R Notebook

```
library(haven)
conf06 <- read_dta("classification-master/problem-set-2-master/PSET 2 Files/conf06.dta")
```

```
conf061 <- subset(conf06, conf06$nominee!=" ALITO ")
vars <- c("vote", "nominee", "sameprty", "qual", "lackqual",
"EuclDist2", "strngprs") # vector of vars
conf <- conf061[vars] # retain only key vars from above object
conf$numvote <- as.numeric(conf$vote)-1 # from 1/2 to 0/1
conf$numstrngprs <- as.numeric(conf$strngprs)-1 # same as above
```

Question 1:

```
#creating an 80-20 data split
samples <- sample(1:nrow(conf),
                  nrow(conf)*0.8, #take 80% of our data
                  replace = FALSE)
train <- conf[samples, ]
test <- conf[-samples, ]
```

Question 2:

```
#building a logit model

vote <- test$vote



logit <- glm(vote ~ sameprty + qual + EuclDist2 + strngprs,
             data = train,
             family = binomial)

#storing output in a meaningful way

logit.probs1 <- predict(logit, newdata=test, type="response")

logit.pred1 <- ifelse(logit.probs1 > 0.5, 1, 0)

table(logit.pred1, vote)
```

```
##            vote
## logit.pred1   0   1
##           0  46  20
##           1  55 641
```

```
mean(logit.pred1 == vote)
```

```
## [1] 0.9015748
```

Looking at the output of this confusion matrix, it seems that our logit classifier does a fairly good job of predicting the binary output of vote choice. Using the output of the confusion matrix, I find that the majority of our logit models predictions for the observations fall in the true negative and true positive quadrants. Only a few observations fall into the false negative and false positive quadrants. The classification rate for the model is 91.33% which is pretty high. The false negative rate is 58.42% while the false positive rate is 2.1%. While the false negative rate is high, there are not many negative implications for a no vote when it is really a yes vote so this may be negliable.

Question 3:

```
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```
library(randomForest)
```

```
## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

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

```
library(AUC)
```

```
## AUC 0.3.0

## Type AUCNews() to see the change log and ?AUC to get an overview.

##
## Attaching package: 'AUC'

## The following objects are masked from 'package:caret':
##
##     sensitivity, specificity
```

```
library(MASS)


lda2 <- lda(vote ~ sameprty + qual + EuclDist2 + strngprs, data=train)

vote = test$vote

lda2.pred <- predict(lda2, newdata=test)


lda.pred <- predict(lda2, newdata=test)
table <- table(lda.pred$class, vote)
table
```

