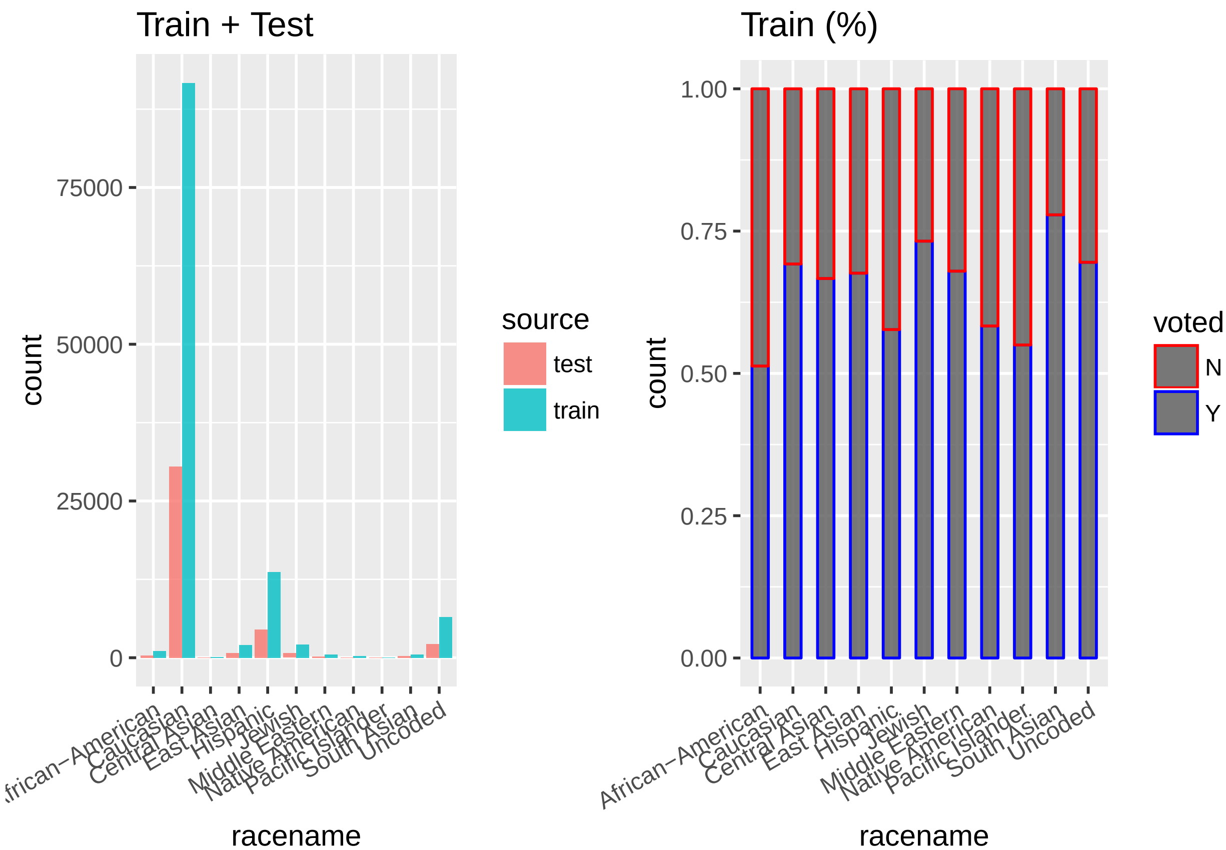


racename

(Race or religious aliation)

```
#table(train["racename"], exclude = NULL)
#table(test["racename"], exclude = NULL)
#table(all["racename"], exclude = NULL)

bar_plot("racename", angle = 30)
```



