Homework 2

Sanittawan Tan

11/6/2019

Load libraries

```
library(tidyverse)
```

```
## Registered S3 method overwritten by 'rvest':
##   method            from
##   read_xml.response xml2
```

```
## -- Attaching packages ---------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(foreign) # for loading Stata data
library(MASS) # for LDA
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(faraway)
library(ggplot2)
library(arm)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loading required package: lme4
```

```
##
## arm (Version 1.10-1, built: 2018-4-12)
```

```
## Working directory is /Users/sanittawan/Documents/Nikki/UChicago/Classes/Autumn_2019/com_methods_am_p
```

```
##
## Attaching package: 'arm'
```

```
## The following objects are masked from 'package:faraway':
##
##     fround, logit, pfround
```

```
library(pROC) # for ROC curve
```

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

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

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

```
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:pROC':
##
##     auc, roc
```

```
library(wnominate)
```

```
## 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)
```

Part 1: Classification

Load and prepare the dataset

```
input_file <- "conf06.dta"
data <- read.dta(input_file)
conf06 <- subset(data, data$nominee!="ALITO")
vars <- c("vote", "nominee", "sameprty", "qual",
```

```
          "lackqual", "EuclDist2", "strngprs")
conf <- conf06[vars]
conf$numvote <- as.numeric(conf$vote)-1
conf$numstrngprs <- as.numeric(conf$strngprs)-1
```

Question 1: Train-Test Split

```
set.seed(611)
samples <- sample(1:nrow(conf),
                  nrow(conf) * 0.8,
                  replace = FALSE)
train_conf <- conf[samples, ]
test_conf <- conf[-samples, ]
```

Question 2: Logit Classifier

I first inspected the data's summary statistics against the paper (Table 1; p. 300). I found that the means of each variable are the same for the data that we are using and the one that the paper used.

```
summary(conf)
```

```
##   vote        nominee             sameprty           qual
##  no : 444   Length:3709        Min.   :0.0000   Min.   :0.1100
##  yes:3265   Class :character   1st Qu.:0.0000   1st Qu.:0.6500
##             Mode  :character   Median :1.0000   Median :0.8850
##                                Mean   :0.5549   Mean   :0.7781
##                                3rd Qu.:1.0000   3rd Qu.:1.0000
##                                Max.   :1.0000   Max.   :1.0000
##     lackqual        EuclDist2           strngprs       numvote
##  Min.   :0.0000   Min.   :0.0000001   weak  :1587   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0205085   strong:2122   1st Qu.:1.0000
##  Median :0.1150   Median :0.0938021                 Median :1.0000
##  Mean   :0.2219   Mean   :0.1823883                 Mean   :0.8803
##  3rd Qu.:0.3500   3rd Qu.:0.2812036                 3rd Qu.:1.0000
##  Max.   :0.8900   Max.   :1.2681841                 Max.   :1.0000
##   numstrngprs
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :1.0000
##  Mean   :0.5721
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

Next, I fit the logistic regression model to classify yes and no votes.

```
logit <- glm(numvote ~ EuclDist2 + qual + numstrngprs + sameprty,
             data = train_conf,
             family = binomial); summary(logit)
```

```
##
## Call:
## glm(formula = numvote ~ EuclDist2 + qual + numstrngprs + sameprty,
##     family = binomial, data = train_conf)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2676   0.0840   0.1906   0.3889   2.1544
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.0959     0.2109  -5.196 2.04e-07 ***
## EuclDist2    -4.0903     0.3180 -12.864  < 2e-16 ***
## qual          4.3239     0.2623  16.482  < 2e-16 ***
## numstrngprs   1.5592     0.1551  10.054  < 2e-16 ***
## sameprty      1.4257     0.1706   8.358  < 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: 2189.1  on 2966  degrees of freedom
## Residual deviance: 1345.8  on 2962  degrees of freedom
## AIC: 1355.8
##
## Number of Fisher Scoring iterations: 6
```

As we can see from the regression output, all of our explanatory variables are statistically significant at a high level of confidence. The result suggests that each of the explanatory variable is a good predictor of how a senator would vote in the Supreme Court nomination. Since this is a logistic model, the interpretation of the coefficients is not as straightforward as a linear regression model. However, we can roughly tell that the distant between a senator's ideal point and the nominee's inferred ideal point and the log odds of voting yes for the nominee does have a negative relationship as opposed to the remaining variables.

Next, I predicted the probabilities of voting yes on supreme court nominees using the test set. If the predicted probabilities are greater than 0.5, I will predict the observation as yes.

```
logit.probs <- predict(logit,
                       newdata = test_conf,
                       type = "response")
# check the predictions
head(logit.probs)
```

```
##         4         7        16        19        21        24
## 0.1602836 0.8848071 0.9287092 0.9269489 0.9041158 0.9282559
```

```
logit.pred <- ifelse(logit.probs > 0.5, 1 ,0)
# check and see if the label works as expected. It did.
head(logit.pred)
```

```
##  4  7 16 19 21 24
##  0  1  1  1  1  1
```

Next, I evaluate the test set performance.

```
table(logit.pred, test_conf$numvote)
```

```
##
## logit.pred   0   1
##          0  35  11
##          1  50 646
```

```
mean(logit.pred == test_conf$numvote)
```

```
## [1] 0.9177898
```

The confusion matrix indicates that the columns are actual votes and the rows are predicted votes from our logistic regression model. The table tells us that:

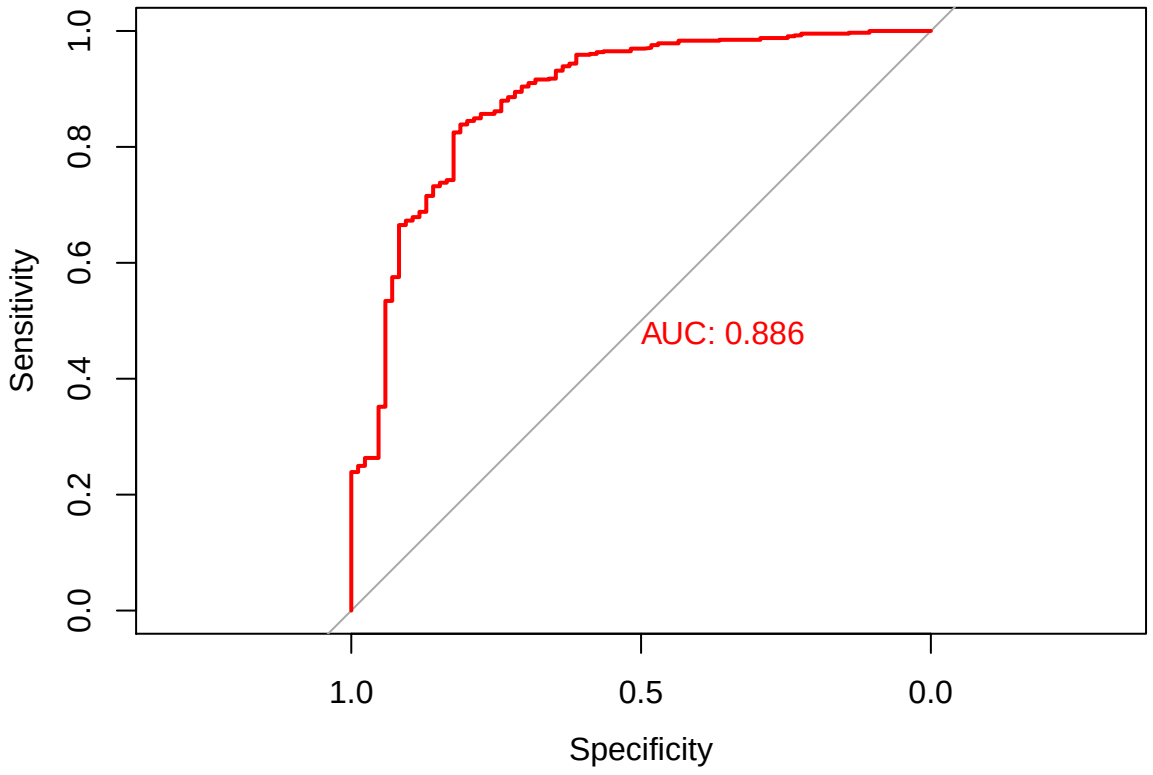
- row 1, col 1 is the true negatives which are the number of no votes that are predicted correctly.
- row 1, col 2 is the false negatives which are the number of *actual yes votes* that the model incorrectly predicts as no votes.
- row 2, col 1 is the false positives which are the number of *actual no votes* that our model incorrectly predicted as yes votes.
- row 2, col 2 is the true positives which are the number of actual yes votes that our model predicts correctly.

We can see from the cross table above that there are 50 false positives and 11 false negatives. The accuracy rate of the test set is 91.78 per cent. This tells us that the logistic regression model performs quite well. However, the below ROC curve also indicates that the fit is good since the area under the curve is 0.886.