```
##    vote
##       0   1
##   0  50  30
##   1  51 631
```

```
mean(lda2.pred$class == vote)
```

```
## [1] 0.8937008
```

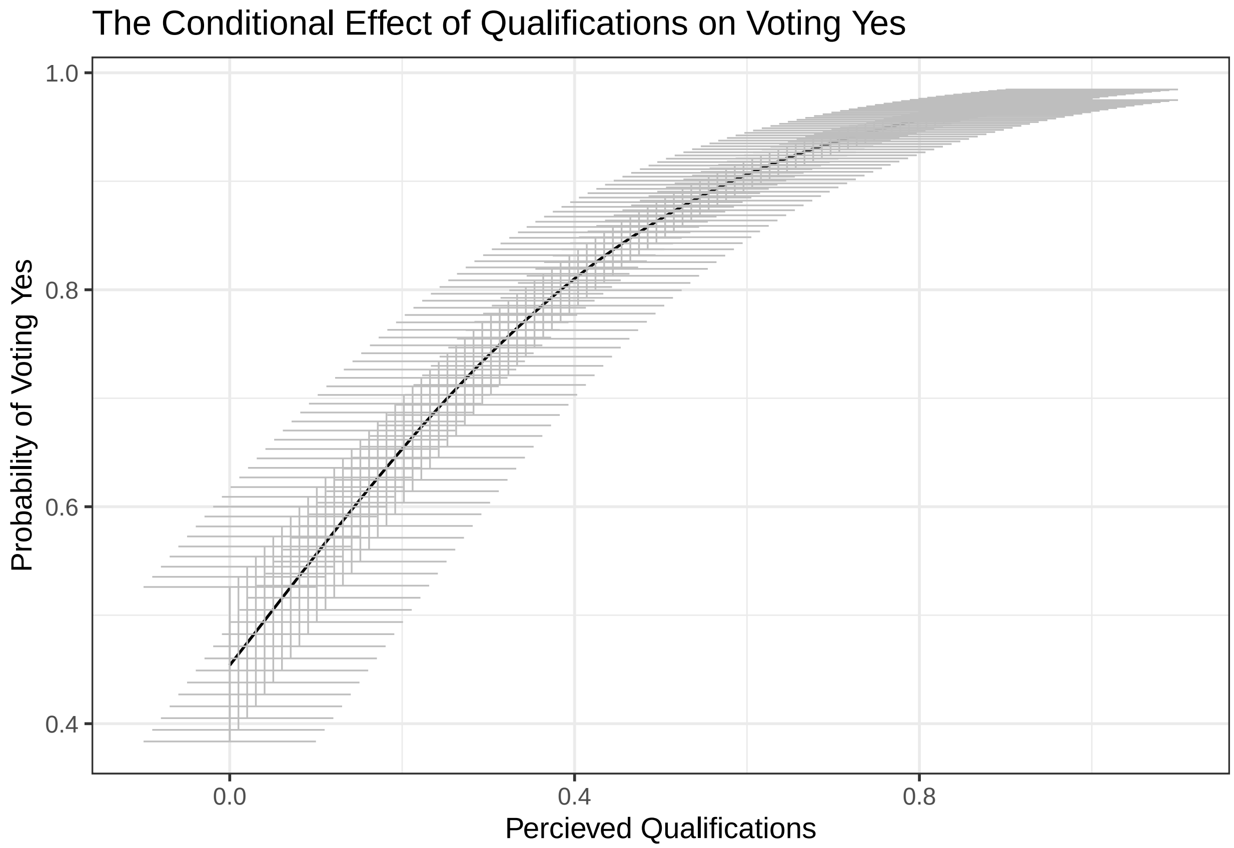
After fitting a LDA classifier model, the resulting confusion matrix indicates that the model does a fairly good job classifying the final vote of legislators to their binary outcome based on the explanatory variables. The number of false positives and false negatives are pretty low in comparision to the number of true positives and true negatives. The classification rate is 91.33% which is very high indicating that this is a good model. The false negative rate is about 2.6% (18/(18+655)) which is very low and the false negative rate is about

53.9% which is relatively high and concerning for the model if a false negative has bad implications, but in this case it does not.

Question 4:

```
logitmod3 <- glm(vote ~ sameprty + qual + EuclDist2 + strngprs,
             data = conf,
             family = binomial)

# CIs for predicted probabilities
#creating synthetic data
newdata2 <- with(conf, data.frame(qual = rep(seq(from = 0, to = 1, length.out = 100)),
                                  sameprty = mean(sameprty),
                                  EuclDist2 = mean(EuclDist2), strngprs = mean(strngprs)))


newdata3 <- cbind(newdata2, predict(logitmod3,
                                    newdata = newdata2,
                                    type ="link",
                                    se = TRUE))

# Add CIs
newdata3 <- within(newdata3, {
  PredictedProb <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit))
  UL <- plogis(fit + (1.96 * se.fit))
})


# Plot predictions with CIs
ggplot(newdata3, aes(x = qual, y = PredictedProb)) +
  geom_line() +
  geom_errorbar(aes(ymin = LL, ymax = UL),
                color="gray",
                size=.3,
                width=.2,
                position = position_dodge(.9)) +
  labs(x = "Percieved Qualifications",
       y = "Probability of Voting Yes") +
  ggtitle("The Conditional Effect of Qualifications on Voting Yes") + theme_bw() +
  theme(legend.justification = c(.7,1),
        legend.position = c(.9,.3))
```

```
## Warning: position_dodge requires non-overlapping x intervals
```



The plot above shows that as the percieveed qualifications of the nominee increases, the probability that they recieve a yes vote increases. The probability that the nominee recieves a yes vote increases substantially as their percieved qualifications increases at the lower end of the qualification scale. As their percieved qualifications become incredibly high (.8-1), the candidate sees very little difference in the probability that they will be supported. Additionally, at lower values of percieved qualifications there is more uncertainity around the probability the candidate will recieve support than at higher values of percieved qualifications.

Problem 5:

