CMAP: Problem Set 2

Simón Hicks-Prieto

November 11, 2019

Part 1: Classification

Part 1: Loading Data and Libraries

```
library(tinytex)
library(Matrix)
library(lme4)
library(foreign)
library(MASS)
library(class)
library(faraway)
library(arm)
```

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

## Working directory is /Users/simonprieto

##
## Attaching package: 'arm'

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

```
library(ggplot2)
library(OOmisc)
library(pROC)
```

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

##
## Attaching package: 'pROC'

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

```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'lattice'

## The following object is masked from 'package:faraway':
##
##     melanoma
```

```
data <- read.dta("~/Downloads/problem-set-2-master/PSET 2 Files/conf06.dta")

conf06 <- subset(data, data$nominee!="ALITO")
```

```r
vars <- c("vote", "nominee", "sameprty", "qual", "lackqual", "EuclDist2", "strngprs")
conf <- conf06[vars]
conf$numvote <- as.numeric(conf$vote)-1 # from 1/2 to 0/1
conf$numstrngprs <- as.numeric(conf$strngprs)-1 # same as above
```

Question 1.1: Building an 80/20 Split to Test and Train Set

```r
#80/20 SPLIT
samples <- sample(1:nrow(conf),
                  nrow(conf)*0.8,
                  replace = FALSE)

#TESTING/TRAINING
train <- conf[samples, ]
test <- conf[-samples, ]

#STORING TEST SET
vote <- test$numvote
```

Question 1.2: Building a Logit Classifier

```r
vote <-test$numvote


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

```
##
## Call:
## glm(formula = vote ~ sameprty + numstrngprs + qual + lackqual +
##     EuclDist2, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3248   0.0821   0.1838   0.3802   2.0375
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.084e+07  5.179e+06   4.024 5.72e-05 ***
## sameprty     1.389e+00  1.713e-01   8.111 5.02e-16 ***
## numstrngprs  1.734e+00  1.699e-01  10.202  < 2e-16 ***
## qual        -2.084e+07  5.179e+06  -4.024 5.72e-05 ***
## lackqual    -2.084e+07  5.179e+06  -4.024 5.72e-05 ***
## EuclDist2   -4.161e+00  3.240e-01 -12.841  < 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: 2149.1  on 2966  degrees of freedom
## Residual deviance: 1321.7  on 2961  degrees of freedom
```

```
## AIC: 1333.7
##
## Number of Fisher Scoring iterations: 6
```

```
#here we are inserting a new argument, saying newdata=test
logit.probs <- predict(logit,
                       newdata = test,
                       type="response")

head(logit.probs)
```

```
##        19        20        21        23        53        54
## 0.8915690 0.8850415 0.8587344 0.8716711 0.8626460 0.8143364
```

```
logit.pred <- ifelse(logit.probs > 0.5, 1, 0)

table(logit.pred,vote)
```

```
##           vote
## logit.pred   0   1
##          0  42   9
##          1  53 638
```

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

```
## [1] 0.916442
```

Four main observations stand out from the output– the logit classifier was highly accurate, all variables were statistically significant, qualifications of the nominee have the highest coefficient estimates, and partisanship is influential in the voting process.

Starting with the accuracy of the logit classifier, over 92% of votes were correctly classified (given that the mean was 0.9285714). There were relatively few false-negative (46 in total) and false-positive (7) classifications in comparison to the amount of correctly classified votes (689).

Second, all the variables used in the model were statistically significant (given that all had a p-value that was far less than 0.05).

Third, according to this logit classifier, the strongest factor in the decision making of the legislators was the perceived qualifications (or lack thereof) of the nominee. For this variable, the coefficient estimate was -2.183 x 10ˆ7, meaning for every one unit decrease in lack of percieved qualifications there was a 2.183 x 10ˆ7 decrease in the log likelihood of the nominee being approved. However, it is worth noting that the "qual" variable and the "lackqual" variable also have by far the highest standard errors (both have 5.117 x 10ˆ6 Standard Error Score). Thus, we should take the effect of qualifications on Supreme Court nomination with a grain of salt.

On the opposite end of the spectrum, the variables sameprty and strngprs have the smallest coefficient estimates (1.454e+00 and 1.692e+00, respectively), but these variables also the smallest standard of errors (1.705e-01 for sameprty and 1.647e-01 for strngprs). Both variables can be interpreted as proxy measures for partisanship, which suggests that the percieved partisanship of the nominee, and the degree to which that matches the legislators, has a small but certain effect on nomination process.

Question 1.3: Building LDA Classifier