```
plot.roc(test_conf$numvote, logit.probs, col = "red", print.auc = TRUE)
```

```
## Setting levels: control = 0, case = 1
```

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

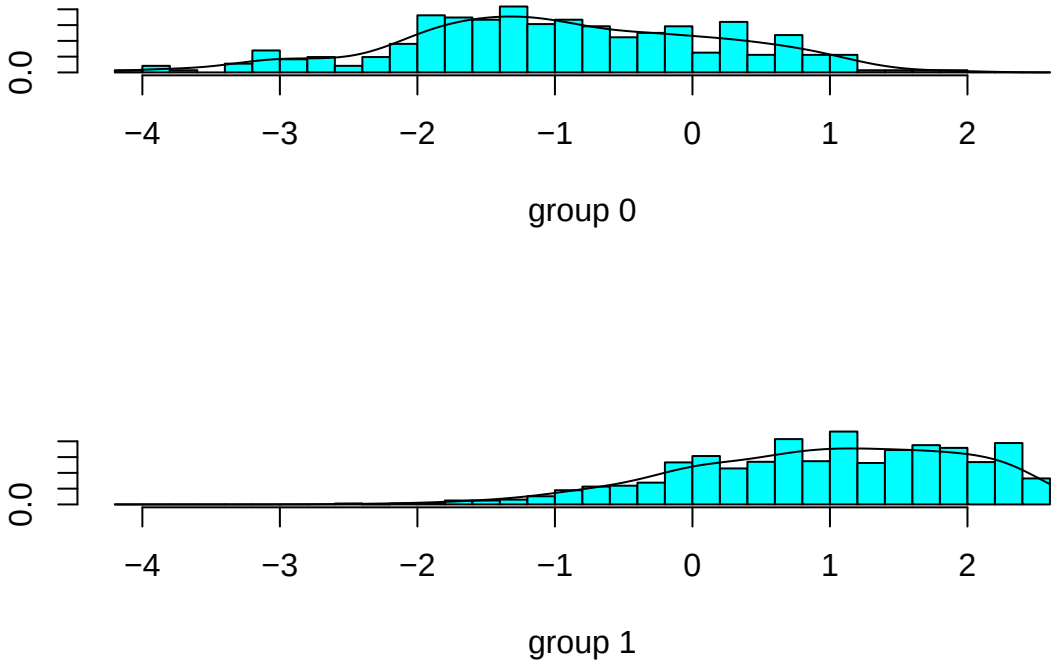


Question 3: LDA Classifier

Firstly, I fit an LDA model using the train set.

```
lda <- lda(numvote ~ EuclDist2 + qual + numstrngprs + sameprty,
           data = train_conf)
```

```
plot(lda, type = "both")
```



From the plot, we can see that the LDA models the distribution of the explanatory variables in each class separately. As there are 2 classes, we obtain 2 different distributions.

Next, I evaluate the LDA model.

```
lda.pred <- predict(lda, newdata=test_conf)
lda_pred_df <- data.frame(lda.pred)
```

```
table(lda.pred$class, test_conf$numvote)
```

```
##
##       0   1
##   0  39  15
##   1  46 642
```

The cross table indicates that there are 46 false positives and 15 false negatives. Compared to the logistic regression model, the number of false positives decreases but the number of false negatives increases. The accuracy rate of the test set is 91.78 per cent which the same as that of the logistic regression model. (This might have been because I set a seed.)

```
mean(lda.pred$class == test_conf$numvote)
```

```
## [1] 0.9177898
```

Question 4: Logit Predicted Probabilities Plot

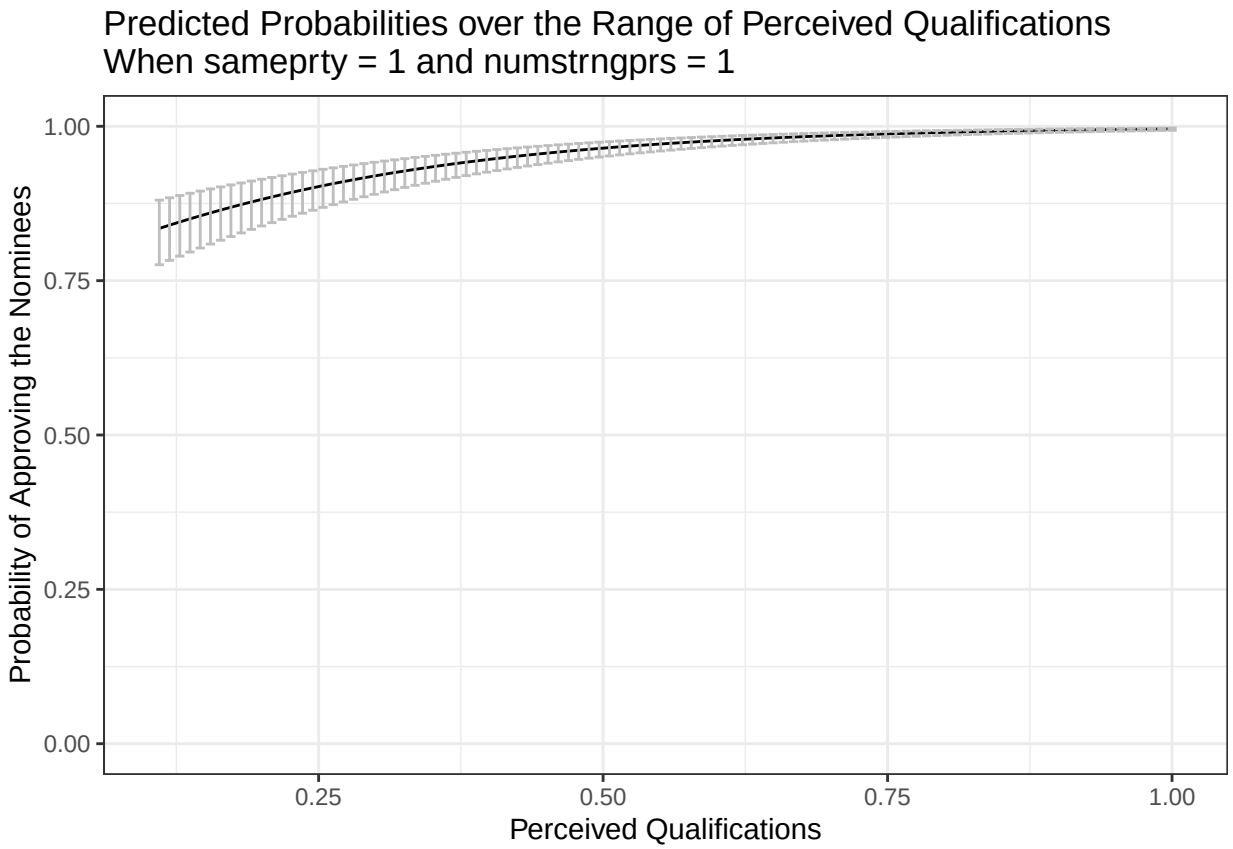
I generated several different plots to see effects of the predicted probabilities over a range of perceived qualification values while holding the rest of the variables constant. As 2 of our explanatory variables, sameprty and numstrngprs, are dummy, it is only possible to hold a dummy variable constant at one value at the time. Therefore, 4 plots are necessary to see *all* possible effects at varying values of sameprty and numstrngprs. I am curious about all the possibilities; therefore, I will generate all the plots. I am forcing the

y-axis to be from 0 to 1 for the sake of comparison across all the plots. I believe it makes the results clearer. Note that all the plot is given that the ideology distance is at the mean of the dataset.

When a senator shares the president's party affiliation (sameprty = 1) and the president is strong (numstrngprs = 1)