```
summary(logit)
```

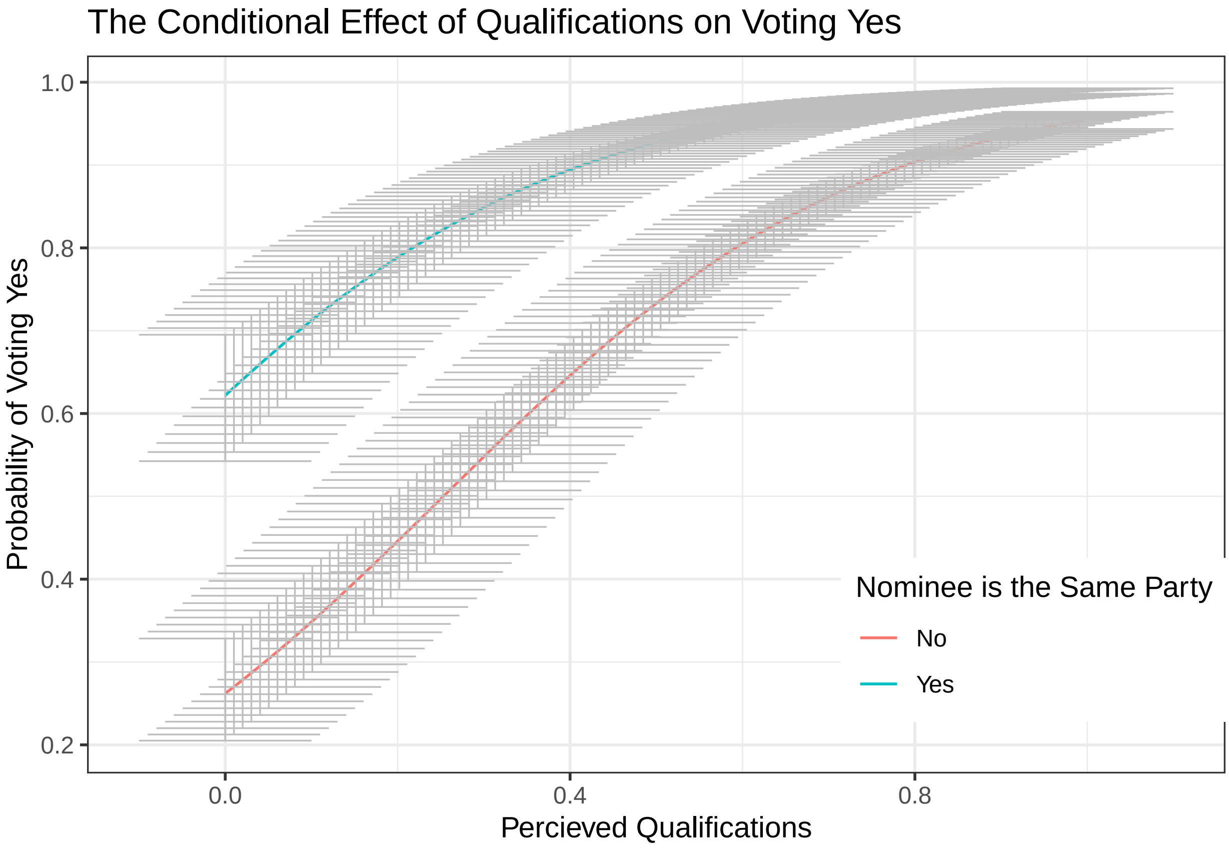
```
##
## Call:
## glm(formula = vote ~ sameprty + qual + EuclDist2 + strngprs,
##     family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1637   0.0985   0.2085   0.4279   2.1363
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.0338     0.2104  -4.915 8.90e-07 ***
## sameprty      1.5854     0.1675   9.467  < 2e-16 ***
## qual          4.1969     0.2533  16.571  < 2e-16 ***
## EuclDist2    -4.1031     0.3069 -13.371  < 2e-16 ***
## strngprs      1.1078     0.1404   7.892 2.98e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2308.2  on 3046  degrees of freedom
## Residual deviance: 1456.1  on 3042  degrees of freedom
## AIC: 1466.1
##
## Number of Fisher Scoring iterations: 6
```

Overall, my exploration of this dataset leads me to believe that there are several important considerations that go into Senator's votes on Supreme Court nominees. First, if the nominee is the same party as the Senator, the Senator is significantly more likely to vote in support of the nominee, suggesting that partisanship plays a large part in the confirmation process. Additionally, if the nominee is percieved to be more qualified the Senator is more likely to vote to confirm them. Also, if the incumbent president is strong or in his last year of his term, senators are more likely to vote for his nominee. Finally, the larger the difference between the president's and the nominee's ideological ideal points, the less likely that the senator will confirm the nominee. Each of these variables help us to better understand how to classify a senator's vote for a potential nominee.

Problem 6:

