1992 U.S. Presidential election

Ali Tarek Maher Ibrahim Ali Seada and Paul Lovis Maximilian Trüstedt

24 6 2021

Read the data into R environment

```
library(pacman)
p_load(ggplot2,   # reportable graphs
       cowplot,   # arranges ggplot graphs nicely
       stargazer, # nice tables
       glmnet, # for regularization (lasso, ridge, elastic net)
       caret,       #  splitting the data and more
       rpart,       #  building decision trees
       rpart.plot,
       pROC)      # ROC AUC
rm(list=ls())
vote<-read.csv("vote92.csv", sep = ",", header = T,stringsAsFactors = T)
str(vote)
```

```
## 'data.frame':    909 obs. of  10 variables:
##  $ X          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ vote       : Factor w/ 3 levels "Bush","Clinton",..: 1 1 2 1 2 2 3 1 1 3 ...
##  $ dem        : int  0 0 1 0 0 1 1 0 0 0 ...
##  $ rep        : int  1 1 0 1 0 0 0 1 1 1 ...
##  $ female     : int  1 1 1 0 1 1 1 0 1 0 ...
##  $ persfinance: int  1 0 0 0 0 -1 1 0 1 0 ...
##  $ natlecon   : int  0 -1 -1 -1 -1 -1 0 0 -1 0 ...
##  $ clintondis : num  4.0804 4.0804 1.0404 0.0004 0.9604 ...
##  $ bushdis    : num  0.102 0.102 1.742 5.382 11.022 ...
##  $ perotdis   : num  0.26 0.26 0.24 2.22 6.2 ...
```

```
summary(vote)
```

```
##        X             vote          dem              rep             female
##  Min.   :  1   Bush   :310   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:228   Clinton:416   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :455   Perot  :183   Median :0.0000   Median :0.0000   Median :0.0000
##  Mean   :455                 Mean   :0.4884   Mean   :0.4301   Mean   :0.4752
##  3rd Qu.:682                 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :909                 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##   persfinance           natlecon         clintondis         bushdis
##  Min.   :-1.000000   Min.   :-1.0000   Min.   : 0.0004   Min.   : 0.1024
##  1st Qu.:-1.000000   1st Qu.:-1.0000   1st Qu.: 0.9604   1st Qu.: 0.4624
##  Median : 0.000000   Median :-1.0000   Median : 1.0404   Median : 1.7424
##  Mean   :-0.009901   Mean   :-0.6722   Mean   : 3.5062   Mean   : 3.3793
##  3rd Qu.: 1.000000   3rd Qu.: 0.0000   3rd Qu.: 4.0804   3rd Qu.: 5.3824
##  Max.   : 1.000000   Max.   : 1.0000   Max.   :16.1600   Max.   :18.6620
```

```
##     perotdis
##  Min.   : 0.2401
##  1st Qu.: 0.2401
##  Median : 2.2201
##  Mean   : 2.1710
##  3rd Qu.: 2.2801
##  Max.   :12.1800
```

```
# ??remove cowplot, stargazer & pROC
```

Preprocess the data, preparing it for the modeling

```
vote$vote_num <- as.numeric(vote$vote)
vote$dem<-as.factor(vote$dem)
vote$rep<-as.factor(vote$rep)
vote$female<-as.factor(vote$female)
vote$persfinance<-as.factor(vote$persfinance)
vote$natlecon<-as.factor(vote$natlecon)
vote$polID<-as.factor((as.numeric(vote$dem)-1)+(as.numeric(vote$rep)*2-1))
str(vote)
```

```
## 'data.frame':    909 obs. of  12 variables:
##  $ X          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ vote       : Factor w/ 3 levels "Bush","Clinton",..: 1 1 2 1 2 2 3 1 1 3 ...
##  $ dem        : Factor w/ 2 levels "0","1": 1 1 2 1 1 2 2 1 1 1 ...
##  $ rep        : Factor w/ 2 levels "0","1": 2 2 1 2 1 1 1 2 2 2 ...
##  $ female     : Factor w/ 2 levels "0","1": 2 2 2 1 2 2 2 1 2 1 ...
##  $ persfinance: Factor w/ 3 levels "-1","0","1": 3 2 2 2 2 1 3 2 3 2 ...
##  $ natlecon   : Factor w/ 3 levels "-1","0","1": 2 1 1 1 1 1 2 2 1 2 ...
##  $ clintondis : num  4.0804 4.0804 1.0404 0.0004 0.9604 ...
##  $ bushdis    : num  0.102 0.102 1.742 5.382 11.022 ...
##  $ perotdis   : num  0.26 0.26 0.24 2.22 6.2 ...
##  $ vote_num   : num  1 1 2 1 2 2 3 1 1 3 ...
##  $ polID      : Factor w/ 3 levels "1","2","3": 3 3 2 3 1 2 2 3 3 3 ...
```

We decided to change some of the numeric variables to factors, because it makes more sense to have them as categorical than as numeric variables. Also this way, we can see, that there are no problems with the categorical variables regarding wrong values, because all provided levels are described by the given data set definition. Additionally we create a categorical variable called polID to summarize which political party the respondent is identifying himself with.

- treat missing values

```
colSums(is.na(vote))
```

```
##           X        vote         dem         rep      female persfinance
##           0           0           0           0           0           0
##    natlecon  clintondis     bushdis    perotdis    vote_num       polID
##           0           0           0           0           0           0
```

There are no missing values in this data set. No NAs, as well as data, that could otherwise be identified as missing.

- handle sparse classes of categorical predictors

```
table(vote$vote) # !!make these tables pretty (bar plot coloured)
```

```
##
```

```
##    Bush Clinton   Perot
##     310     416     183
```

```
table(vote$dem)
```

```
##
##   0   1
## 465 444
```

```
table(vote$rep)
```

```
##
##   0   1
## 518 391
```

```
table(vote$female)
```

```
##
##   0   1
## 477 432
```

```
table(vote$persfinance)
```

```
##
##  -1   0   1
## 308 302 299
```

```
table(vote$natlecon)
```

```
##
##  -1   0   1
## 656 208  45
```

```
vote$natlecon[vote$natlecon==1]<-0
vote$natlecon[vote$natlecon==-1]<-1
vote$natlecon=droplevels(vote$natlecon)
table(vote$natlecon)
```