```
same_strongdta <- with(conf, data.frame(qual = seq(from = min(qual), to = max(qual),
                                        length.out = 100),
                       EuclDist2 = mean(EuclDist2),
                       # set sameparty to 1
                       sameprty = 1,
                       # set numstrongpres to 1
                       numstrngprs = 1)
                )
same_strongdta2 <- cbind(same_strongdta, predict(logit,
                                        newdata = same_strongdta,
                                        type = "link",
                                        se = TRUE))
# Calculate confidence intervals
same_strongdta2 <- within(same_strongdta2, {
  PredictedProb <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit))
  UL <- plogis(fit + (1.96 * se.fit))
})
```

```
ggplot(same_strongdta2, aes(x = qual, y = PredictedProb)) +
    geom_line() +
    geom_errorbar(aes(ymin = LL, ymax = UL),
                  position = position_dodge(width = 0.9),
                  color="gray") +
    ylim(0, 1) +
  labs(x = "Perceived Qualifications",
       y = "Probability of Approving the Nominees") +
  ggtitle("Predicted Probabilities over the Range of Perceived Qualifications\nWhen sameprty = 1 and nu
  theme_bw() +
  theme(legend.justification = c(.7,1),
        legend.position = c(.9,.3))
```

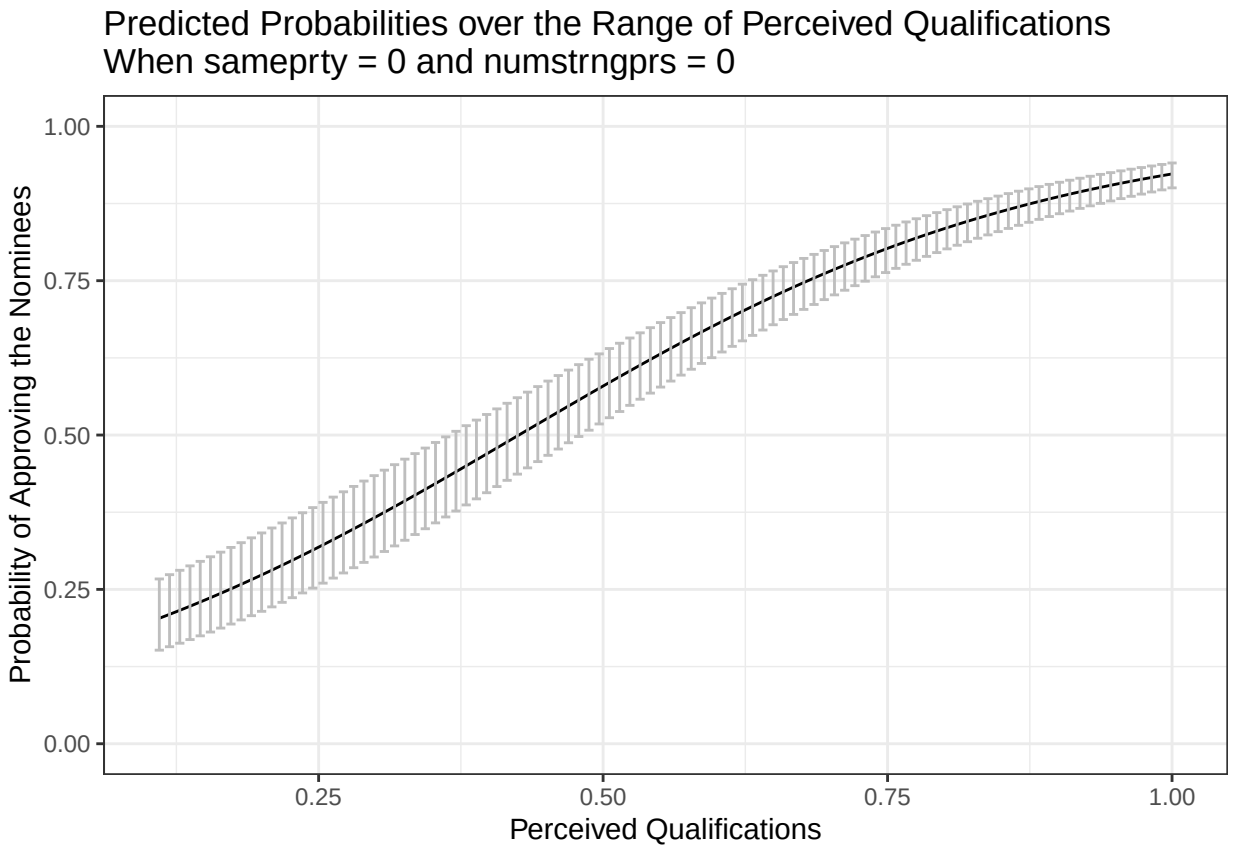


When a senator does not share the president's party affiliation (sameprty = 0) and the president is not strong (numstrngprs = 0)

```
nsame_nstrongdta <- with(conf, data.frame(qual = seq(from = min(qual), to = max(qual),
                                 length.out = 100),
                       EuclDist2 = mean(EuclDist2),
                       # set sameparty to 0
                       sameprty = 0,
                       # set numstrongpres to 0
                       numstrngprs = 0)
                )
nsame_nstrongdta2 <- cbind(nsame_nstrongdta, predict(logit,
                                 newdata = nsame_nstrongdta,
                                 type = "link",
                                 se = TRUE))
# Calculate confidence intervals
nsame_nstrongdta2 <- within(nsame_nstrongdta2, {
  PredictedProb <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit))
  UL <- plogis(fit + (1.96 * se.fit))
})
```

```
ggplot(nsame_nstrongdta2, aes(x = qual, y = PredictedProb)) +
  geom_line() +
  geom_errorbar(aes(ymin = LL, ymax = UL),
                position = position_dodge(width = 0.9), color="gray") +
  ylim(0, 1) +
  labs(x = "Perceived Qualifications",
```