```r
logitmod3 <- glm(vote ~ sameprty + qual + EuclDist2 + strngprs,
             data = conf,
             family = binomial)

# CIs for predicted probabilities
#creating synthetic data
newdata2 <- with(conf, data.frame(qual = rep(seq(from = 0, to = 1, length.out = 100),2),
                                  sameprty = rep(0:1, each=100),
                                  EuclDist2 = mean(EuclDist2), strngprs = mean(strngprs)))


newdata3 <- cbind(newdata2, predict(logitmod3,
                                    newdata = newdata2,
                                    type ="link",
                                    se = TRUE))

# Add CIs
newdata3 <- within(newdata3, {
  PredictedProb <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit))
  UL <- plogis(fit + (1.96 * se.fit))
})

newdata3$sameprty <- factor(newdata3$sameprty, labels=c("No", "Yes"))

# Plot predictions with CIs
ggplot(newdata3, aes(x = qual, y = PredictedProb, color=sameprty)) +
  geom_line() +
  geom_errorbar(aes(ymin = LL, ymax = UL),
                color="gray",
                size=.3,
                width=.2,
                position = position_dodge(.9)) +
  labs(x = "Percieved Qualifications",
      y = "Probability of Voting Yes",
      color= "Nominee is the Same Party") +
  ggtitle("The Conditional Effect of Qualifications on Voting Yes") + theme_bw() +
  theme(legend.justification = c(.7,1),
        legend.position = c(.9,.3))
```

```
## Warning: position_dodge requires non-overlapping x intervals
```



This graph shows us that if the nominee is the same party as the senator, at any level of the nominee's percieved qualifications, they are more likely to recieve support from the senator than if they were not the same party as the senator.

Question 2:

Problem 1:

```
library(wnominate) # for algorithm
```

```
## Loading required package: pscl

## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis

##
## ## W-NOMINATE Ideal Point Package
## ## Copyright 2006 -2019

## ## Keith Poole, Jeffrey Lewis, James Lo, and Royce Carroll

## ## Support provided by the U.S. National Science Foundation

## ## NSF Grant SES-0611974
```

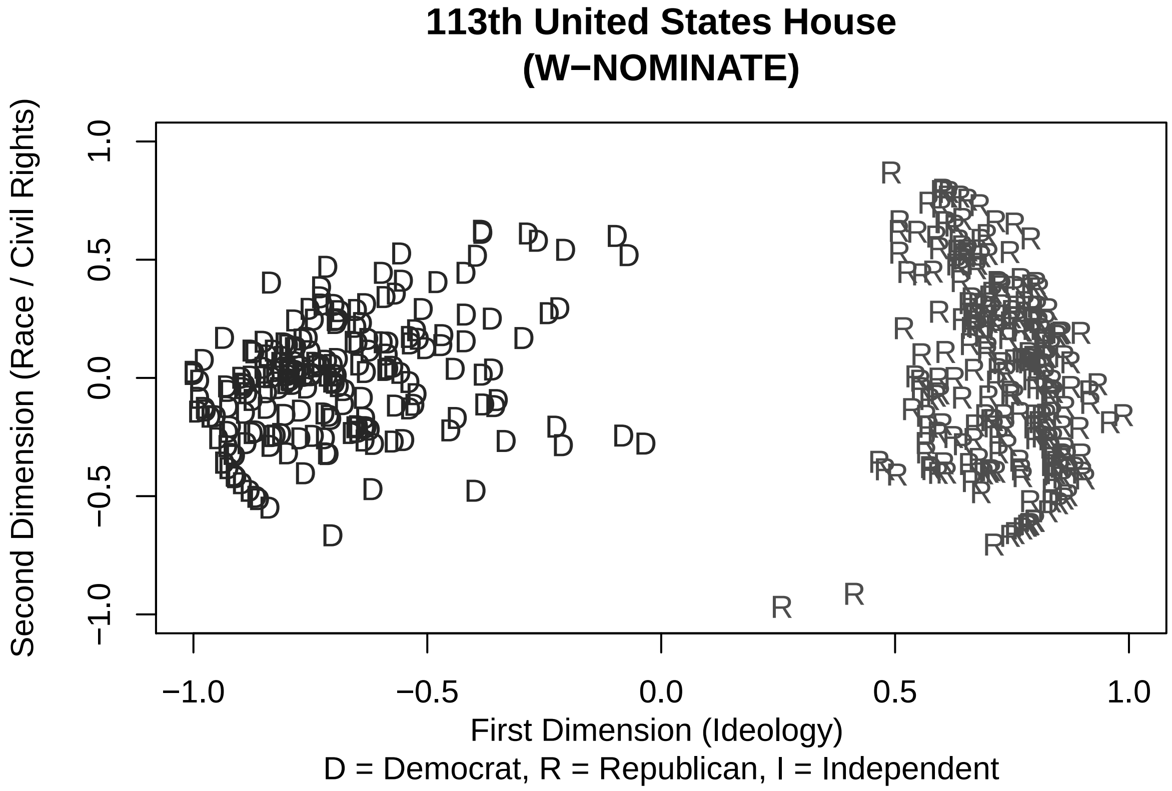
```
library(pscl)
rollcall1 <- readKH("classification-master/problem-set-2-master/PSET 2 Files/hou113kh.ord",
                    dtl=NULL,
                    yea=c(1,2,3),
                    nay=c(4,5,6),
                    missing=c(7,8,9),
                    notInLegis=0,
                    desc="113th_House_Roll_Call_Data",
                    debug=FALSE)
```