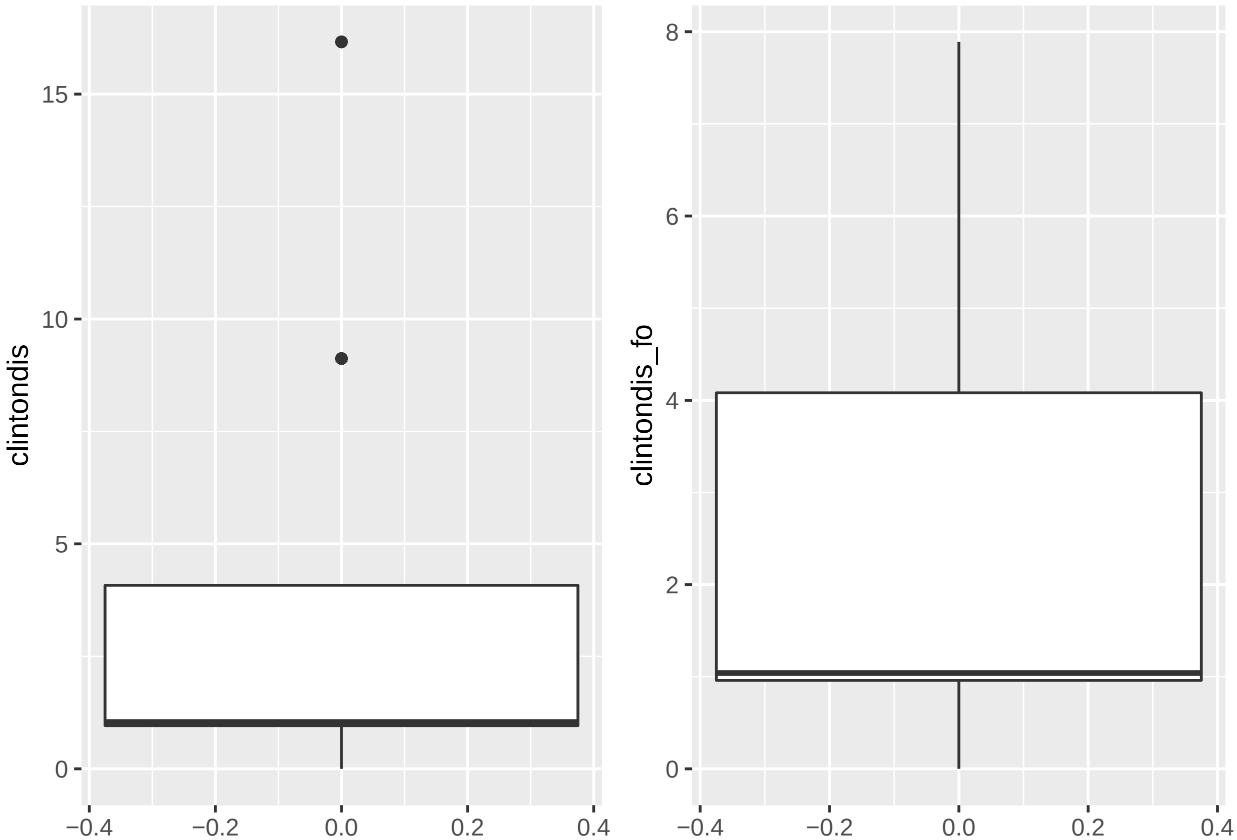
```
##
##   0   1
## 253 656
```

We leave everything as is except for natlecon which has a sparse class regarding the level 1. As solution we combine 0 and 1 as the level 0, meaning national economic conditions have gotten better or stayed the same over the last 12 months. Level -1 gets changed to 1 as well which now means that conditions have gotten better. The change from -1 to 1 is executed just because it is more common to have levels 0 and 1 instead of 0 and -1.

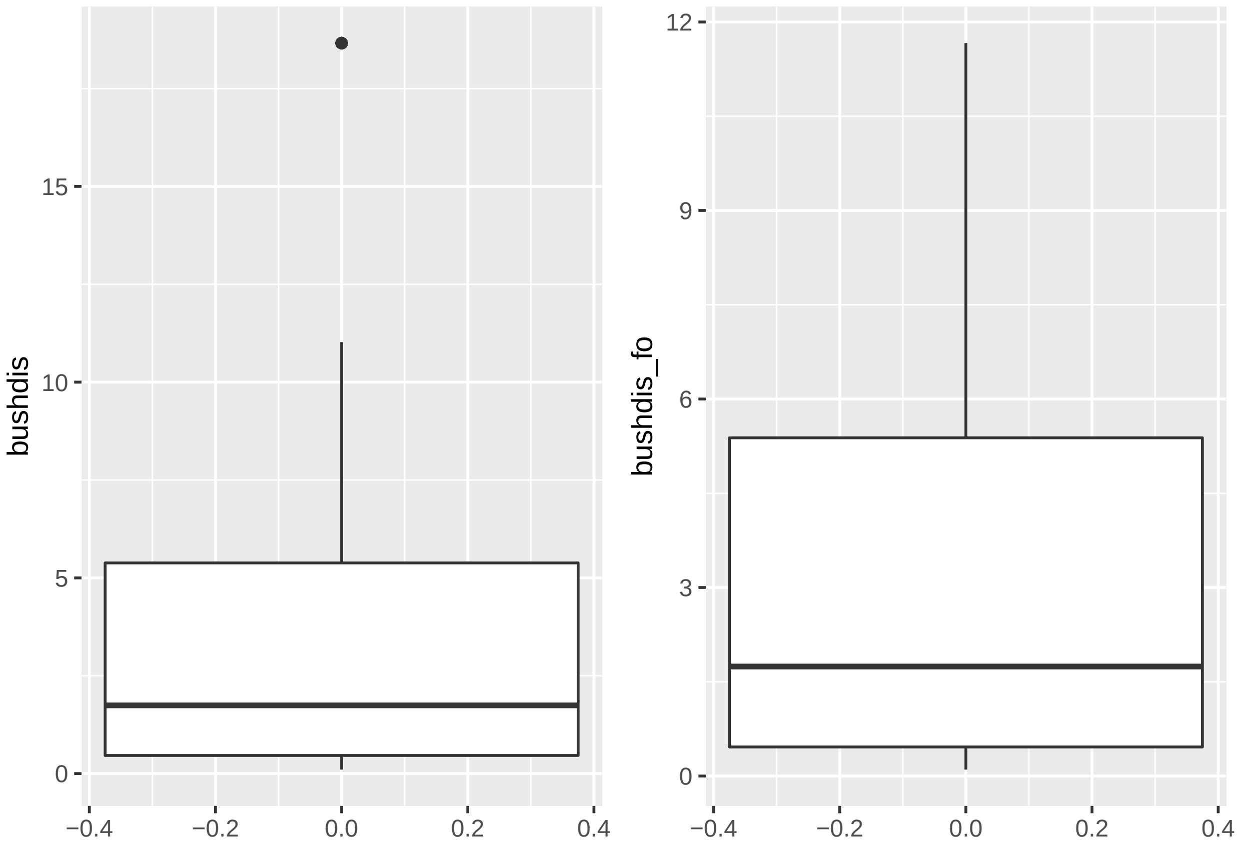
- take care of outliers, treat the skewed distributions and create new features

```
zScores<-function(var) {
    mu<-mean(var)
    sd<-sd(var)
    return((var-mu)/sd)
}

# treating clintondis
tp1<-ggplot(vote,aes(clintondis))+geom_boxplot()+coord_flip()
vote$clintondis_fo<-vote$clintondis
vote$clintondis_fo[zScores(vote$clintondis_fo)>1]<-
```