```
      y = "Probability of Approving the Nominees") +
  ggtitle("Predicted Probabilities over the Range of Perceived Qualifications\nWhen sameprty = 0 and nu
  theme_bw() +
  theme(legend.justification = c(.7,1),
        legend.position = c(.9,.3))
```



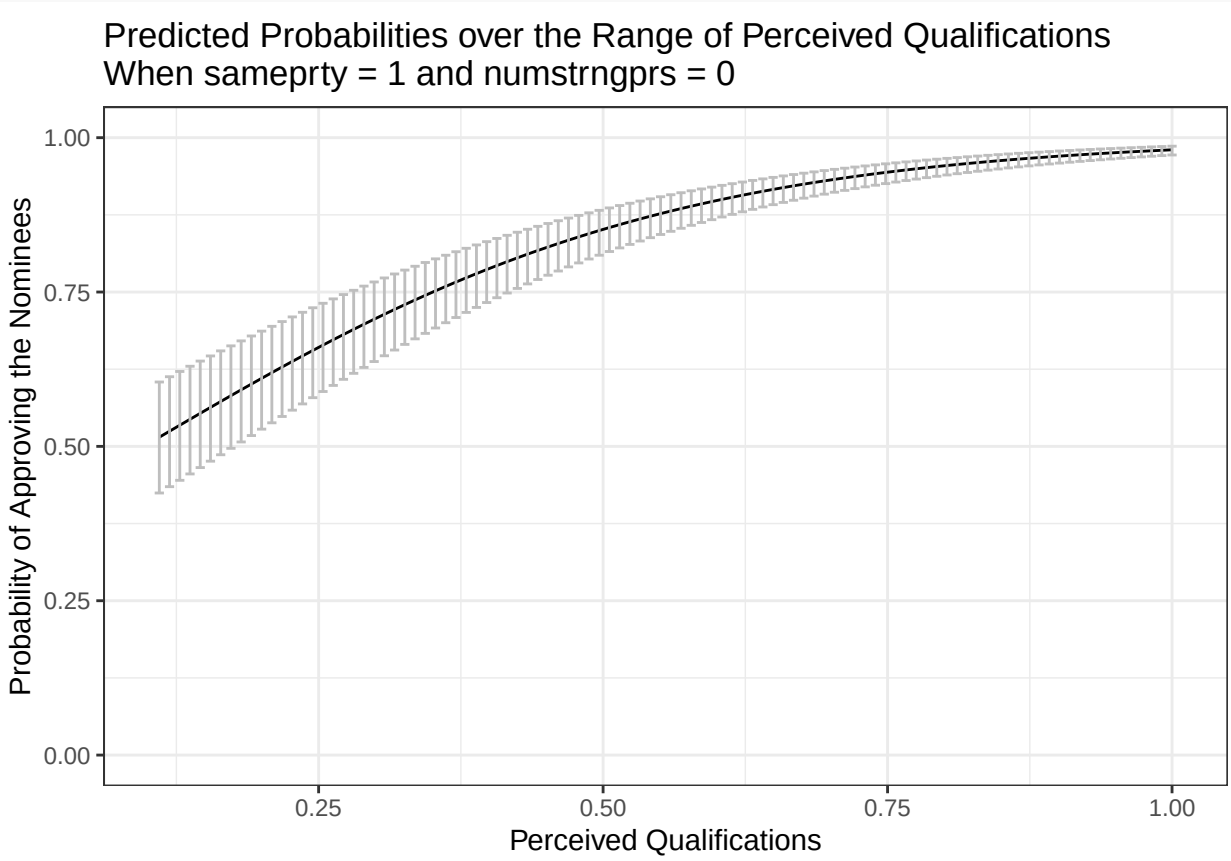
Discussion The two predicted probabilities plots above suggest that, over a range of perceived qualification values, when a senator shares the president's party affiliation and when the president is strong, the probabilities of approving the Supreme Court nomination is much higher than when a senator does not share the president's party affiliation and when the president is not strong. The most interesting takeaway from the two plots is when the perceived qualifications are low (below 0.5), the predicted probabilities of approval is much lower than when a senator does not share the president's party affiliation. The result does resonate with Epstein et al. and conventional wisdom that ideology plays an important role in the voting. However, as the perceived qualification values increases, the probabilities of approval also increase. This also comports with Epstein et al.'s findings that a candidate's merit is still an important factor to the success of the nomination.

When a senator does not share the president's party affiliation (sameprty = 1) and the president is not strong (numstrngprs = 0)

```
same_nstrongdta <- with(conf, data.frame(qual = seq(from = min(qual), to = max(qual),
                                         length.out = 100),
                       EuclDist2 = mean(EuclDist2),
                       # set sameparty to 1
                       sameprty = 1,
                       # set numstrongpres to 0
                       numstrngprs = 0)
                )
same_nstrongdta2 <- cbind(same_nstrongdta, predict(logit,
```

```
                                    newdata = same_nstrongdta,
                                    type = "link",
                                    se = TRUE))
# Calculate confidence intervals
same_nstrongdta2 <- within(same_nstrongdta2, {
  PredictedProb <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit))
  UL <- plogis(fit + (1.96 * se.fit))
})
```

```
ggplot(same_nstrongdta2, aes(x = qual, y = PredictedProb)) +
  geom_line() +
  geom_errorbar(aes(ymin = LL, ymax = UL),
                position = position_dodge(width = 0.9), color="gray") +
  ylim(0, 1) +
  labs(x = "Perceived Qualifications",
       y = "Probability of Approving the Nominees") +
  ggtitle("Predicted Probabilities over the Range of Perceived Qualifications\nWhen sameprty = 1 and nu
  theme_bw() +
  theme(legend.justification = c(.7,1),
        legend.position = c(.9,.3))
```

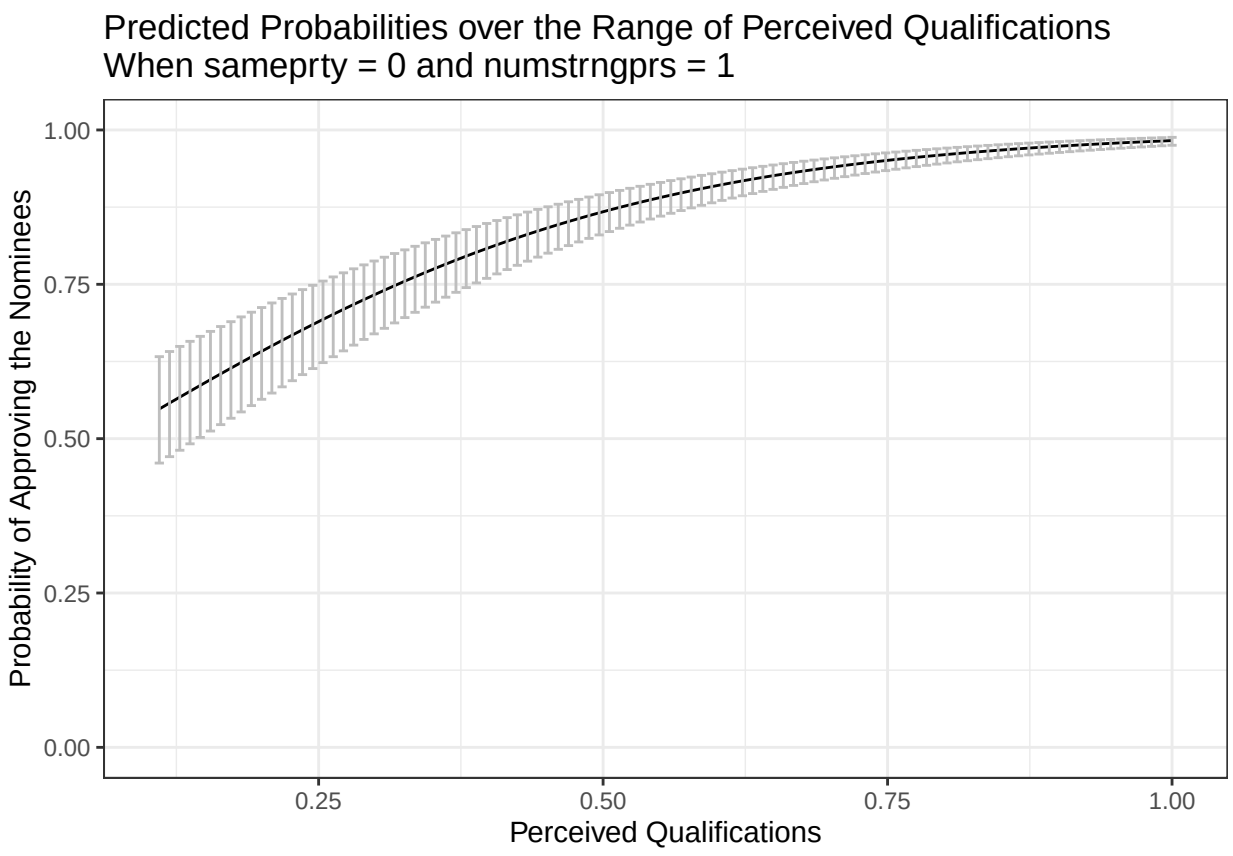


When a senator does not share the president's party affiliation (sameprty = 0) and the president is not strong (numstrngprs = 1)