```
## Attempting to read file in Keith Poole/Howard Rosenthal (KH) format.
## Attempting to create roll call object
## 113th_House_Roll_Call_Data
## 445 legislators and 1202 roll calls
## Frequency counts for vote types:
## rollCallMatrix
##      0      1      6      7      9
##  14576 295753 202943    290  21328
```

```
wnom_result <- wnominate(rollcall1, #data matrix
                          dims = 2,#number of dimensions that are characterizing (k=2)
                          minvotes = 20, #minimum number of votes needed to get your ideal point
                          lop = 0.025,
                          polarity = c(2,2)) #need to intialize on extreme legislatures, 1st entry is th
```

```
##
## Preparing to run W-NOMINATE...
##
##  Checking data...
##
##      ... 1 of 445 total members dropped.
##
##      Votes dropped:
##      ... 181 of 1202 total votes dropped.
##
##  Running W-NOMINATE...
##
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Starting estimation of Beta...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Starting estimation of Beta...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Estimating weights...
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Estimating weights...
##      Getting bill parameters...
##      Getting legislator coordinates...
##
##
## W-NOMINATE estimation completed successfully.
## W-NOMINATE took 233.037 seconds to execute.
```

```
wnom1 <- wnom_result$legislators$coord1D
wnom2 <- wnom_result$legislators$coord2D
party <- rollcall1$legis.data$party