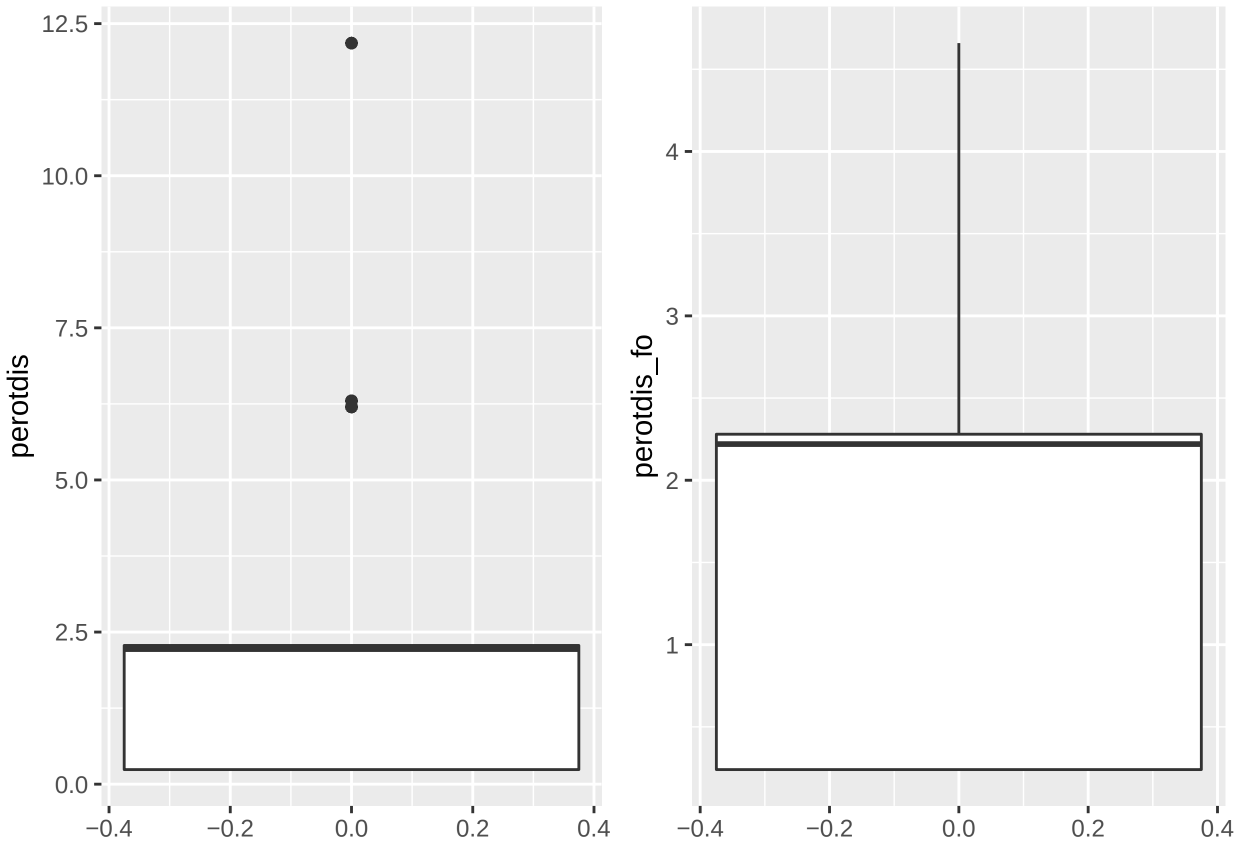
```
    round(mean(vote$clintondis_fo))+sd(vote$clintondis_fo)
tp2<-ggplot(vote,aes(clintondis_fo))+geom_boxplot()+coord_flip()
plot_grid(tp1,tp2,ncol=2)
```




```
# treating bushdis
tp1<-ggplot(vote,aes(bushdis))+geom_boxplot()+coord_flip()
vote$bushdis_fo<-vote$bushdis
vote$bushdis_fo[zScores(vote$bushdis_fo)>2]<-
    round(mean(vote$bushdis_fo))+2*sd(vote$bushdis_fo)
tp2<-ggplot(vote,aes(bushdis_fo))+geom_boxplot()+coord_flip()
plot_grid(tp1,tp2,ncol=2)
```




```
# treating perotdis
tp1<-ggplot(vote,aes(perotdis))+geom_boxplot()+coord_flip()
vote$perotdis_fo<-vote$perotdis
vote$perotdis_fo[zScores(vote$perotdis_fo)>1]<-
    round(mean(vote$perotdis_fo))+sd(vote$perotdis_fo)
tp2<-ggplot(vote,aes(perotdis_fo))+geom_boxplot()+coord_flip()
plot_grid(tp1,tp2,ncol=2)
```

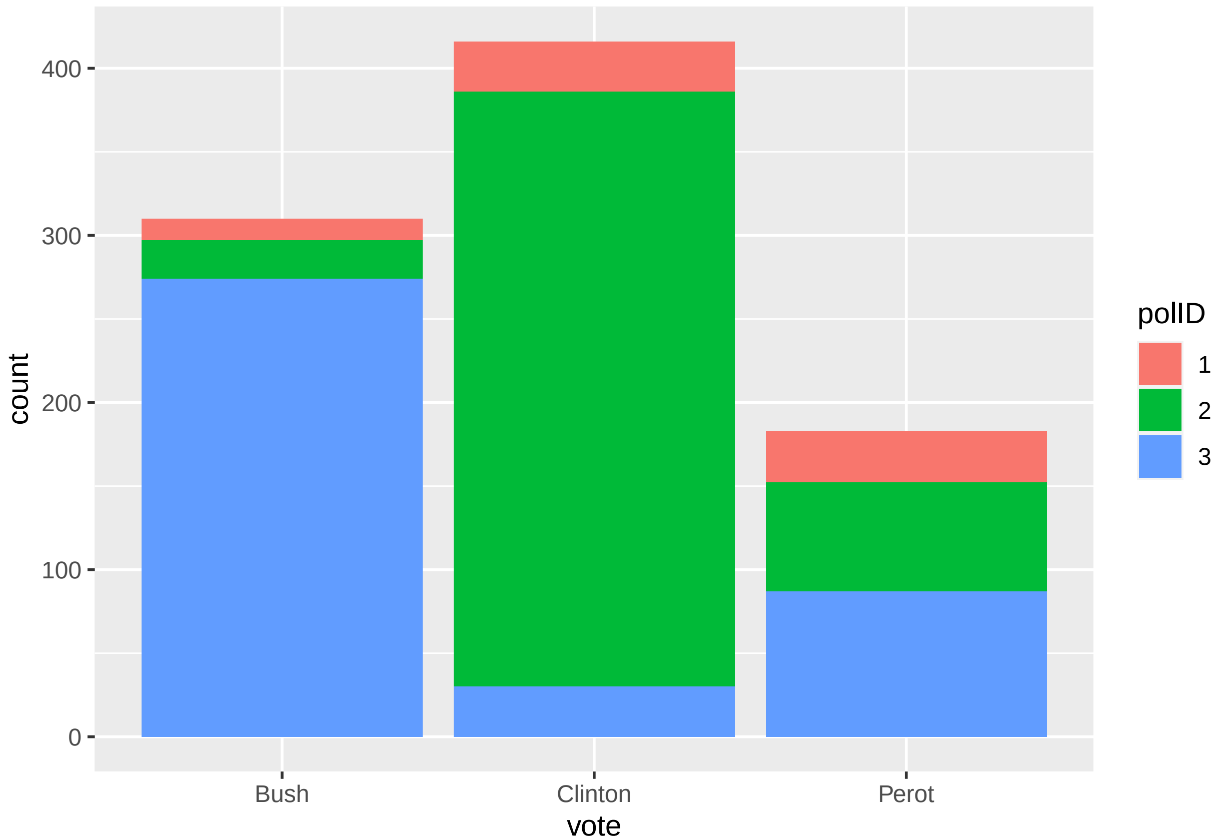



```
# !!make this shit beautiful
# !!reduce the outlier fixes with function
```

There are a few outliers in the variables clintondis, bushdis and perotdis. We fix those outliers and save the fixed data in the variables called [original_var_name]_fo. The ending "fo" is derived from "fixed outliers".