```
nsame_strongdta <- with(conf, data.frame(qual = seq(from = min(qual), to = max(qual),
                                    length.out = 100),
                        EuclDist2 = mean(EuclDist2),
```

```
                        # set sameparty to 0
                        sameprty = 0,
                        # set numstrongpres to 1
                        numstrngprs = 1)
                )
nsame_strongdta2 <- cbind(nsame_strongdta, predict(logit,
                                    newdata = nsame_strongdta,
                                    type = "link",
                                    se = TRUE))
# Calculate confidence intervals
nsame_strongdta2 <- within(nsame_strongdta2, {
  PredictedProb <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit))
  UL <- plogis(fit + (1.96 * se.fit))
})
```

```
ggplot(nsame_strongdta2, aes(x = qual, y = PredictedProb)) +
  geom_line() +
  geom_errorbar(aes(ymin = LL, ymax = UL),
                position = position_dodge(width = 0.9), color="gray") +
  ylim(0, 1) +
  labs(x = "Perceived Qualifications",
       y = "Probability of Approving the Nominees") +
  ggtitle("Predicted Probabilities over the Range of Perceived Qualifications\nWhen sameprty = 0 and nu
  theme_bw() +
  theme(legend.justification = c(.7,1),
        legend.position = c(.9,.3))
```



Discussion The last two plots on predicted probabilities do not look significantly different. It seems that whether or not a senator shares the president's party affiliation when the president is strong (or not strong) both results in quite high probabilities of approval even though the perceived qualification values are not high. However, the predicted probabilities do increase as the perceived qualification values increase as we expected.

Question 5: Discussion on the Findings

According to the logistic regression model which offers a more interpretable result, we found that all of the explanatory variables, namely the perceived qualifications, the ideological distance, the same party dummy and the strength of the president dummy, contribute to the accuracy of the prediction of the yea or nay votes as they are all statistically significant. As mentioned earlier, we see that the ideological distance negatively correlates with the probability of approval while the remaining variables positively correlate with the probabilities. The effect of a nominee's prefessional merit measured by perceived qualification values is analyzed via the predicted probabilities plots. Given the average ideological distance between the nominee and a senator, I found that when a senator shares the president's party affiliation and when the president is strong, the probabilities of approving the Supreme Court nomination is much higher than when a senator does not share the president's party affiliation and when the president is not strong. This is true when the perceived qualification values are low too.

Looking at the conditional effect on perceived qualifications plots below, I see the same pattern as in the plots that I generated above. Regardless of whether or not the president is strong, if a senator shares the president's party affiliation at a certain level of perceived qualification value, the predicted probabilities of approval is always higher. This interpretation suggests that ideology plays a role in the Supreme Court nomination voting as the conventional wisdom and what Eptein et al. found. However, as Epstein et al. point out, if the nominee is highly qualified, the predicted probabilities of being approved is higher than when they are unqualified regardless of whether or not a senator shares the president's party affiliation.

Question 6: Bonus - Logit Predicted Probabilities Plot (Heterogenoeus)

Firstly, I fitted a new logistic regression model on the train dataset. The main difference between this new model and the previous one is that this model includes an interaction term between perceived qualifications and whether or not a senator shares the president's party affiliation. This allows us to see the heterogeneous effect on party affiliation.

```
logitx <- glm(numvote ~ EuclDist2 + qual + numstrngprs + sameprty +
                    qual*sameprty,
                    family = binomial(link=logit),
                    data = train_conf); summary(logitx)
```

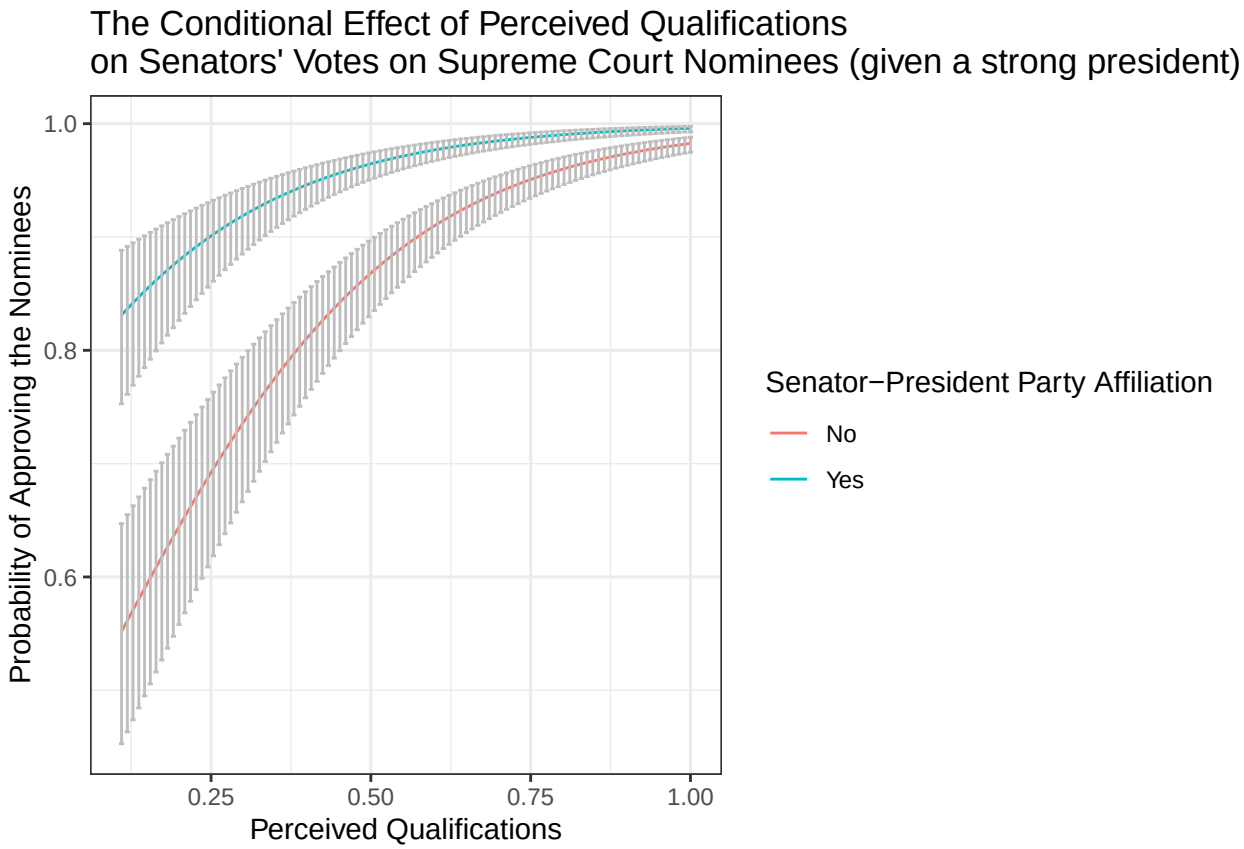
```
##
## Call:
## glm(formula = numvote ~ EuclDist2 + qual + numstrngprs + sameprty +
##     qual * sameprty, family = binomial(link = logit), data = train_conf)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2736   0.0830   0.1905   0.3903   2.1482
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.07630    0.24210  -4.446 8.76e-06 ***
## EuclDist2     -4.09539    0.31959 -12.815  < 2e-16 ***
## qual           4.29824    0.30496  14.094  < 2e-16 ***
## numstrngprs    1.55893    0.15500  10.058  < 2e-16 ***
```