```
lda <- lda(vote ~ nominee + sameprty + qual + lackqual + EuclDist2 + numstrngprs,
           data=train)
```

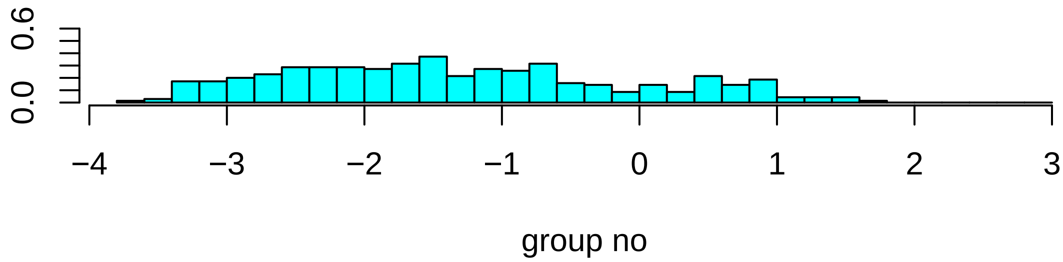
```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

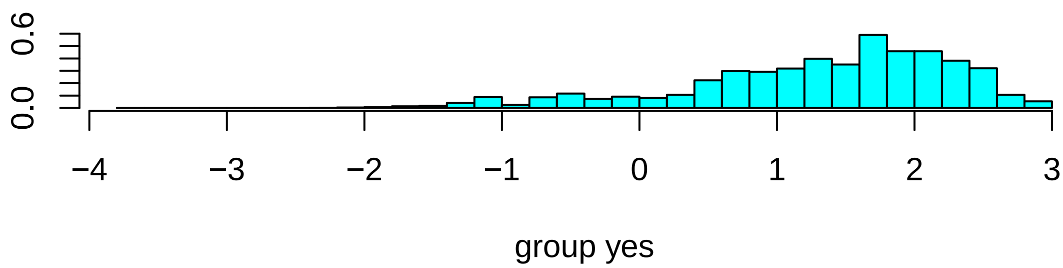
```r
lda
```

```
## Call:
## lda(vote ~ nominee + sameprty + qual + lackqual + EuclDist2 +
##     numstrngprs, data = train)
##
## Prior probabilities of groups:
##        no       yes
## 0.1176272 0.8823728
##
## Group means:
##     nomineeBLACKMUN nomineeBORK nomineeBRENNAN nomineeBREYER nomineeBURGER
## no       0.00000000  0.14899713     0.00000000    0.01432665   0.008595989
## yes      0.02902979  0.01298701     0.03132162    0.02788388   0.022154316
##     nomineeBURTON nomineeBYRNES nomineeCARSWELL nomineeCLARK
## no     0.00000000    0.00000000      0.11461318   0.02005731
## yes    0.03055768    0.02979374      0.01375095   0.02368220
##     nomineeDOUGLAS nomineeFORTAS1 nomineeFORTAS2 nomineeFRANKFURTER
## no     0.008595989     0.00000000     0.10028653         0.00000000
## yes    0.017952636     0.03055768     0.01375095         0.03055768
##     nomineeGINSBURG nomineeGOLDBERG nomineeHARLAN nomineeHAYNSWORTH
## no      0.008595989      0.00000000     0.0286533        0.10888252
## yes     0.028647823      0.03132162     0.0210084        0.01489687
##     nomineeJACKSON nomineeKENNEDY nomineeMARSHAL nomineeMINTON
## no      0.00000000     0.00000000     0.02578797    0.03724928
## yes     0.02750191     0.03055768     0.02062643    0.01375095
##     nomineeMURPHY nomineeOCONNOR nomineePOWELL nomineeREED
## no     0.00000000     0.00000000    0.00286533  0.00000000
## yes    0.03246753     0.02750191    0.02597403  0.02902979
##     nomineeREHNQUIST1 nomineeREHNQUIST2 nomineeROBERTS nomineeRUTLEDGE
## no          0.0487106        0.08595989     0.03724928      0.00000000
## yes         0.0210084        0.02215432     0.02215432      0.02559206
##     nomineeSCALIA nomineeSOUTER nomineeSTEVENS nomineeSTEWART nomineeSTONE
## no     0.00000000    0.02292264     0.00000000     0.03724928   0.00000000
## yes    0.03017571    0.02788388     0.03017571     0.02291826   0.02864782
##     nomineeTHOMAS nomineeVINSON nomineeWARREN nomineeWHITE
## no     0.09742120    0.00000000    0.00000000   0.00000000
## yes    0.01604278    0.02788388    0.02979374   0.02979374
##     nomineeWHITTAKER  sameprty      qual  lackqual EuclDist2 numstrngprs
## no        0.00000000 