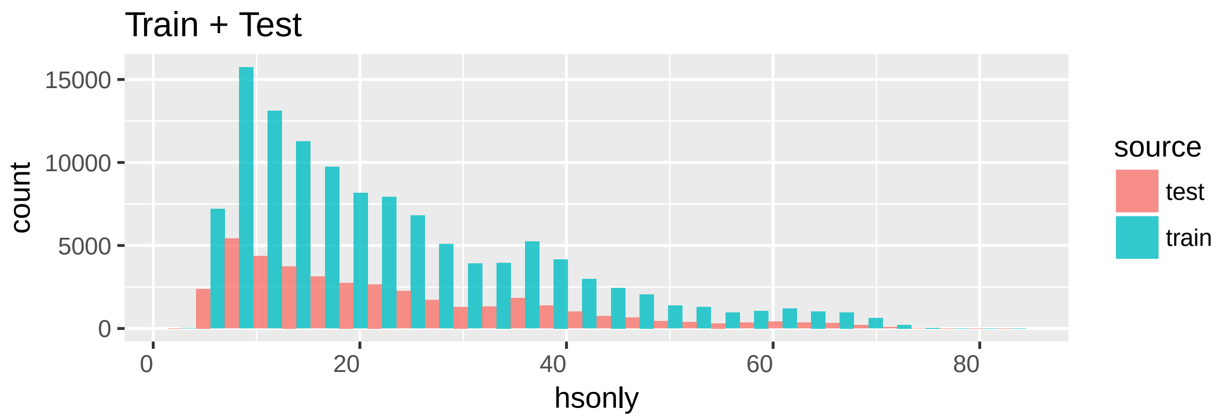
hsonly

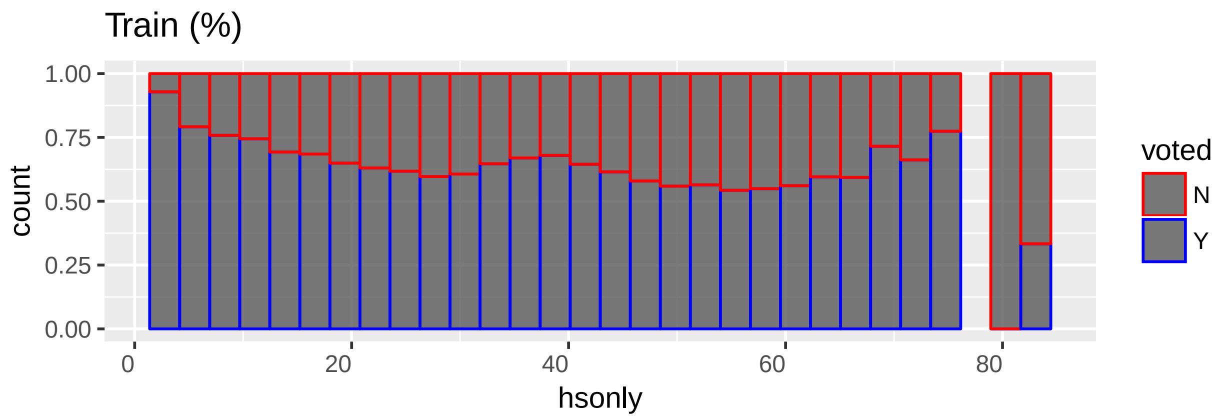
(score for likelihood of having high school as highest completed degree)

```r
his_plot("hsonly", ncol = 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```



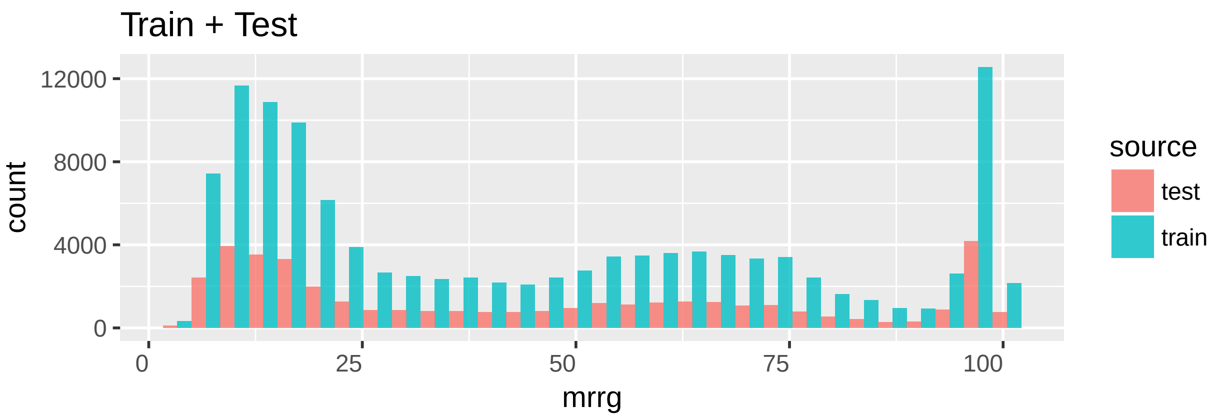


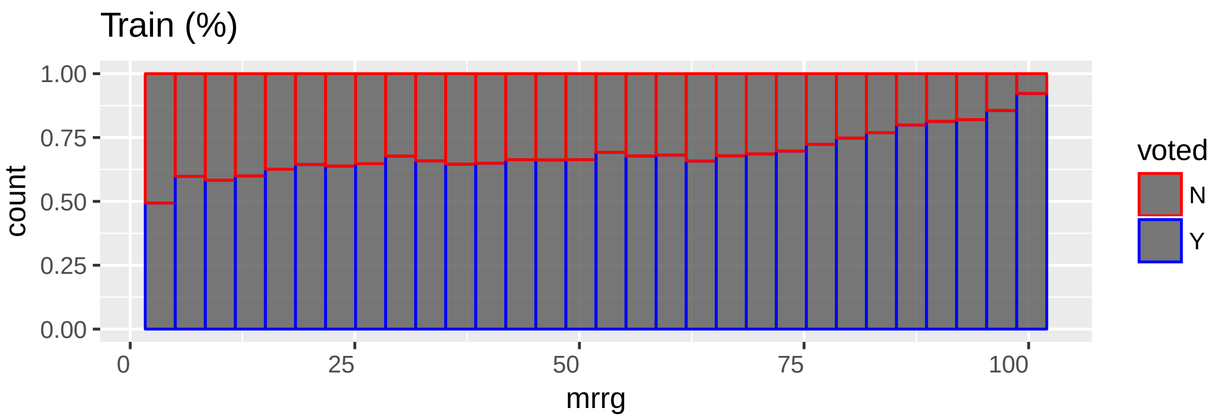
mrrg

(score for likelihood of being married) * two extrmes for plot 1; marraiged and voting seem negatiely correlated.

```r
his_plot("mrrg", ncol = 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