- explore the relationships between predictors and the target

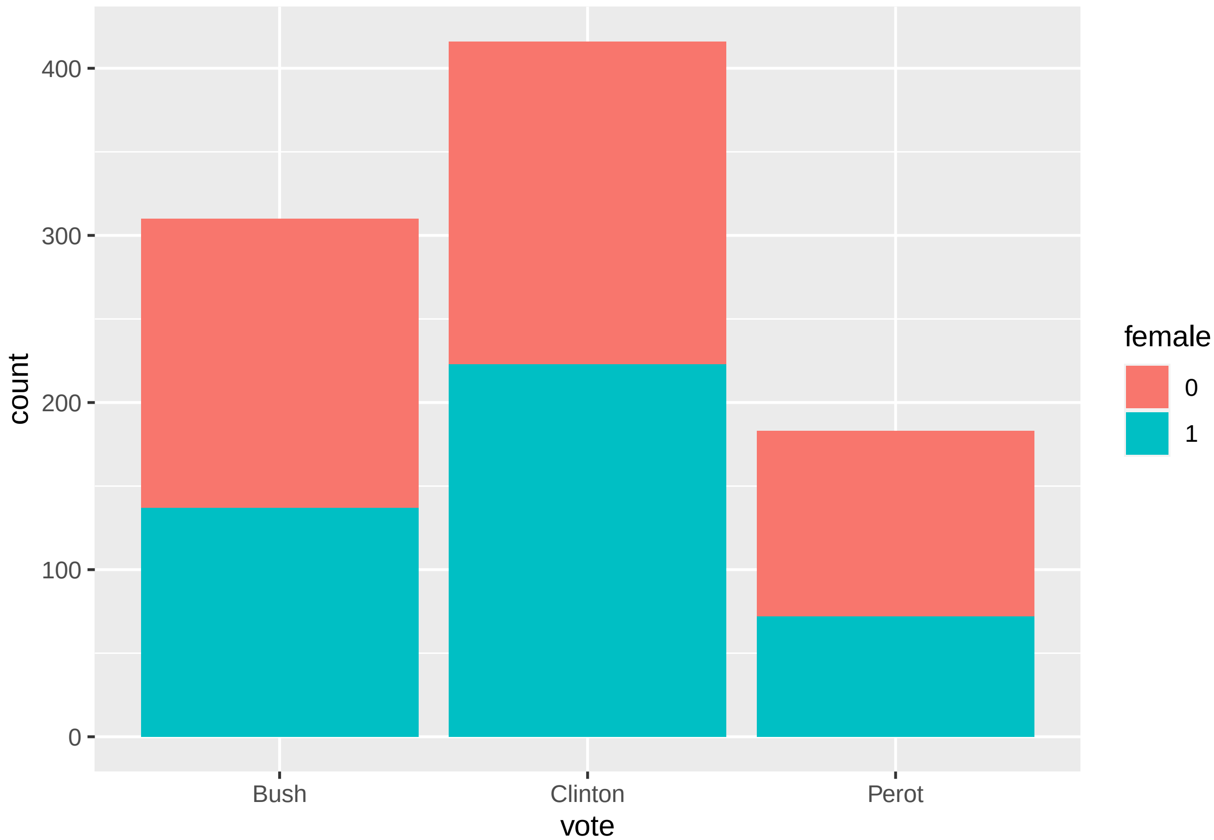
```
ggplot(vote,aes(vote,fill=polID))+geom_bar() # !!fix colours and descriptions
```




```
# ??maybe plot respective predictions by alignment next to actual vote
# !!add percentiles to those splitted barplots somehow
# !!show amounts
```

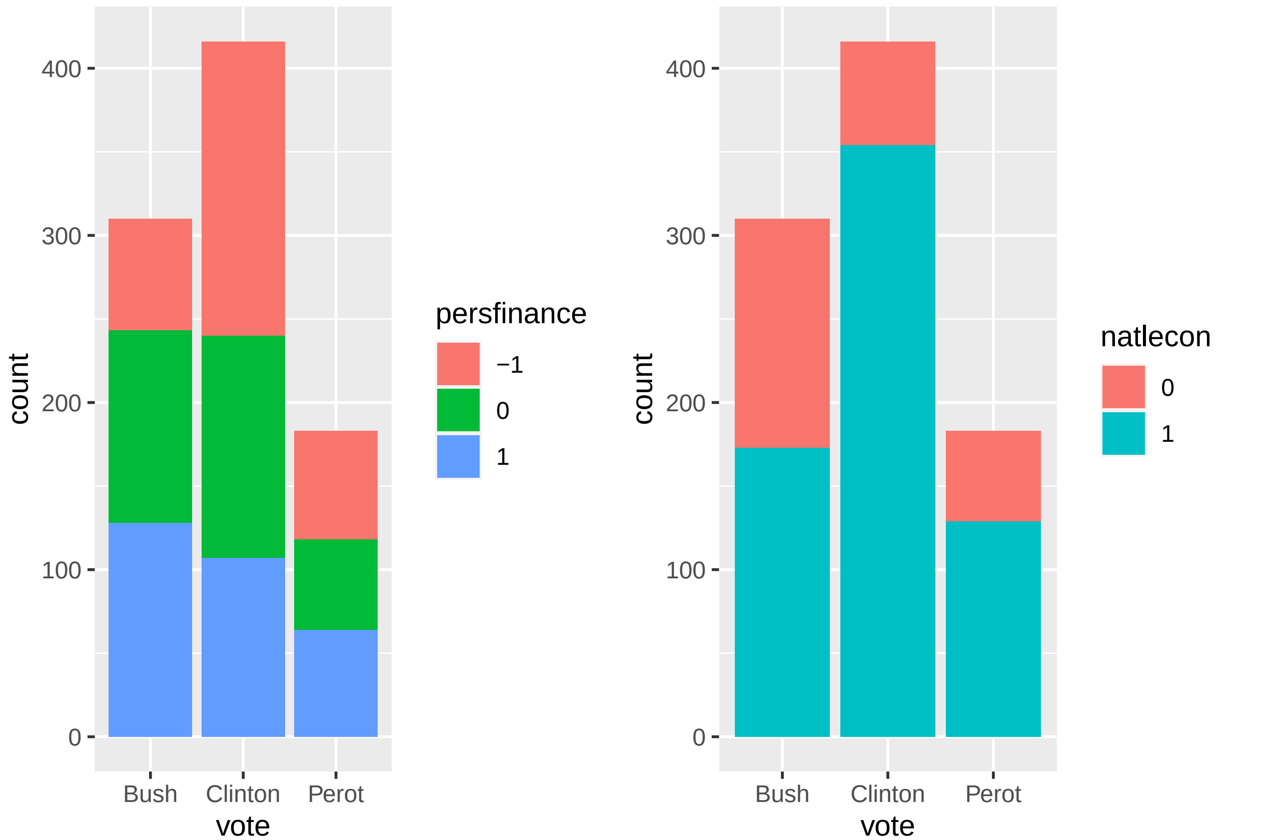
Looking at the votes for Perot, most prominently visible is, that half of his votes came from republican voters, which Bush lost. Also Bush got least votes from voters who didn't clearly align with either the democratic or the republican party. Those undecided who didn't vote for Bush about equally voted for Clinton and Perot. Not only did more undecided voters vote for Clinton instead of Bush, but also more democrats voted for Clinton than republicans for Bush. To top it off, even more republicans voted for the democratic as for the republican party. Though it must be noted that more respondents reported aligning with the democratic party in the first place.

```
ggplot(vote,aes(vote,fill=female))+geom_bar() # !!show percentages
```



Both Bush and Perot had significantly more male than female voters, though that could also because more respondents in this data set are male than female. Regardless of the imbalance of genders among the respondents though, Clinton had not only the highest percentage of female voters but also received about balanced votes by gender.