# custom plot
plot(wnom1, wnom2,
     main="113th United States House\n(W-NOMINATE)",
     xlab="First Dimension (Ideology) \nD = Democrat, R = Republican, I = Independent",
     ylab="Second Dimension (Race / Civil Rights)",
     xlim=c(-1,1), ylim=c(-1,1), type="n") #type="n" allows us to manually overlay the graph
points(wnom1[party=="D"], wnom2[party=="D"], pch="D", col="gray15")
points(wnom1[party=="R"], wnom2[party=="R"], pch="R", col="gray30")
points(wnom1[party=="Indep"], wnom2[party=="Indep"], pch="I", col="red")
```

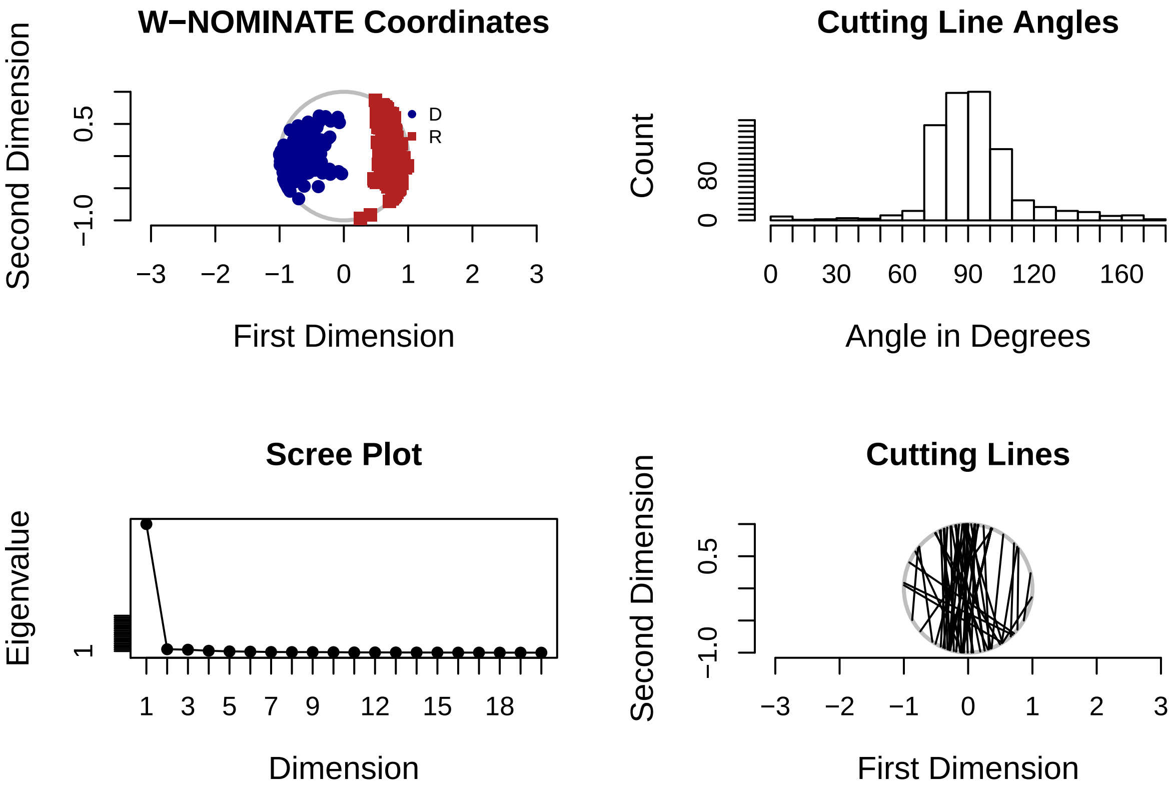


After fitting a W-NOMINATE algorithm to the roll call votes of the 113th congress, I find that Democrats all have negative values on the first dimensin of ideology (indicating their liberalism), while the Republicans all have positive values (indicating their conservatism). Unlike the ideological dimension, along the second dimension of race/civil rights both parties have substantial cohorts along the moderate section of the dimension.

The Republican party moreso than the Democratic Party occupies some of the extreme spaces on the civil rights/race dimension. Additionally, the Democratic party is more spread out among the ideological dimension than the Republican party which is centralized at a pretty conservative placement of (.5-1). Nonetheless, the parties seem pretty polarized along this dimension and their is little to no overlap of party members in the center of the dimension.

Problem 2

```
plot(wnom_result)
```



```
## NULL
```

```
wnom_result$fits
```

```
## correctclass1D correctclass2D        apre1D        apre2D         gmp1D
##     92.7942200     93.5996170     0.8168426     0.8373143     0.8398967
##          gmp2D
##      0.8566988
```

Looking at the output of these dimensionality plots, it is clear that at least one dimension maps the political preferences of these legislators well. This dimension is the ideological dimension. As one can see from the top left chart, the ideological dimension almost exactly divides the Republican and the Democratic parties. The two plots to the right further demonstrate that the ideological dimension is the best dimension to map legislators as the distribution of the cutting line angle has a mean of about 90 degrees, suggesting a perfect split between the parties on this dimension. There is also likely a second dimension, race policy, that maps some of these legislators preferences. On the Scree plot we see that with two dimensions, the Eigenvalue is slightly above 1. This dimension does not map preferences as well as the ideological one though.

The GMP's and APRE's of the 1st and 2nd dimension show that there is a 81.6% and 83.73% reduction in error in predicting the roll call votes of the 113th House of Representatives which are both high values, leading me to believe that these dimensions do a good job in spatially mapping the preferences of the members.

Problem 3:

The three common methods for unfolding binary data are NOMINATE, IRT, and optimal classifcation. NOMINATE and IRT view voting as a probabilistic event, or the result of deterministic components. But NOMINATE assumes Gaussian normal utility while IRT assumes quadratic utilies in their probability distributions. We should use NOMINATE if we think that as we move away from the legislator's ideal point, they lose utility exponentially, whereas with IRT we should use it when we assume that the legislators don't use a lot of utility if the policy is slightly off from their ideal point. Nonparametric optimal classification does not treat voting probabilistically and does not assume a probability distribution of utility.