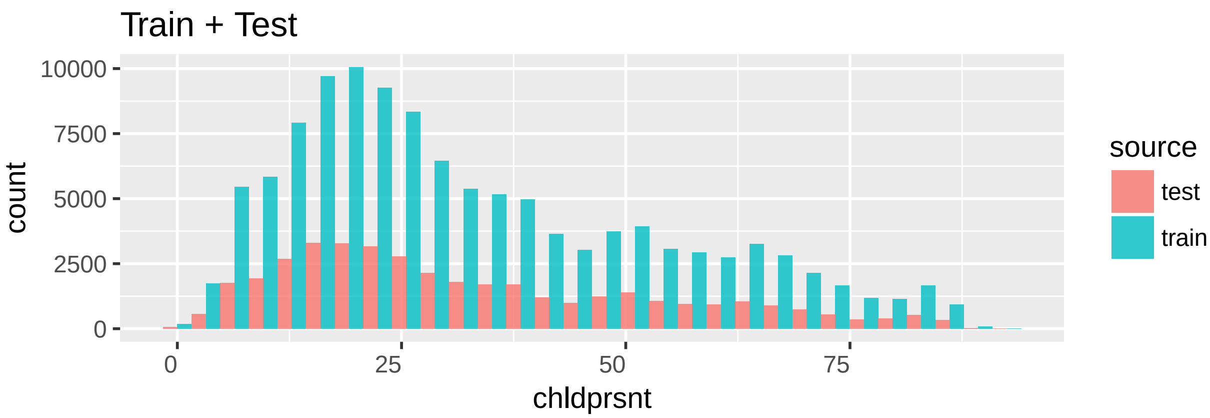


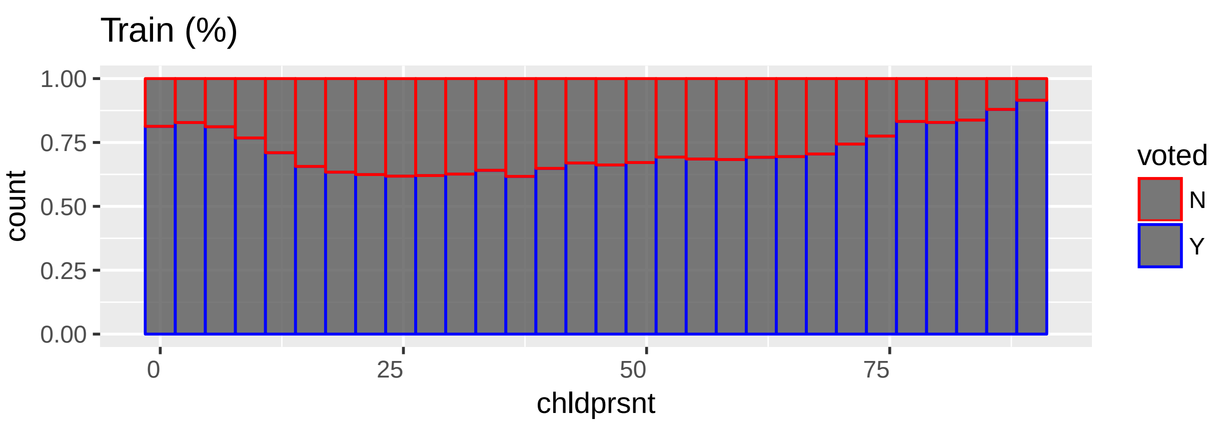
chldprsnt

(score for likelihood of having children at home) * medium score correlates with no

```
his_plot("chldprsnt", ncol = 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



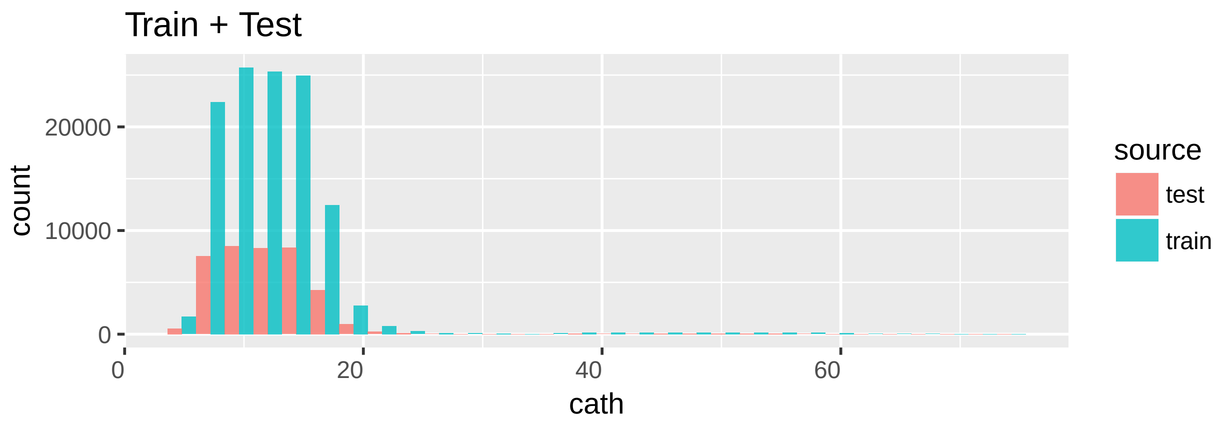


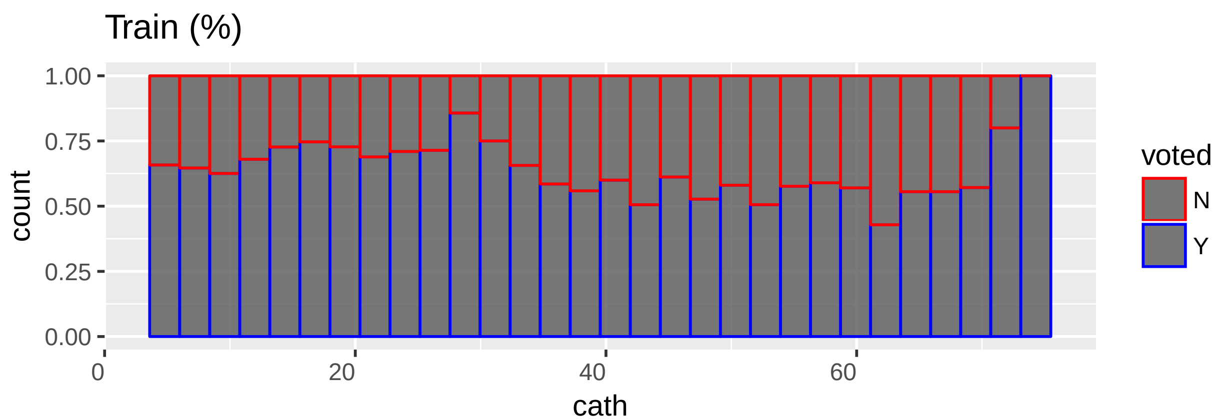
cath

(score for likelihood of being Catholic) * most people are not likely to be Catholic

```r
his_plot("cath", ncol = 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