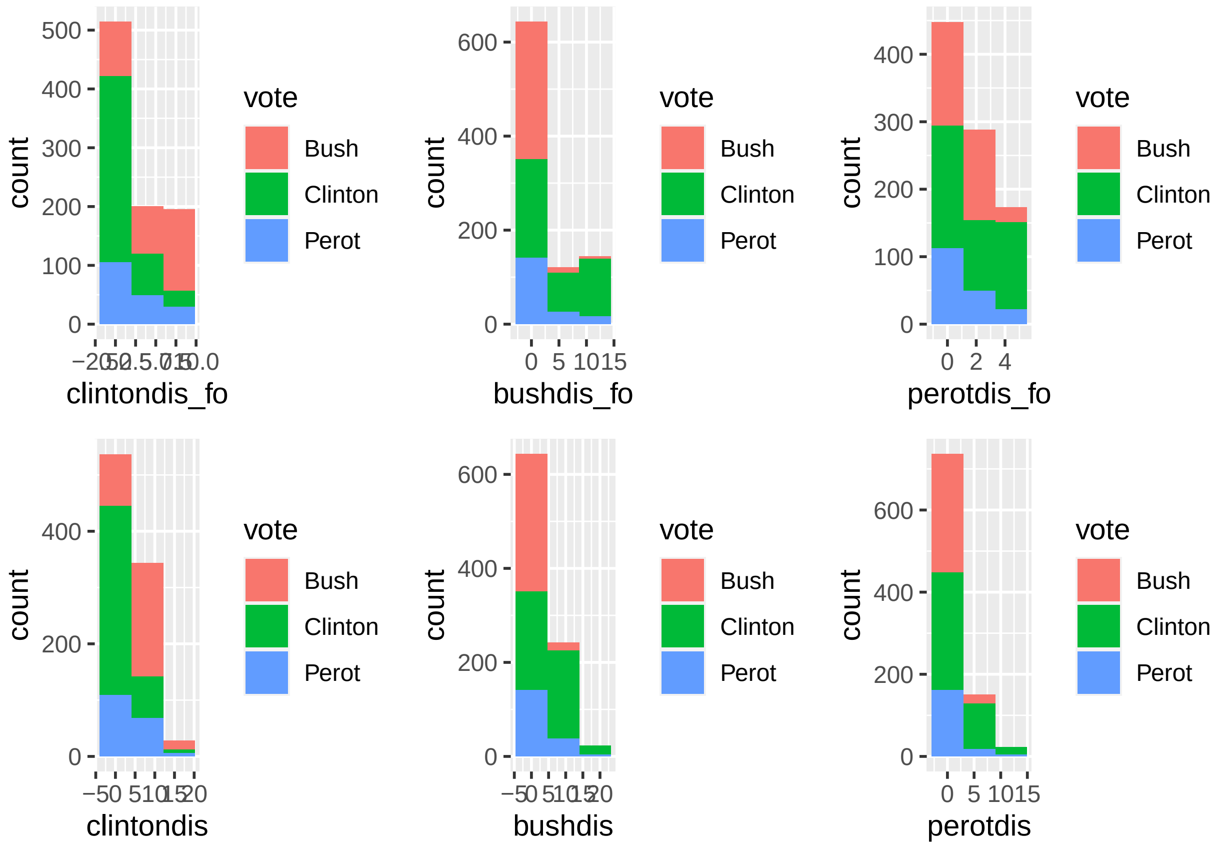
```
p1<-ggplot(vote,aes(vote,fill=persfinance))+geom_bar()
p2<-ggplot(vote,aes(vote,fill=natlecon))+geom_bar()
plot_grid(p1,p2,ncol=2)
```




```
# !!fix repeating barplot by creating a more diverse visual representation
```

Most of the people, who felt like their own financial situation had worsened, voted for Clinton, while people that had a positive change regarding their personal finances voted for Bush. While even most of those, who felt no personal financial change voted for Clinton, more who had a change for the worst voted for Clinton than for Bush and Perot combined. Perots source of votes are mostly balanced regarding the respondents personal financial situation, but those who felt like the national economy has gotten worse, were much more likely to vote for Perot, than those who felt no change or an increase in national wealth. That being said, those who felt the national economy was getting worse, seem to overwhelmingly have voted for Clinton. Even despite the mentioned bias regarding Perot, Clinton received about as much votes from those who felt a negative change in the national economy as Perot and Bush together, but more people, that were comfortable with the change of the national economy voted for Bush, than for Clinton and Perot combined.

```
p1<-ggplot(vote,aes(clintondis_fo,fill=vote))+geom_histogram(bins=3)
p2<-ggplot(vote,aes(bushdis_fo,fill=vote))+geom_histogram(bins=3)
p3<-ggplot(vote,aes(perotdis_fo,fill=vote))+geom_histogram(bins=3)
p4<-ggplot(vote,aes(clintondis,fill=vote))+geom_histogram(bins=3)
p5<-ggplot(vote,aes(bushdis,fill=vote))+geom_histogram(bins=3)
p6<-ggplot(vote,aes(perotdis,fill=vote))+geom_histogram(bins=3)
plot_grid(p1,p2,p3,p4,p5,p6,ncol=3)
```




```
# !!fix legend print and axis descriptions, also colours
```

Those who ideologically identify themselves most with the candidate of the democratic party unsurprisingly overwhelmingly voted for Clinton. Moving away from the optimal ideological alignment of the respondent with Clinton, more and more voters mostly chose Bush over Clinton, as well as Perot.
Voters, who ideologically identify more with the candidate of the republican party did not vote as decisively for Bush as democrats for Clinton. Not only did more voters, aligning with Bush, vote for Perot but also much more for Clinton than the voters, aligning with Clinton, for Bush. Again, moving away from the ideological alignment of the voters with Bush, this trend is much more visible, voters absolute decisively voting Clinton instead of anyone else. But recognizing those majorities, we have to keep in mind that most of the respondents as a whole did vote Clinton, so majorities in favour of Clinton are to be expected.
The same trend is visible in the histogram regarding ideological alignment with Perot. While those, who align more with Perots ideology tend to vote Bush a little more often than Clinton, moving away from ideological alignment we again see a strong tendency towards Clinton. The small trend towards Bush by voters aligning with Perot is not surprising, because Perot later was considered a republican, but in 92 disagreed with Bush on some things, most prominently regarding war among other topics. But even those who align with Perot the most, did not end up voting for Perot. This could be because a vote for Perot would not determine presidency, because in the american voting system only two presidential candidates can be voted for and Perot in 92 was a third party candidate, that couldn't be elected president. !! factcheck some of that