```
## sameprty       1.37561    0.34906   3.941 8.12e-05 ***
## qual:sameprty  0.08623    0.52533   0.164     0.87
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2189.1  on 2966  degrees of freedom
## Residual deviance: 1345.8  on 2961  degrees of freedom
## AIC: 1357.8
##
## Number of Fisher Scoring iterations: 7
```

Note that the coefficient of the interaction term is not statistically significant.

```
interact_strong <- with(conf, data.frame(qual = rep(seq(from = min(qual), to = max(qual),
                                    length.out = 100), 2),
                                    sameprty = rep(0:1, each = 100),
                                    EuclDist2 = mean(EuclDist2),
                                    # set numstrngprs = 1
                                    numstrngprs = 1)
                )

interact_strong2 <- cbind(interact_strong, predict(logitx,
                                    newdata = interact_strong,
                                    type = "link",
                                    se = TRUE))

interact_strong2 <- within(interact_strong2, {
  PredictedProb <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit))
  UL <- plogis(fit + (1.96 * se.fit))
})

interact_strong2$sameprty <- factor(interact_strong2$sameprty, labels=c("No", "Yes"))
```

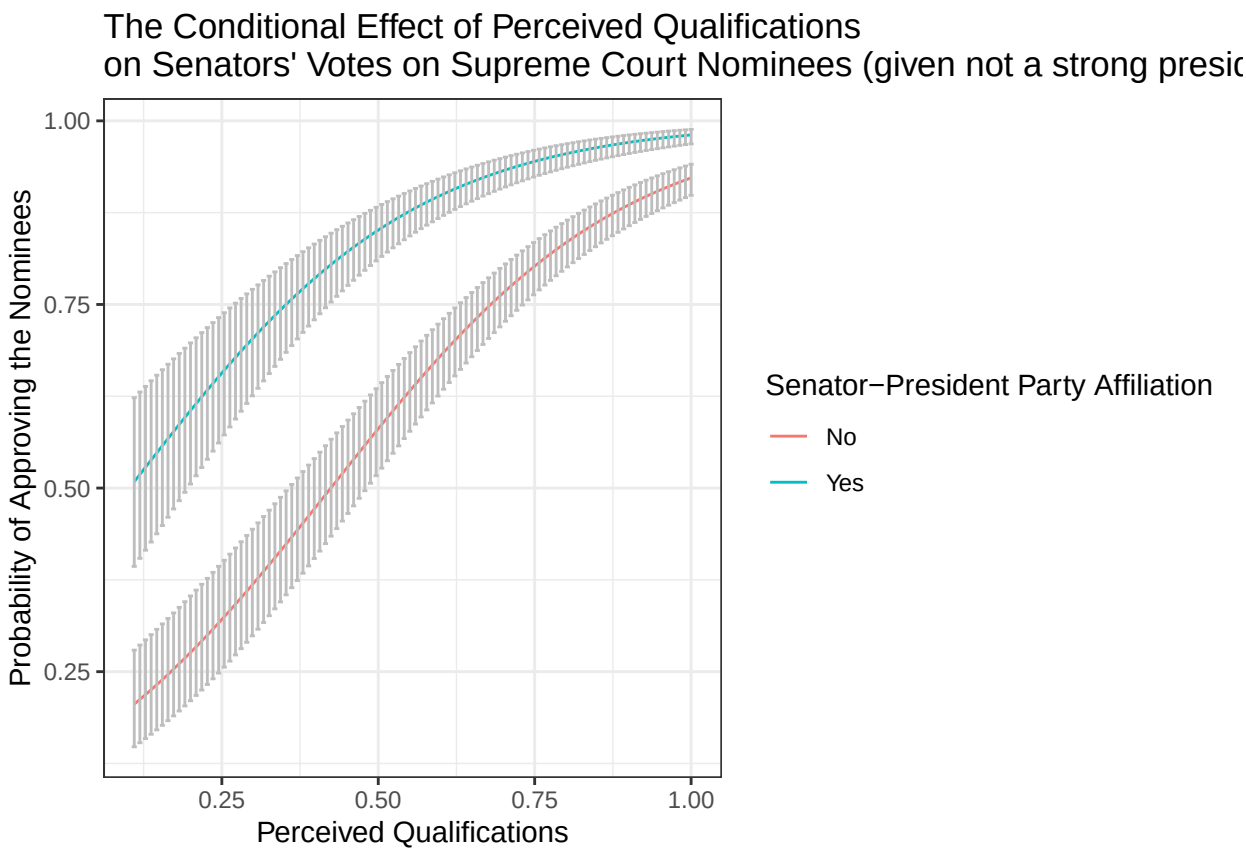
```
ggplot(interact_strong2, aes(x = qual, y = PredictedProb, color = sameprty)) +
  geom_line() +
  geom_errorbar(aes(ymin = LL, ymax = UL),
                color="gray",
                position = position_dodge(.9)) +
  labs(x = "Perceived Qualifications",
       y = "Probability of Approving the Nominees",
       color = "Senator-President Party Affiliation") +
  scale_fill_hue(breaks = c("No", "Yes"),
                 labels = c("No",  "Yes")) +
  ggtitle("The Conditional Effect of Perceived Qualifications\non Senators' Votes on Supreme Court Nomi
  theme_bw()
```




```
interact_nstrong <- with(conf, data.frame(qual = rep(seq(from = min(qual), to = max(qual),
                                    length.out = 100), 2),
                                    sameprty = rep(0:1, each = 100),
                                    EuclDist2 = mean(EuclDist2),
                                    # set numstrngprs = 0
                                    numstrngprs = 0)
                  )

interact_nstrong2 <- cbind(interact_nstrong, predict(logitx,
                                    newdata = interact_nstrong,
                                    type = "link",
                                    se = TRUE))

interact_nstrong2 <- within(interact_nstrong2, {
  PredictedProb <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit)) # LL lower level
  UL <- plogis(fit + (1.96 * se.fit)) # UL uppr level
})

# Recode usparty as a factor
interact_nstrong2$sameprty <- factor(interact_nstrong2$sameprty, labels=c("No", "Yes"))
```

```
ggplot(interact_nstrong2, aes(x = qual, y = PredictedProb, color = sameprty)) +
  geom_line() +
  geom_errorbar(aes(ymin = LL, ymax = UL),
                color="gray",
                position = position_dodge(.9)) +
```

```
  labs(x = "Perceived Qualifications",
       y = "Probability of Approving the Nominees",
       color = "Senator-President Party Affiliation") +
  scale_fill_hue(breaks = c("No", "Yes"),
                 labels = c("No",  "Yes")) +
  ggtitle("The Conditional Effect of Perceived Qualifications\non Senators' Votes on Supreme Court Nomi
  theme_bw()
```



Note: I discussed the result of the plots in the previous question.

Part 2: Unfolding

Load and prepare the dataset

I obtained the 113th U.S. House roll call data in the ORD format from Voteview.