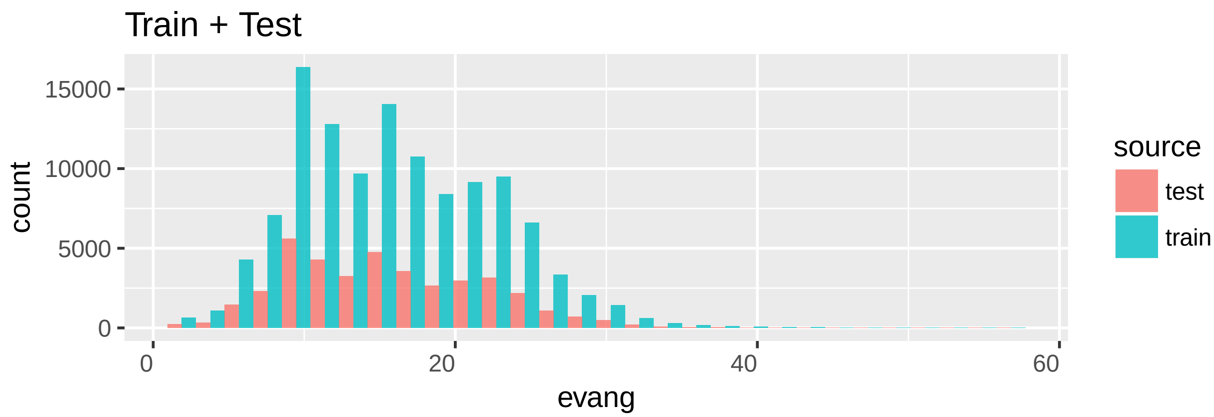
evang

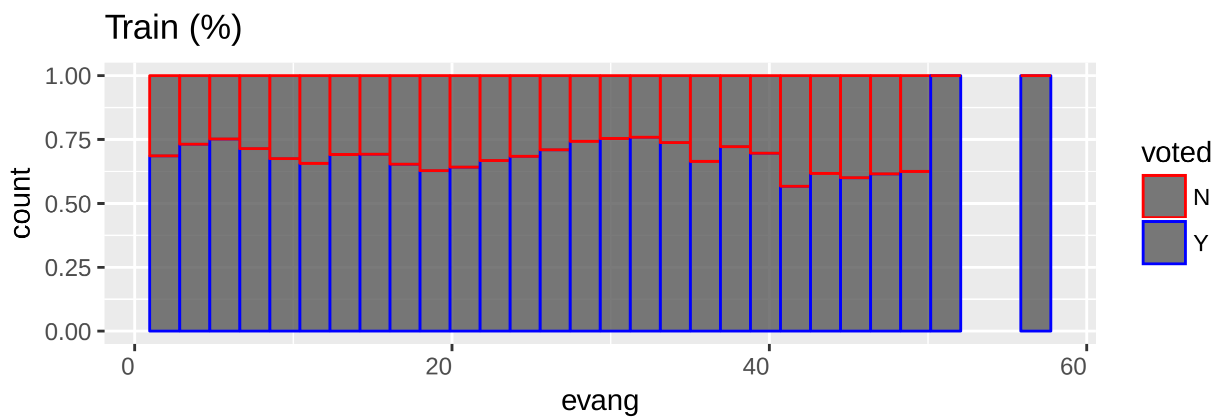
(score for likelihood of being Evangelical)

```
his_plot("evang", ncol = 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```



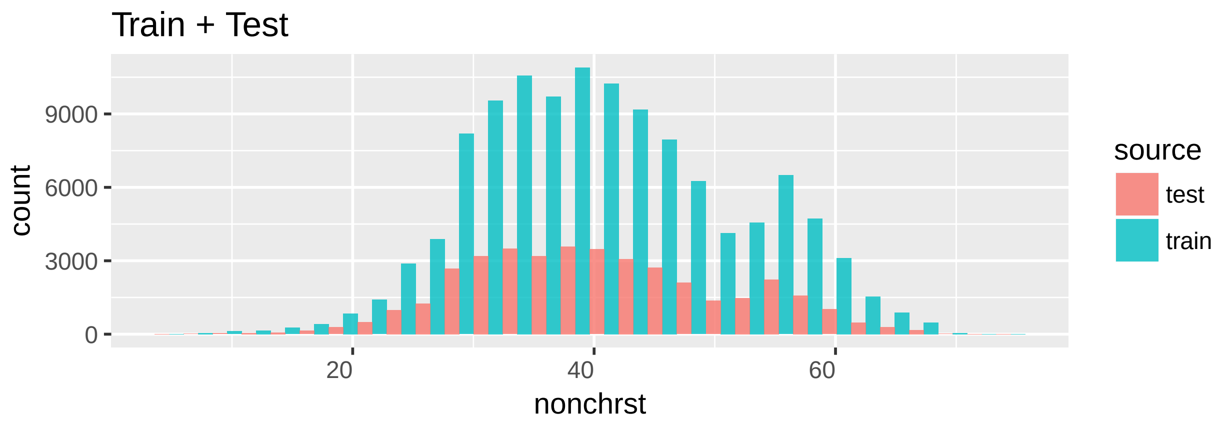


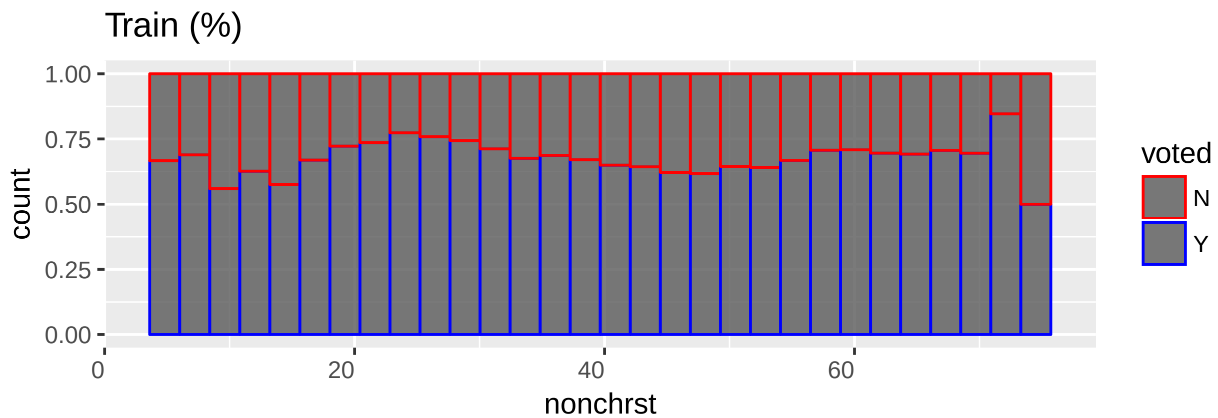
nonchrst

(score for likelihood of being non-Christian)

```
his_plot("nonchrst", ncol = 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