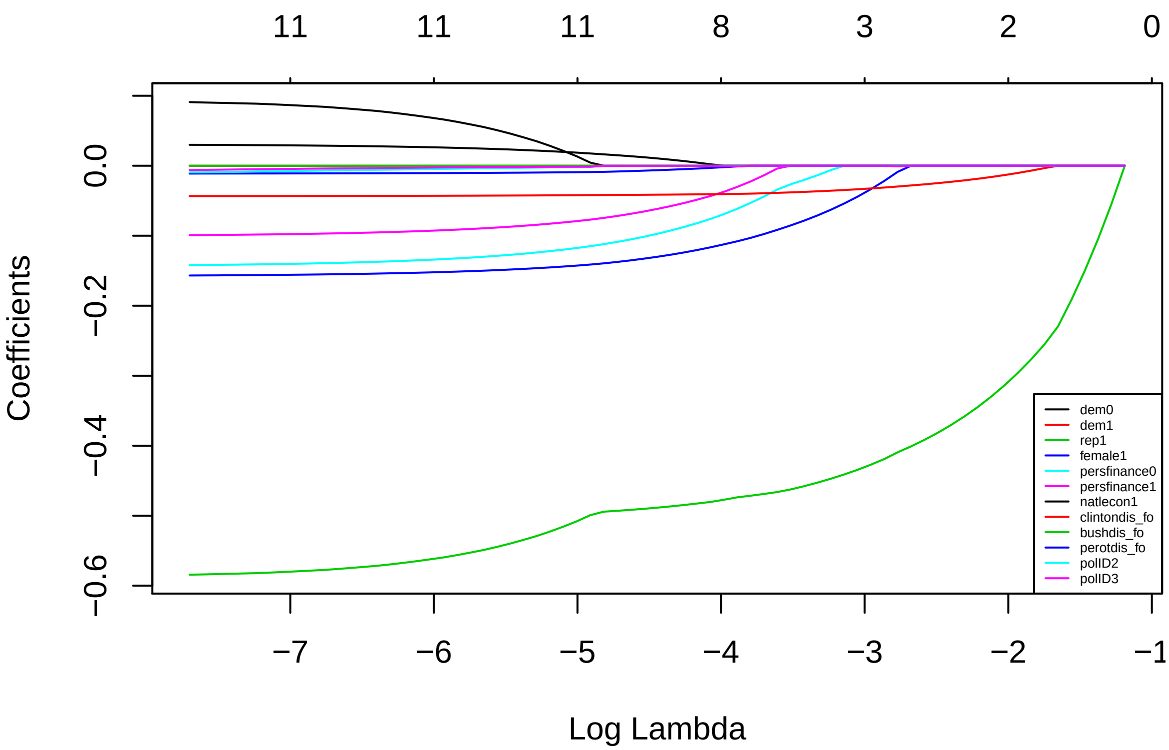
Building the models

```
# splitting the data into train and test
set.seed(777)
train.Index<-sample(1:nrow(vote),round(0.7*nrow(vote)),replace=F)

# creating the train and test sets using train.Index
```

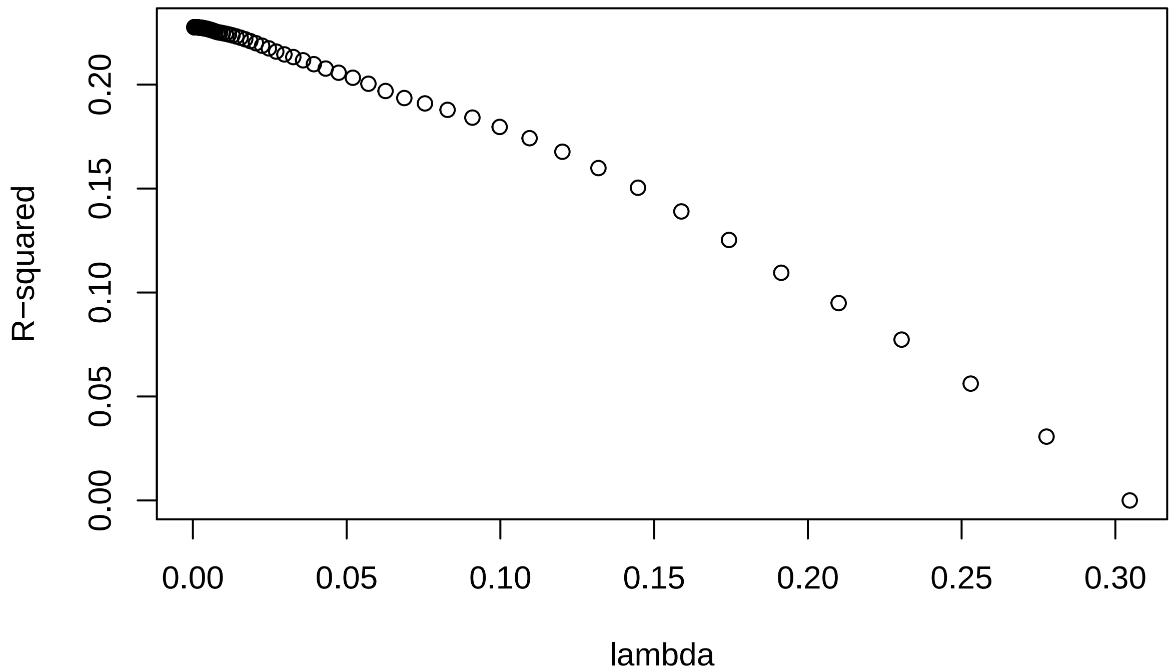
```r
vote.train<-vote[train.Index,]
vote.test<-vote[-train.Index,]

# creating x and y for model training
# y - a target vector
y.train<-vote.train$vote_num
y.test<-vote.test$vote_num

# X - a matrix with features/predictors
features<-c('dem','rep','female','persfinance','natlecon','clintondis_fo',
            'bushdis_fo','perotdis_fo','polID') # !!set this with colnames

#model.matrix( ~ ., data = scoring.train[, features])
X.train<-model.matrix(~ . -1, data=vote.train[,features]) # ??discrepancy between vote. & y.
X.test<-model.matrix(~ . -1, data=vote.test[,features])
```

1. L1-nrom / Lasso

```r
log_l1<-glmnet(X.train,y.train,alpha=1)

plot(log_l1, xvar = "lambda")
legend("bottomright", lwd = 1, col = 1:6, legend = colnames(X.train), cex = .4)
```




```r
plot(y = log_l1$dev.ratio,
     x = log_l1$lambda,
     xlab = "lambda",
     ylab = "R-squared")
```