```
house113 <- readKH(
    "hou113kh.ord", # locate the .ord file saved locally
    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
```

A brief inspection of the data.

```
summary(house113)
```

```
##
## Summary of rollcall object house113
##
## Description:  113th_House_Roll_Call_Data
## Source:       hou113kh.ord
##
## Number of Legislators:        445
## Number of Roll Call Votes:    1202
##
##
## Using the following codes to represent roll call votes:
## Yea:      1 2 3
## Nay:      4 5 6
## Abstentions:  7 8 9
## Not In Legislature:   0
##
## Party Composition:
##   D   R
## 205 240
##
## Vote Summary:
##                  Count Percent
## 0 (notInLegis)  14576     2.7
## 1 (yea)        295753    55.3
## 6 (nay)        202943    37.9
## 7 (missing)       290     0.1
## 9 (missing)     21328     4.0
##
## Use summary(house113,verbose=TRUE) for more detailed information.
```

Question 1: DW-NOMINATE Algorithm and Results

I decided to initialize the algorithm with Rep. Lynn Westmoreland from GA who was rated as most conservative in the 113th Congress by govtrack. I first inspected `house113$legis.data` and saw that Rep. Westmoreland has the `icpsrLegis` code as 20506. I used this information to grab his index.

```
which(house113$legis.data$icpsrLegis == 20506)
```

```
## [1] 121
```

```
result <- wnominate(house113,
                    dims = 2,
                    minvotes = 20,
```

```
                    lop = 0.025,
                    polarity = c(121, 121))
```

```
##
## 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 149.786 seconds to execute.
```

```
summary(result)
```

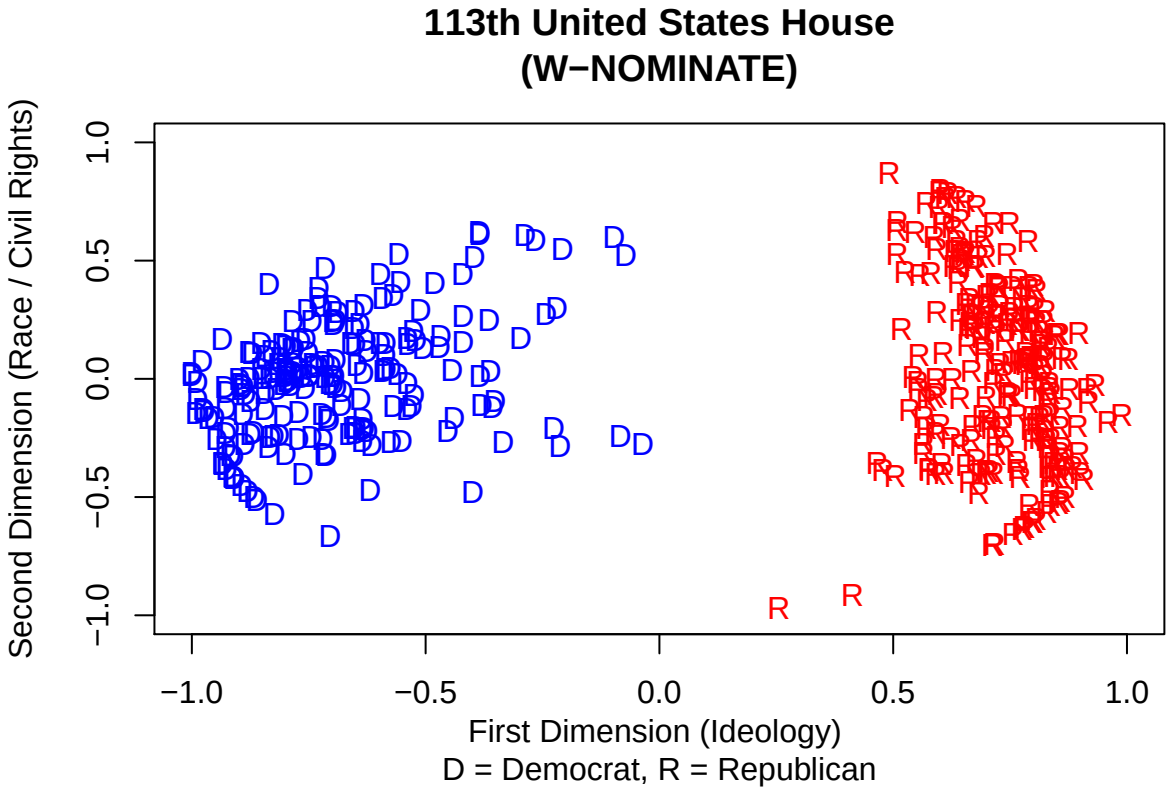
```
##
##
## SUMMARY OF W-NOMINATE OBJECT
## ----------------------------
##
## Number of Legislators:     444 (1 legislators deleted)
## Number of Votes:   1021 (181 votes deleted)
## Number of Dimensions:      2
## Predicted Yeas:        212927 of 225718 (94.3%) predictions correct
## Predicted Nays:        185010 of 199413 (92.8%) predictions correct
## Correct Classifiction:     92.79% 93.6%
## APRE:              0.817 0.837
## GMP:           0.84 0.857
##
##
## The first 10 legislator estimates are:
```

```
##                     coord1D coord2D
## OBAMA (D USA)        -0.936   0.171
## BONNER (R AL-1)       0.642   0.556
## BYRNE (R AL-1)        0.811   0.205
## ROBY (R AL-2)         0.636   0.772
## ROGERS (R AL-3)       0.724   0.393
## ADERHOLT (R AL-4)     0.678   0.735
## BROOKS (R AL-5)       0.792  -0.007
## BACHUS (R AL-6)       0.632   0.541
## SEWELL (D AL-7)      -0.560   0.024
## YOUNG (R AK-1)        0.565  -0.311
```

Visual Inspection of the Result

```
wnomd1 <- result$legislators$coord1D
wnomd2 <- result$legislators$coord2D
party <- house113$legis.data$party
```

```
plot(wnomd1, wnomd2,
     main="113th United States House\n(W-NOMINATE)",
     xlab="First Dimension (Ideology) \nD = Democrat, R = Republican",
     ylab="Second Dimension (Race / Civil Rights)",
     xlim=c(-1,1), ylim=c(-1,1), type="n")
points(wnomd1[party=="D"], wnomd2[party=="D"], pch="D", col="blue") #pch = type of plot
points(wnomd1[party=="R"], wnomd2[party=="R"], pch="R", col="red")
```



Discussion Based on my eyeball inspection, the separation (distance) between the Democrat and the Republic members of the 113th U.S. House of Representatives is clear. While the members of the House from the Democrats party are more spread out in terms of the liberal-conservative ideology scale, it looks like they are not as spread out in terms of race and civil rights aspects. On the contrary, the Republican representatives

are more clustered on the ideology scale, but they are fairly spread out on the second dimension with two outliers who are considered very liberal on the second dimension (more so than the Democrats). I found that the two outliers are Rep. Walter B. Jones from North Carolina and Rep. Chris Gibson from New York. According to his obituaries, Rep. Jones were not an orthodox conservative and had records of voting against the party line before. Without having looked more closely into the two outliers, I believe that the NOMINATE algorithm has captured the differences in ideaology between the members of the House quite well though I cannot confidently conclude so.