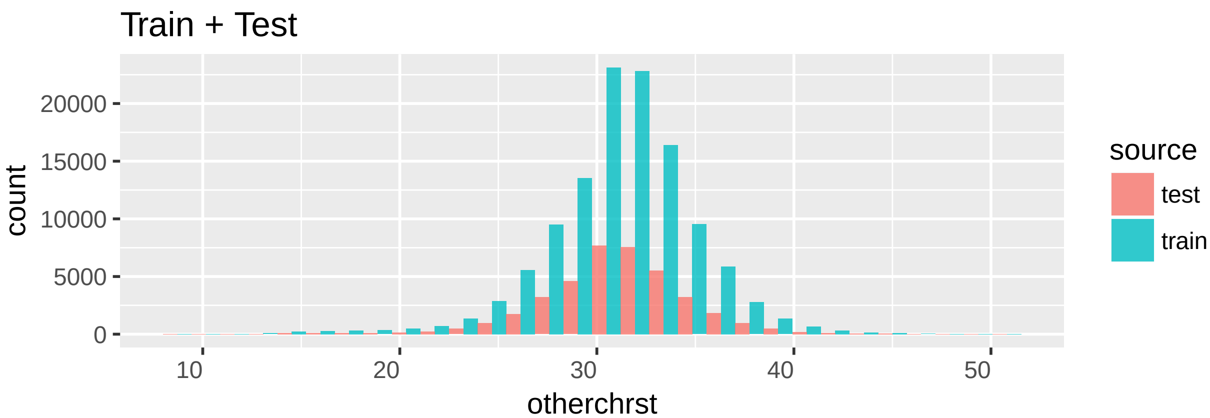


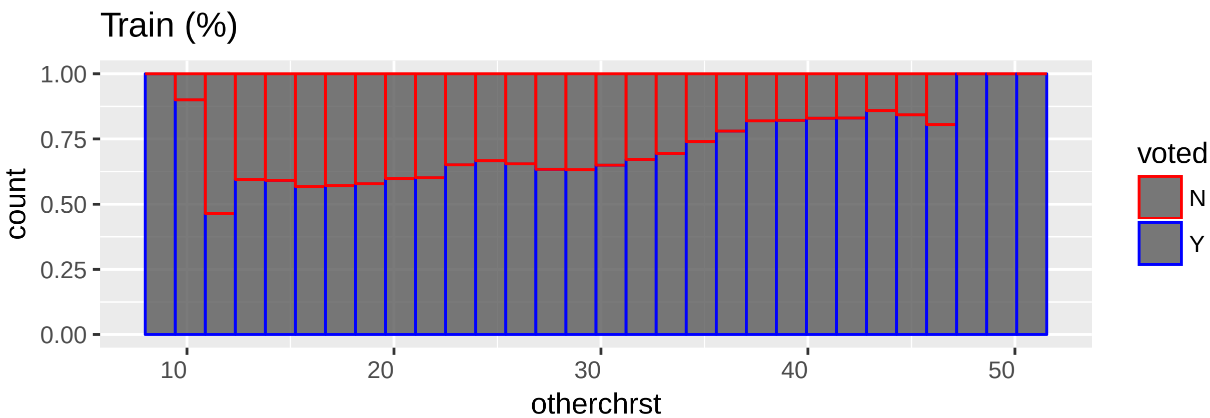
otherchrst

(score for likelihood of being another form of Christian)

```
his_plot("otherchrst", ncol = 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



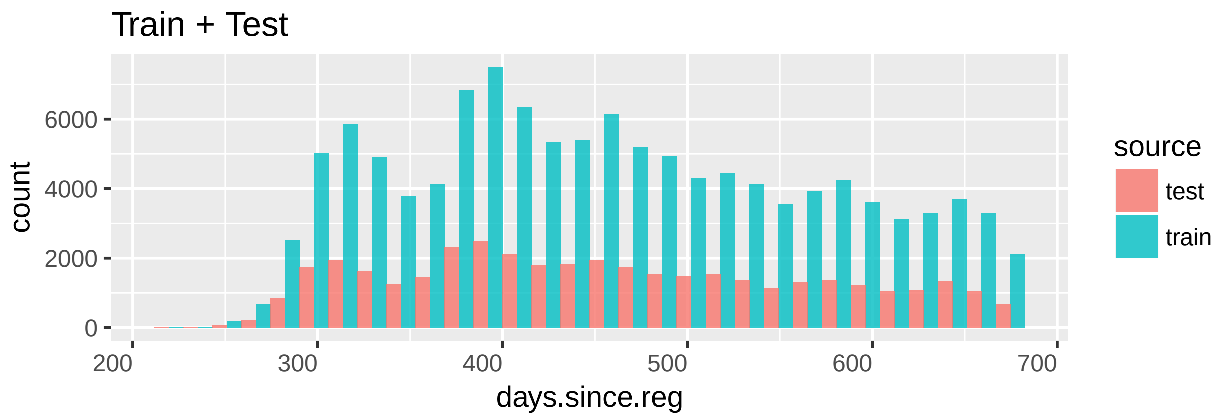


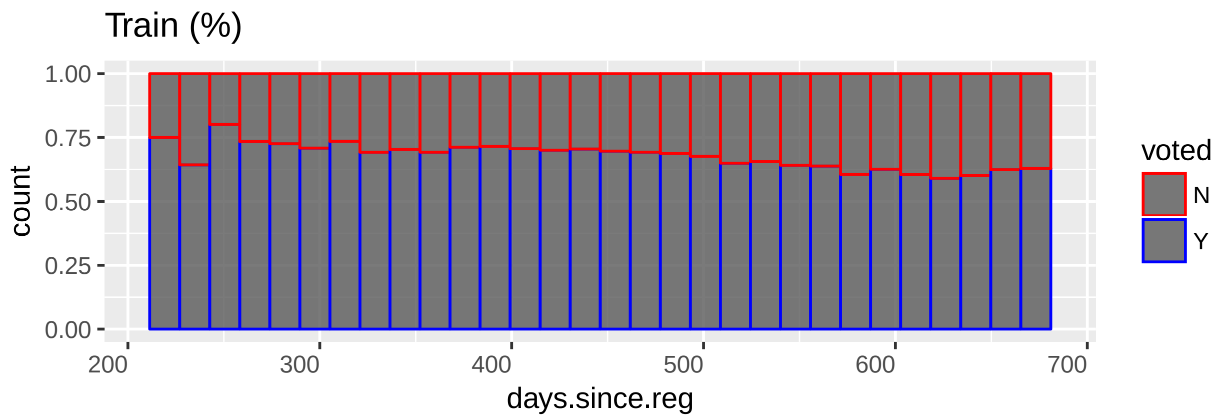
days.since.reg

(number of days since registered as a voter)

```r
his_plot("days.since.reg", ncol = 1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