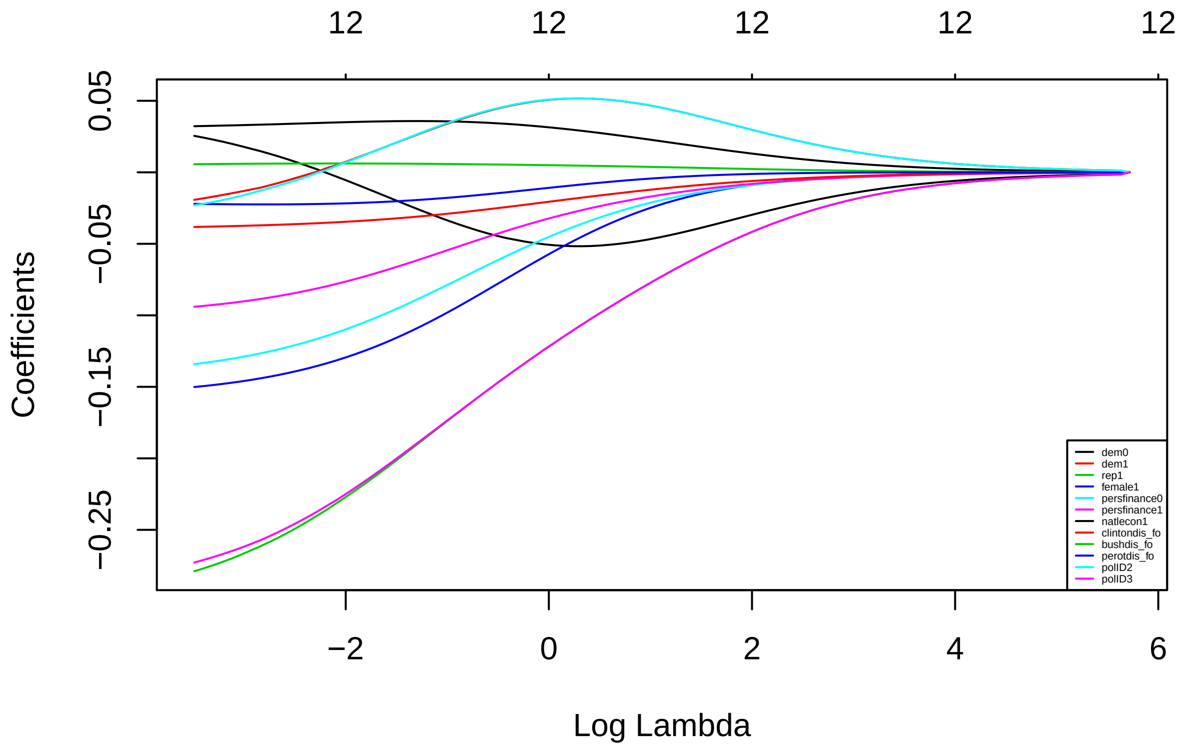



```r
# selecting the optimal lambda
set.seed(77)
log_l1_cv <- cv.glmnet(X.train, y.train, alpha = 1, type.measure = "class",
                       lambda = 10^seq(-5, 1, length.out = 100) , nfolds = 10)
```

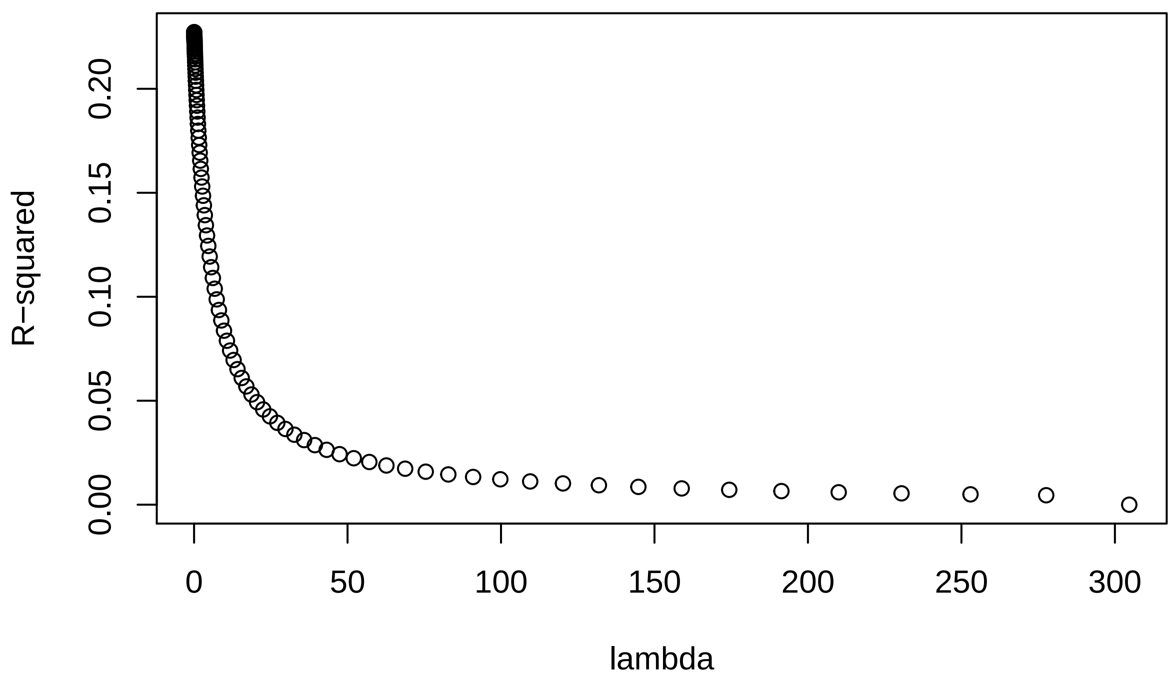
```
## Warning: Only mse, deviance, mae available as type.measure for Gaussian models;
## mse used instead
```

```r
y.predlog_l1 <-  predict(log_l1, newx = X.test,
                         type = "response", s = log_l1_cv$lambda.min)
```

```r
# Setting alpha = 0 implements ridge regression
log_r1 <- glmnet(X.train, y.train, alpha = 0)


plot(log_r1, xvar = "lambda")
legend("bottomright", lwd = 1, col = 1:6, legend = colnames(X.test), cex = .3)
```




```r
plot(y = log_r1$dev.ratio,
     x = log_r1$lambda,
     xlab = "lambda",
     ylab = "R-squared")
```




```r
# selecting the optimal lambda
set.seed(77)
log_r1_cv <- cv.glmnet(X.train, y.train, alpha = 0, type.measure = "class",
                       lambda = 10^seq(-5, 1, length.out = 100),
```

```r
                        nfolds = 10)
```

```
## Warning: Only mse, deviance, mae available as type.measure for Gaussian models;
## mse used instead
```

```r
y.predlog_r1 <-  predict(log_r1,    newx = X.test,
                         type = "response", s = log_r1_cv$lambda.min)
```

```r
#only catagorical

log1 <- glm(vote_num ~  dem + rep + female + persfinance +
              natlecon + clintondis_fo + persfinance + perotdis_fo +bushdis_fo,
           data = vote)
#both
log2 <- glm(vote_num ~ dem + rep + female +persfinance +
          natlecon ,
           data = vote)

#only continuous
log3 <- glm(vote_num ~ clintondis_fo + bushdis_fo +
              perotdis_fo,
           data = vote)


pred.log1 <- predict(log1, vote, type = "response")
pred.log2 <- predict(log2, vote, type = "response")
pred.log3 <- predict(log3, vote, type = "response")
```

```r
set.seed(7)
train.Index <- caret::createDataPartition(vote$vote, p = 0.7, list = F)
vote.train <- vote[ train.Index,]
vote.test  <- vote[-train.Index,]

# features to be used for model training
features <- c('vote', 'dem','rep','female','persfinance','natlecon',
              'clintondis_fo', 'bushdis_fo','perotdis_fo','polID')

# ----- Fitting a model ------
# Training classification decision tree
dt <- rpart(vote ~ .,
            data = vote.train[,features],
            method = "class",   #cause we have a classification problem
            parms = list(split = "information"),  # the splitting index
            model = T)

# ----- Deriving Predictions ------
# Predicting the instance of surviving
# first column  - probability of 0 for each observation
# second column - probability of 1
pred.dt <- predict(dt, newdata = vote.test, type = "prob")[, 2]
```