0.1891117 0.5395129 0.4604871 0.3871803   0.2808023
## yes       0.02750191 0.5993125 0.8052884 0.1947116 0.1547041   0.6065699
##
## Coefficients of linear discriminants:
##                             LD1
## nomineeBLACKMUN     0.816174552
## nomineeBORK        -2.680982411
## nomineeBRENNAN      1.001100316
## nomineeBREYER       0.368827751
## nomineeBURGER       0.422569893
## nomineeBURTON       0.172995486
## nomineeBYRNES       0.291748753
## nomineeCARSWELL    -0.877486235
## nomineeCLARK        0.805438063
## nomineeDOUGLAS     -0.310137044
```

```
## nomineeFORTAS1       0.318406121
## nomineeFORTAS2      -2.044038769
## nomineeFRANKFURTER -0.153987373
## nomineeGINSBURG    -0.167083070
## nomineeGOLDBERG     0.171714253
## nomineeHARLAN       0.267789626
## nomineeHAYNSWORTH  -1.329578865
## nomineeJACKSON      0.496917878
## nomineeKENNEDY      0.521663707
## nomineeMARSHAL     -0.246430397
## nomineeMINTON      -0.499617092
## nomineeMURPHY       1.185593836
## nomineeOCONNOR      0.010156462
## nomineePOWELL       0.580106178
## nomineeREED        -0.054975111
## nomineeREHNQUIST1  -0.481167721
## nomineeREHNQUIST2  -0.638135811
## nomineeROBERTS     -0.863157462
## nomineeRUTLEDGE     0.633392742
## nomineeSCALIA       0.627149772
## nomineeSOUTER       0.215187849
## nomineeSTEVENS      0.721560856
## nomineeSTEWART     -0.666261309
## nomineeSTONE       -0.004129658
## nomineeTHOMAS      -1.168646880
## nomineeVINSON       0.544197751
## nomineeWARREN       0.419359344
## nomineeWHITE        0.781640878
## nomineeWHITTAKER    0.319850716
## sameprty            0.484273654
## qual                0.841551577
## lackqual           -0.841551580
## EuclDist2          -2.862623472
## numstrngprs         0.310028092
```

```r
plot(lda)
```






```
vote <- test$vote

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

data.frame(lda.pred)[1:5,]
```

```
##    class posterior.no posterior.yes        LD1
## 19   yes   0.01480472     0.9851953 -0.1790079
## 20   yes   0.01664909     0.9833509 -0.2242805
## 21   yes   0.02532336     0.9746766 -0.3868113
## 23   yes   0.02080413     0.9791959 -0.3104475
## 53   yes   0.02390395     0.9760960 -0.3643658
```

```
# presenting as confusion matrix
table <- table(lda.pred$class, vote)

table
```

```
##      vote
##        no yes
##   no   63  26
##   yes  32 621
```

```
# seeing rate of classification
mean(lda.pred$class == vote)
```

```
## [1] 0.9218329
```