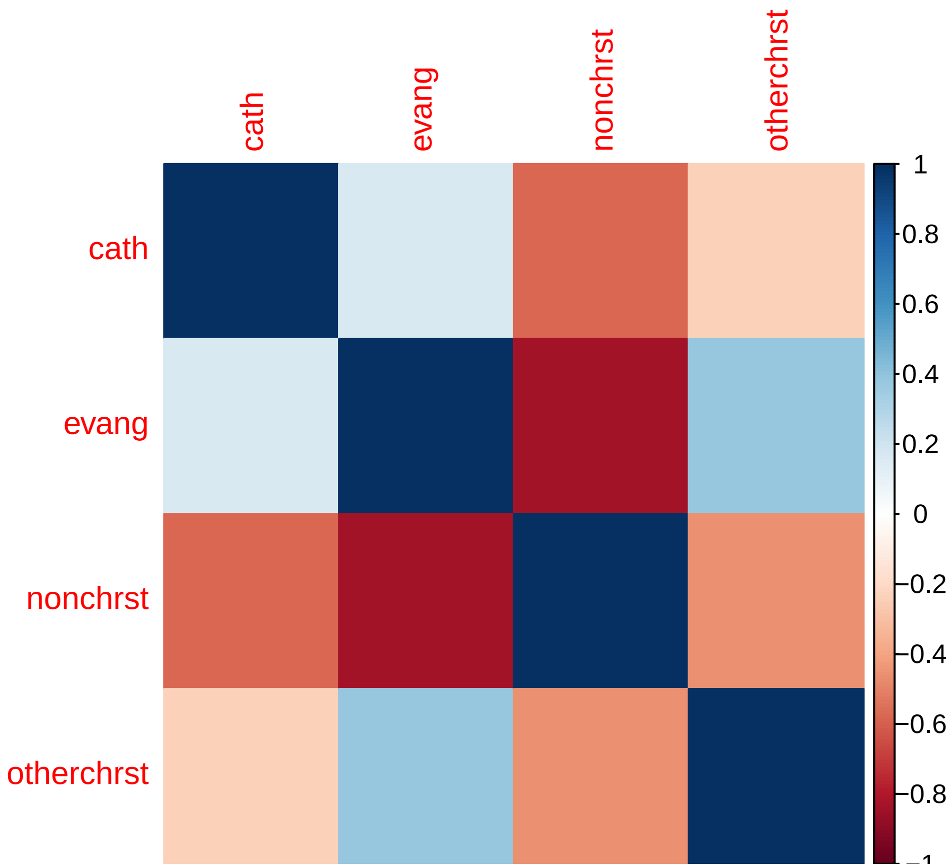
(not finished) 1.5 - check: colinearity

```r
colnames(train)
```

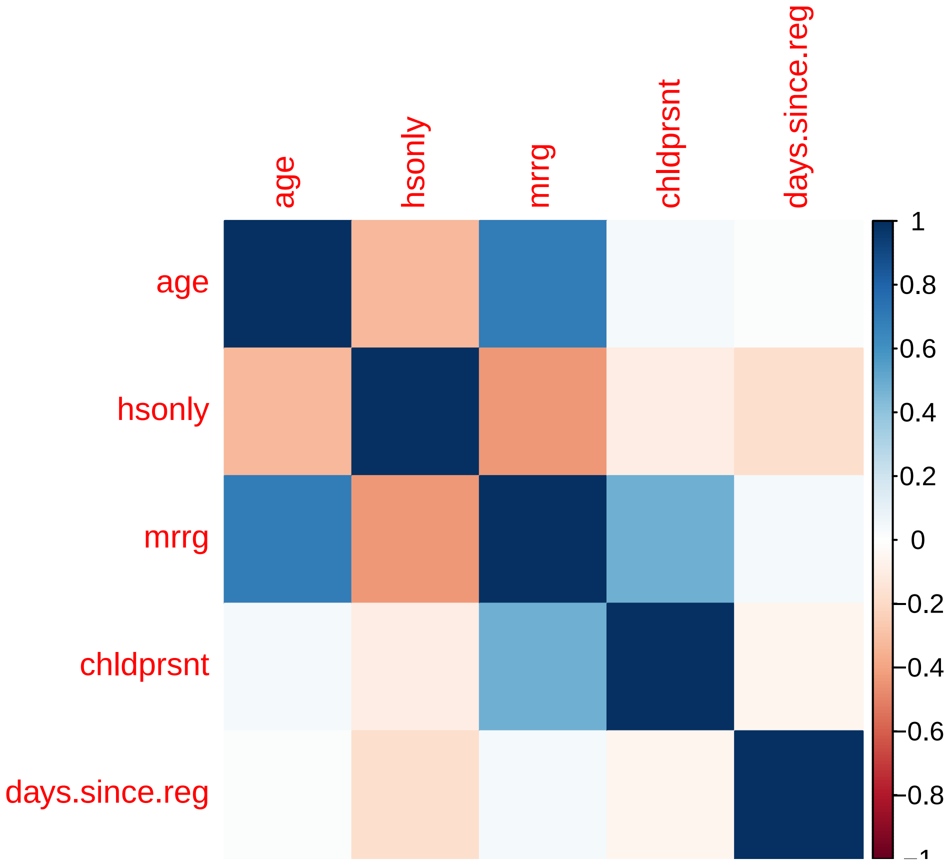
```
##  [1] "voted"          "gender"         "cd"             "hd"
##  [5] "age"            "dbdistance"     "vccdistance"    "party"
##  [9] "racename"       "hsonly"         "mrrg"           "chldprsnt"
## [13] "cath"           "evang"          "nonchrst"       "otherchrst"
## [17] "days.since.reg"
```

```r
## all the quantitative variables
#pairs(train[c("age", "hsonly", "mrrg", "chldprsnt", "days.since.reg")], cex = 0.01)
```

```r
## correlation plot
corrplot(cor(train[c("cath", "evang", "nonchrst", "otherchrst")]), method = "color")
```




```
corrplot(cor(train[c("age", "hsonly", "mrrg", "chldprsnt", "days.since.reg")]), method = "color")
```



(not finished) Part 2. Visualize clustering/dimension reduction

```
## exclude `voted' column and columns with NA
# train.temp <- train[!names(train) %in% c("voted", "dbdistance", "vccdistance")]

## scale column of quantitative variable
#for (i in 1:dim(train.temp)[2]){
#    if (class(train.temp[1,i]) != "factor"){
#        train.temp[i] = scale(train.temp[i])
#    }
#}

# distance metrix
# train.dist <- daisy(temp, metric = "gower")
```

Part 3. Baseline Classifier

```
## ignore the two columns with most NA
train.simple <- train[!names(train) %in% c("dbdistance", "vccdistance")]
## delete all NA values
train.simple <- na.omit(train.simple)
```

```
## function to calculate prediction accuracy from models
train.accu <- function(model, data){
    y.predict = predict(model, newdata = data, type = "response")
    y = ifelse(y.predict > 0.5, "Y", "N")
    table = table(train.simple$voted, y, dnn = c("data", "predict"))

    precision = table[2,2] / (table[1,2] + table[2,2])
    recall = table[2,2] / (table[2,1] + table[2,2])
    accu = (table[1,1] + table[2,2]) / sum(table)
    Fscore = 2* precision * recall / (precision + recall)

    return(list("Confusion Matrix" = table, "Precision" = precision, "Recall" = recall,
                "Accuracy" = accu, "F Score" = Fscore))
}
```