```
which(wnomd1 > 0 & wnomd1 < 0.5 & wnomd2 < -0.5)
```

```
## [1] 284 295
```

```
house113$legis.data[c(284, 295), ]
```

```
##                  state icpsrState cd icpsrLegis party partyCode
## GIBSON (R NY-19)    NY         13 19      21156     R       200
## JONES (R NC-3)      NC         47  3      29546     R       200
```

```
which(wnomd1 >= -0.2 & wnomd1 < 0)
```

```
## [1] 130 225 299 410
```

```
house113$legis.data[c(130, 225, 299, 410), ]
```

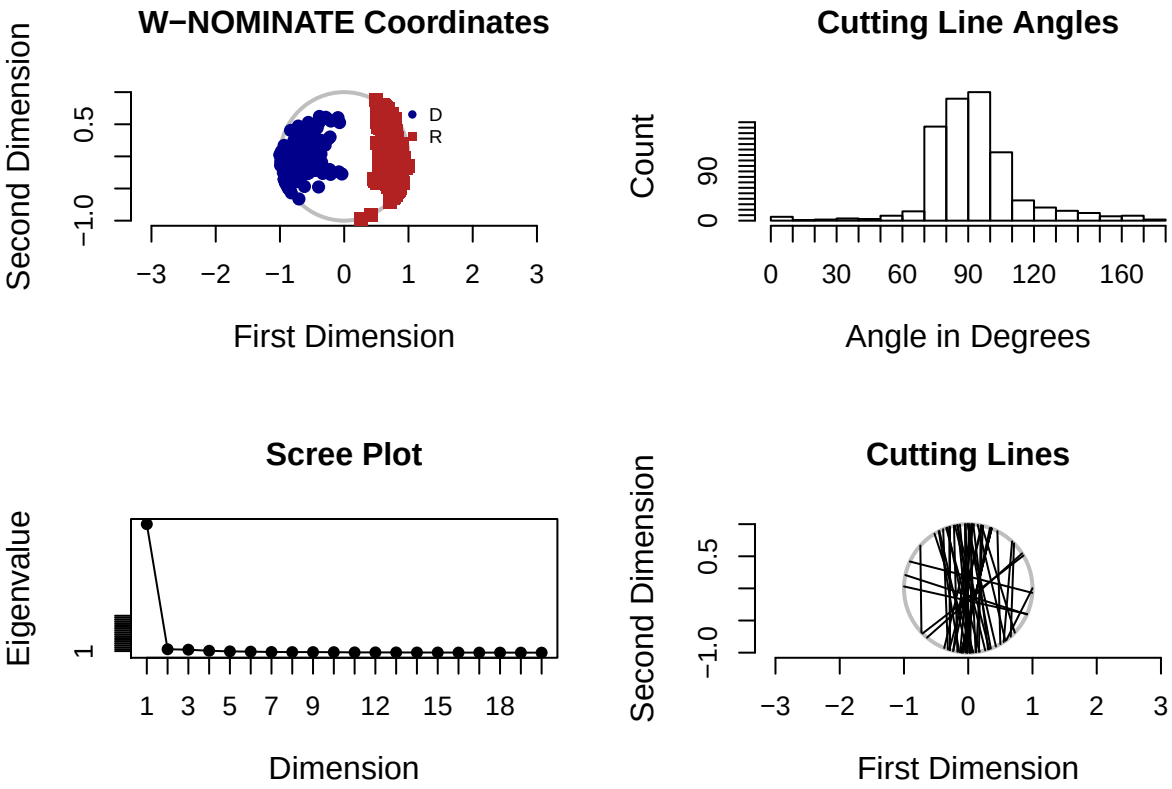
```
##                   state icpsrState cd icpsrLegis party partyCode
## BARROW (D GA-12)     GA         44 12      20507     D       100
## PETERSON (D MN-7)    MN         33  7      29127     D       100
## MCINTYRE (D NC-7)    NC         47  7      29746     D       100
## MATHESON (D UT-4)    UT         67  4      20140     D       100
```

Question 2: Space Dimensionality

I inspected the dimensionality of the space by looking at the pre-made plots provided by the package. The scree plot indicates that the first dimension does capture the majority of the variance in the data. Although the eigenvalue is close to 1 on the second dimension, the plot suggests that the second dimension does pick up an important information in the dataset.

In terms of accuracy, I looked at the summary table provided by the package as well. The summary indicates that 94.3 percent of the yea votes were predicted correctly and 92.8 percent of the nay votes were predicted correctly. The aggregate proportional reduction in errors (0.817 for yeas and 0.837 for nays) and the geometric mean prediction (0.84 for yeas and 0.857 for nays) are considered high as well. These numbers indicate that the NOMINATE model fits the data well.

```
plot(result)
```




```
## NULL
```

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

Question 3: Discussion on Major Unfolding Approaches

Methods for unfolding binary choice data that were discussed in class are NOMINATE, Item Response Theory, and nonparametric method (optimal classification). Poole (2000) proposes a nonparametric method for unfolding binary data. The method is nonparametric because it makes no assumptions about the probability distribution of the errors legislators made when making choices. The main difference between this method and NOMINATE and IDEAL is that it maximizes correct classification rather than maximizing the likelihood of the legislators' choices. As we did not discuss the nonparametric method in great details, I will focus my comparison on the NOMINATE approach and the IRT approach, the most well-recognized one being Clinton, Jackman, and Rivers' IDEAL.

According to Carroll et al. (2009), NOMINATE and IDEAL generally produce similar estimates although there are differences in the choice of utility function, scale, and the implementation of algorithms. While NOMINATE assumes Gaussian utility function, IDEAL uses quadratic utility function. Moreover, IDEAL is a Bayesian framework, but NOMINATE uses maximum likelihood estimation. Carroll et al. (p. 565) found that due to arbitrary choices of scale, ideal points estimated by IDEAL for members located at both ends of the continuum are less precise than those of members in the middle. In contrast, NOMINATE is more accurate in identifying the location of extremists than centrists. However, having used Monte Carlo experiments, the authors discover that the disadvantage of both methods is that they suffer from high variance in ideal point estimates when the number of voters is small (fewer observations). This finding speaks to the limitation of both NOMINATE and IDEAL approaches in general that they perform better when there are more data points and will not produce starkly different estimates.

Carroll, Royce, Jeffrey B. Lewis, James Lo, Keith T. Poole, and Howard Rosenthal. "Comparing NOMINATE and IDEAL: Points of Difference and Monte Carlo Tests." Legislative Studies Quarterly 34, no. 4 (2009):

555–91. https://doi.org/10.3162/036298009789869727.

Poole, Keith T. “Nonparametric Unfolding of Binary Choice Data.” Political Analysis 8, no. 3 (2000): 211–37. https://doi.org/10.1093/oxfordjournals.pan.a029814.