Predictions

```r
Accuracy<-function(pred,real,threshold=.5) {
  predClass<-ifelse(pred>threshold,1,0)
  return(sum(predClass==real)/length(real))
}

# Accuracy
(acc1 <- Accuracy(pred = pred.log1, real = vote$vote_num))
```

```
## [1] 0.3410341
```

```r
(acc2 <- Accuracy(pred = pred.log2, real = vote$vote_num))
```

```
## [1] 0.3410341
```

```r
(acc3 <- Accuracy(pred = pred.log3, real = vote$vote_num))
```

```
## [1] 0.3410341
```

```r
# Brier Score
(BS.log1 <- sqrt(mean((vote$vote_num - pred.log1)^2)))
```

```
## [1] 0.6459124
```

```r
(BS.log2 <- sqrt(mean((vote$vote_num- pred.log2)^2)))
```

```
## [1] 0.6552625
```

```r
(BS.log3 <- sqrt(mean((vote$vote_num - pred.log3)^2)))
```

```
## [1] 0.6835589
```

```r
(accLasso <- Accuracy(pred = y.predlog_l1, real = y.test))
```

```
## [1] 0.3516484
```

```r
(accLRidge <- Accuracy(pred = y.predlog_r1, real = y.test))
```

```
## [1] 0.3516484
```

```r
(BS.logL1 <- sqrt(mean((y.test - y.predlog_l1)^2)))
```

```
## [1] 0.6852876
```

```r
(BS.logL2 <- sqrt(mean((y.test - y.predlog_r1)^2)))
```

```
## [1] 0.6852333
```

```r
# ----- Evaluating Prediction Quality -----
# Calculate performance with AUC and RMSE
auc(vote.test$vote_num, pred.dt)
```

```
## Warning in roc.default(response, predictor, auc = TRUE, ...): 'response'
## has more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
## Area under the curve: 0.9299
```

```r
( rmse <- sqrt(mean((vote.test$vote_num - pred.dt)^2)) )
```

```
## [1] 1.565776
```

```r
Accuracy(pred=pred.dt, real=vote.test$vote_num)
```

```
## [1] 0.01845018
```

```r
# Naive Classifier
baseline_probability <- sum(vote.train$vote_num == 1)/nrow(vote.train)
pred.baseline <- rep(baseline_probability, nrow(vote.test))

auc(vote.test$vote_num, pred.baseline)
```

```
## Warning in roc.default(response, predictor, auc = TRUE, ...): 'response'
## has more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead

## Setting levels: control = 1, case = 2
## Setting direction: controls < cases

## Area under the curve: 0.5
```

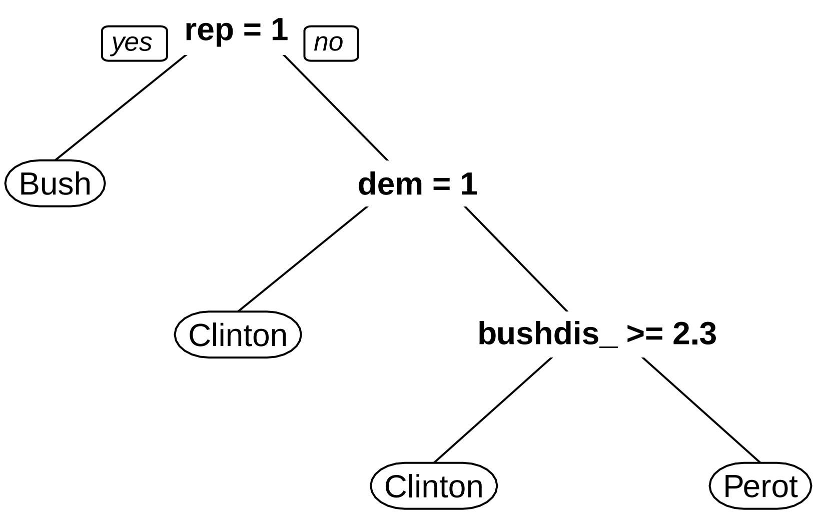
```r
( rmse <- sqrt(mean((vote.test$vote_num - pred.baseline)^2)) )
```

```
## [1] 1.679247
```

```r
Accuracy(pred=pred.baseline, real=vote.test$vote_num)
```

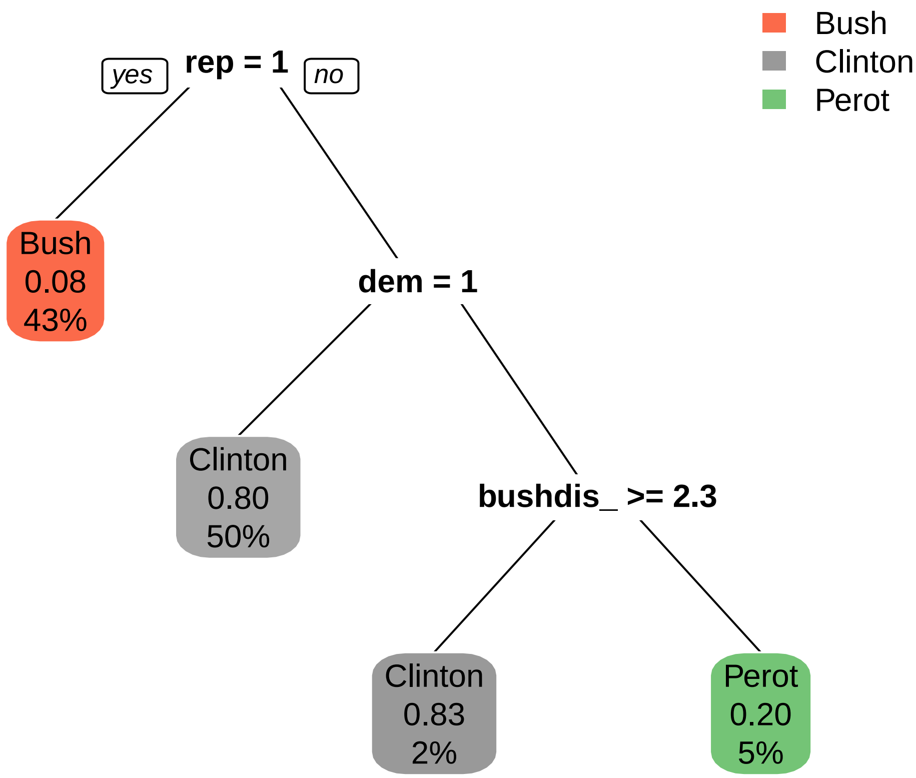
```
## [1] 0
```

```r
# Visualizing the results from "dt" using the prp() function
# default plot
prp(dt)
```




```r
# prints the percentage of observations and class probabilities in each node
prp(dt, extra = 106, border.col = 0, box.palette="auto")
```

```
## Warning: extra=106 but the response has 3 levels (only the 2nd level is
## displayed)
```


Bush
Clinton
Perot
yes
rep = 1
no
Bush
0.08
43%
dem = 1
Clinton
0.80
50%
bushdis_ >= 2.3
Clinton
0.83
2%
Perot
0.20
5%