3.1 GAM

3.1.1 glm

```
train.lg <- glm(voted ~., data = train.simple, family = binomial)
summary(train.lg)
```

```
##
## Call:
## glm(formula = voted ~ ., family = binomial, data = train.simple)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5703  -1.1392   0.6294   0.8710   2.5728
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -1.908e+00  1.148e+01  -0.166 0.868055
## genderM                  -3.868e-01  1.389e-02 -27.856  < 2e-16 ***
## genderU                   1.637e-01  1.707e-01   0.959 0.337693
## cd2                      -3.638e-02  1.343e-01  -0.271 0.786545
## cd3                      -2.657e-01  1.430e-01  -1.858 0.063134 .
## cd4                      -2.292e-01  1.315e-01  -1.743 0.081370 .
## cd5                      -4.069e-01  1.452e-01  -2.803 0.005068 **
## cd6                       1.567e-01  9.849e-02   1.591 0.111503
## cd7                       2.431e-01  1.267e-01   1.919 0.054973 .
## hd2                       7.211e-01  7.752e-02   9.303  < 2e-16 ***
## hd3                       7.229e-02  9.450e-02   0.765 0.444263
## hd4                       6.438e-01  8.634e-02   7.457 8.87e-14 ***
## hd5                       3.135e-01  7.685e-02   4.079 4.53e-05 ***
## hd6                       4.919e-01  8.345e-02   5.894 3.77e-09 ***
## hd7                       1.584e-01  8.359e-02   1.895 0.058070 .
## hd8                       6.413e-01  8.063e-02   7.954 1.80e-15 ***
## hd9                       1.122e-01  7.954e-02   1.411 0.158357
## hd10                      5.833e-01  1.556e-01   3.748 0.000178 ***
## hd11                      2.211e-01  1.548e-01   1.428 0.153212
## hd12                      4.128e-01  1.559e-01   2.648 0.008108 **
```

```
## hd13                          2.564e-01  1.560e-01   1.643 0.100358
## hd14                          5.051e-01  1.656e-01   3.051 0.002281 **
## hd15                          3.200e-01  1.650e-01   1.939 0.052483 .
## hd16                          1.761e-01  1.658e-01   1.063 0.287916
## hd17                         -2.072e-01  1.648e-01  -1.257 0.208654
## hd18                          2.894e-01  1.645e-01   1.760 0.078437 .
## hd19                          6.475e-01  1.667e-01   3.884 0.000103 ***
## hd20                          3.562e-01  1.657e-01   2.149 0.031622 *
## hd21                         -1.149e-01  1.654e-01  -0.695 0.487161
## hd22                          3.192e-01  9.360e-02   3.410 0.000649 ***
## hd23                         -1.400e-01  1.518e-01  -0.922 0.356505
## hd24                          7.418e-02  1.530e-01   0.485 0.627680
## hd25                          4.077e-01  1.575e-01   2.588 0.009654 **
## hd26                          4.640e-01  1.624e-01   2.857 0.004282 **
## hd27                          1.215e-01  1.543e-01   0.787 0.431199
## hd28                         -7.925e-02  1.525e-01  -0.520 0.603192
## hd29                         -1.143e-01  1.545e-01  -0.740 0.459178
## hd30                         -2.492e-01  1.348e-01  -1.849 0.064448 .
## hd31                         -2.037e-01  1.511e-01  -1.348 0.177527
## hd32                         -2.395e-01  1.577e-01  -1.519 0.128751
## hd33                          3.135e-01  1.569e-01   1.998 0.045736 *
## hd34                         -3.486e-01  1.523e-01  -2.290 0.022048 *
## hd35                         -2.129e-01  1.536e-01  -1.386 0.165789
## hd36                         -2.255e-01  1.297e-01  -1.738 0.082163 .
## hd37                         -4.007e-02  1.301e-01  -0.308 0.758078
## hd38                          1.523e-01  1.346e-01   1.131 0.257876
## hd39                          5.410e-01  1.471e-01   3.678 0.000235 ***
## hd40                         -2.148e-01  1.300e-01  -1.652 0.098524 .
## hd41                         -4.666e-02  1.298e-01  -0.359 0.719227
## hd42                         -3.154e-01  1.321e-01  -2.387 0.016965 *
## hd43                          2.276e-01  1.327e-01   1.716 0.086246 .
## hd44                          3.967e-01  1.549e-01   2.560 0.010462 *
## hd45                          5.016e-01  1.556e-01   3.223 0.001267 **
## hd46                          9.357e-02  1.676e-01   0.558 0.576660
## hd47                         -8.795e-02  1.592e-01  -0.552 0.580697
## hd48                          2.929e-01  1.576e-01   1.858 0.063113 .
## hd49                          3.696e-01  1.561e-01   2.369 0.017850 *
## hd50                          5.365e-02  1.568e-01   0.342 0.732182
## hd51                          2.050e-01  1.582e-01   1.296 0.195024
## hd52                          2.498e-01  1.561e-01   1.600 0.109541
## hd53                          6.772e-01  1.567e-01   4.321 1.55e-05 ***
## hd54                          1.768e-01  1.675e-01   1.055 0.291198
## hd55                          3.572e-01  1.673e-01   2.135 0.032790 *
## hd56                          9.641e-02  1.303e-01   0.740 0.459362
## hd57                         -2.527e+00  1.756e-01 -14.389  < 2e-16 ***
## hd58                          2.657e-01  1.673e-01   1.588 0.112246
## hd59                         -3.210e-02  1.635e-01  -0.196 0.844355
## hd60                          2.994e-01  1.616e-01   1.853 0.063866 .
## hd61                         -1.911e-01  1.569e-01  -1.218 0.223221
## hd62                         -7.145e-01  1.686e-01  -4.238 2.26e-05 ***
## hd63                          2.955e-01  1.574e-01   1.877 0.060489 .
## hd64                          2.825e-01  1.606e-01   1.759 0.078497 .
## hd65                         -7.015e-01  1.600e-01  -4.384 1.17e-05 ***
## age                          -5.126e-03  7.941e-04  -6.455 1.08e-10 ***
```

```
## partyG                   -7.014e-02  1.007e-01  -0.697 0.485990    
## partyL                   -2.094e-01  5.318e-02  -3.937 8.26e-05 ***
## partyO                   -8.624e-01  1.006e-01  -8.572  < 2e-16 ***
## partyR                   -1.381e-01  2.028e-02  -6.810 9.76e-12 ***
## partyU                   -7.172e-01  1.631e-02 -43.981  < 2e-16 ***
## racenameCaucasian         6.786e-01  6.554e-02  10.354  < 2e-16 ***
## racenameCentral Asian     4.662e-01  2.356e-01   1.979 0.047847 *  
## racenameEast Asian        5.963e-01  8.186e-02   7.284 3.24e-13 ***
## racenameHispanic          3.800e-01  6.806e-02   5.584 2.36e-08 ***
## racenameJewish            8.394e-01  8.342e-02  10.062  < 2e-16 ***
## racenameMiddle Eastern    6.063e-01  1.181e-01   5.134 2.84e-07 ***
## racenameNative American   3.581e-01  1.518e-01   2.359 0.018306 *  
## racenamePacific Islander  1.309e-01  3.380e-01   0.387 0.698559    
## racenameSouth Asian       1.095e+00  1.250e-01   8.764  < 2e-16 ***
## racenameUncoded           6.901e-01  7.096e-02   9.726  < 2e-16 ***
## hsonly                    7.964e-03  6.826e-04  11.666  < 2e-16 ***
## mrrg                      2.042e-02  4.658e-04  43.845  < 2e-16 ***
## chldprsnt                -1.150e-02  4.815e-04 -23.878  < 2e-16 ***
## cath                      3.273e-02  1.148e-01   0.285 0.775639    
## evang                     2.742e-03  1.148e-01   0.024 0.980953    
## nonchrst                  2.047e-02  1.148e-01   0.178 0.858543    
## otherchrst                5.166e-02  1.148e-01   0.450 0.652784    
## days.since.reg           -1.804e-03  6.667e-05 -27.055  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 148964  on 118526  degrees of freedom
## Residual deviance: 134340  on 118430  degrees of freedom
## AIC: 134534
## 
## Number of Fisher Scoring iterations: 4
```

```
train.accu(train.lg)
```

```
## $`Confusion Matrix`
##     predict
## data     N     Y
##    N 10026 28144
##    Y  6434 73923
## 
## $Precision
## [1] 0.7242596
## 
## $Recall
## [1] 0.9199323
## 
## $Accuracy
## [1] 0.708269
## 
## $`F Score`
## [1] 0.8104526
```

3.1.2 gam

```r
colnames(train)
```

```
##  [1] "voted"          "gender"         "cd"             "hd"            
##  [5] "age"            "dbdistance"     "vccdistance"    "party"         
##  [9] "racename"       "hsonly"         "mrrg"           "chldprsnt"     
## [13] "cath"           "evang"          "nonchrst"       "otherchrst"    
## [17] "days.since.reg"
```

```r
# non-linear for all the quantitative variables
train.gam <- gam(voted ~ s(age) + s(hsonly) + s(mrrg) + s(chldprsnt) +
                     s(cath) + s(evang) + s(nonchrst) + s(otherchrst) +
                     s(days.since.reg) + gender + cd + hd,
                 data = train.simple, family = binomial)

summary(train.gam)
```

```
## 
## Call: gam(formula = voted ~ s(age) + s(hsonly) + s(mrrg) + s(chldprsnt) + 
##     s(cath) + s(evang) + s(nonchrst) + s(otherchrst) + s(days.since.reg) + 
##     gender + cd + hd, family = binomial, data = train.simple)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max 
## -2.7035 -1.1911  0.6600  0.8888  2.5708 
## 
## (Dispersion Parameter for binomial family taken to be 1)
## 
##     Null Deviance: 148963.5 on 118526 degrees of freedom
## Residual Deviance: 135836.9 on 118418 degrees of freedom
## AIC: 136054.9 
## 
## Number of Local Scoring Iterations: 5 
## 
## Anova for Parametric Effects
##                       Df Sum Sq Mean Sq   F value    Pr(>F)    
## s(age)                 1   1123 1123.01 1122.0858 < 2.2e-16 ***
## s(hsonly)              1   1071 1070.87 1069.9847 < 2.2e-16 ***
## s(mrrg)                1    947  946.95  946.1705 < 2.2e-16 ***
## s(chldprsnt)           1    102  101.82  101.7405 < 2.2e-16 ***
## s(cath)                1     38   38.10   38.0658 6.862e-10 ***
## s(evang)               1    110  109.66  109.5725 < 2.2e-16 ***
## s(nonchrst)            1    144  144.47  144.3522 < 2.2e-16 ***
## s(otherchrst)          1      0    0.12    0.1171    0.7322    
## s(days.since.reg)      1   1156 1156.33 1155.3783 < 2.2e-16 ***
## gender                 2    654  327.02  326.7451 < 2.2e-16 ***
## cd                     6    688  114.61  114.5112 < 2.2e-16 ***
## hd                    64   2049   32.02   31.9923 < 2.2e-16 ***
## Residuals         118418 118516    1.00                        
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Anova for Nonparametric Effects
##                   Npar Df Npar Chisq    P(Chi)    
```

```
## (Intercept)
## s(age)                  3      94.97 < 2.2e-16 ***
## s(hsonly)               3      93.17 < 2.2e-16 ***
## s(mrrg)                 3     318.16 < 2.2e-16 ***
## s(chldprsnt)            3      50.95 5.010e-11 ***
## s(cath)                 3     355.29 < 2.2e-16 ***
## s(evang)                3      42.22 3.598e-09 ***
## s(nonchrst)             3     154.09 < 2.2e-16 ***
## s(otherchrst)           3      10.29   0.01627 *
## s(days.since.reg)       3      29.47 1.787e-06 ***
## gender
## cd
## hd
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
train.accu(train.gam)
```

```
## $`Confusion Matrix`
##     predict
## data     N     Y
##    N  8002 30168
##    Y  4988 75369
##
## $Precision
## [1] 0.7141476
##
## $Recall
## [1] 0.937927
##
## $Accuracy
## [1] 0.7033925
##
## $`F Score`
## [1] 0.8108815
```

```r
## convert all categorical variables to dummy coding

#library(dummies)
#train.dummy <- dummy.data.frame(data = train.simple,
#                                 names = c("gender", "cd", "hd", "party", "racename"))
```

3.2 random forest

```r
#library(randomForest)

#train.rf = randomForest(data = train.dummy, voted ~., importance=TRUE)
#print(train.rf)
```

3.3. SVM

```r
#library(e1071)
#train.svm <- svm(voted ~., data = train.dummy, kernel = "radial")
```