93.39623% of votes were accurately classified using the LDA classifier. This LDA classifier has an accuracy (0.9339623) that is incredibly similar to the mean accuracy of the logit model (.9285). Importantly, the LDA classifier yielded less false-negatives (28) than the logit classifier (46), but LDA had more false-positives (21) than the logit classifier (7).

Here, the variable with the greatest impact on the ability to correctly classify the support for a Supreme Court nominee was the Euclidean distance between the legislator and nominee's inferred idea points. This again implies that ideology has become a powerful factor in the nomination process of Supreme Court Justices, which is certainly a frightening thought.

Question 1.4: Plotting the Conditional Effects of Percieved Qualifications

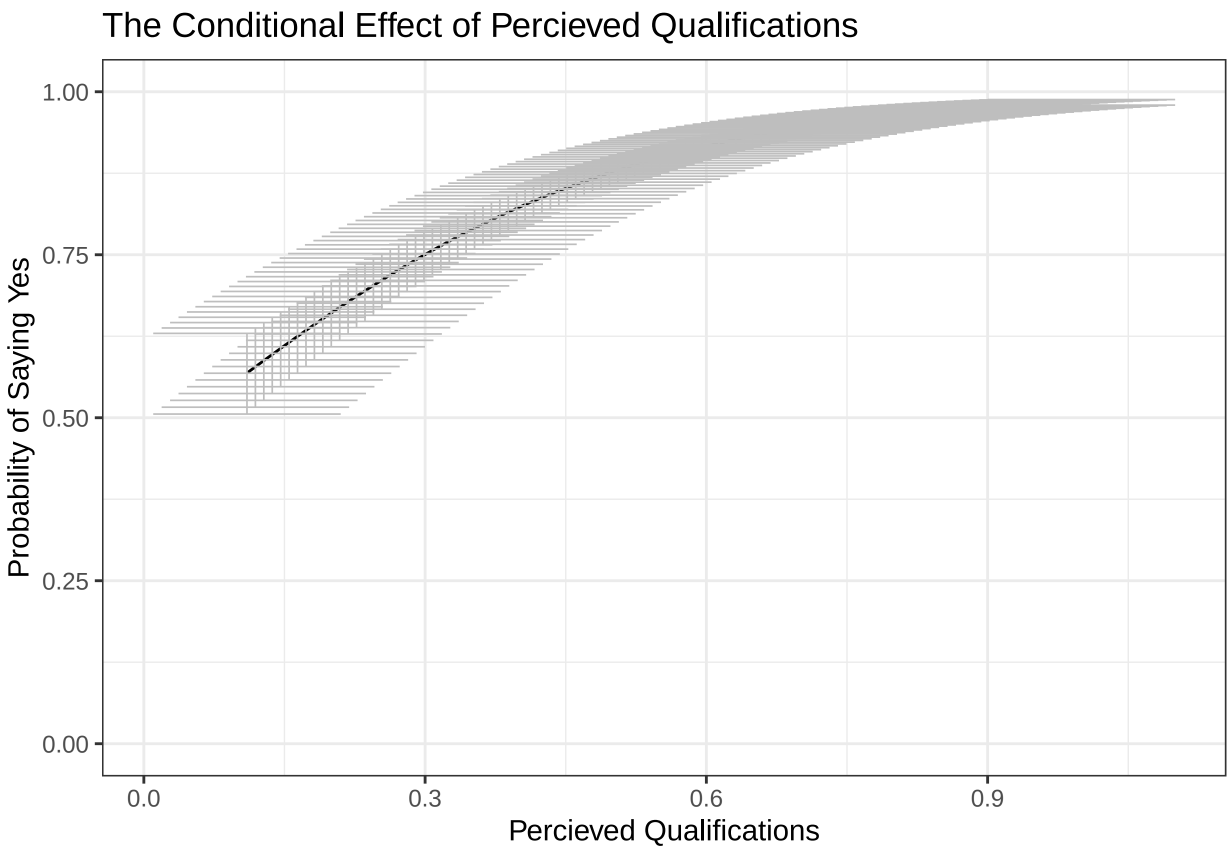
```
library(effects)
```

```
## Loading required package: carData

## Use the command
##     lattice::trellis.par.set(effectsTheme())
##   to customize lattice options for effects plots.
## See ?effectsTheme for details.
```

```
library(ggplot2)


logitmod1 <- glm(vote ~ sameprty + qual + EuclDist2 + numstrngprs,
                 family = binomial(link=logit),
                 data = conf)

# CIs for predicted probabilities. This is creating out synthetic data: let liberal2 range from 20-80,
outputdata <- with(conf, data.frame(qual = rep(seq(from = 0.11, to = 1, length.out = 100)),
                                  sameprty = mean(sameprty),
                                  numstrngprs = mean(numstrngprs),
                                  EuclDist2 = mean(EuclDist2)))

outputdata2 <- cbind(outputdata, predict(logitmod1,
                                   newdata= outputdata,
                                   type = "link",
                                   se = TRUE))


output4 <- within(outputdata2, {
  PredictedProb3 <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit))
  UL <- plogis(fit + (1.96 * se.fit))
})


# Plot predictions with CIs
ggplot(output4, aes(x = qual, y = PredictedProb3)) +
  geom_line() +
  geom_errorbar(aes(ymin = LL, ymax = UL),
                color="gray",
                size=.3,
                width=.2,
                position = position_dodge(.9)) +
  scale_fill_hue(breaks = c("No", "Yes"),
                 labels = c("No",  "Yes")) +
  labs(x = "Percieved Qualifications",
       y = "Probability of Saying Yes") + ylim(0,1) +
  ggtitle("The Conditional Effect of Percieved Qualifications") +
  theme_bw() +
  theme(legend.justification = c(.7,1),
```

```
        legend.position = c(.9,.3))
```

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



When we hold all variables other than perceived qualifications at their mean value, greater perceived qualifications increases the probability that the legislators say yes. This relationship, however, is not perfectly linear. The positive effect of increasing the perceived qualifications is more pronounced between 1.2 and 0.6, but this effect tapers off after 0.6 onward. This indicates that after legislators perceive that qualifications of the nominee surpass a certain threshold, and increase in perceived qualifications has little effect on the probability of saying yes, and correspondingly, other factors come into play in the decision-making of the legislators.

Question 1.5: Explaining Major Findings

Both the logit and LDA classifiers lead us to an important conclusion, the extreme polarization of Congress is clearly influencing the Supreme Court nomination process. For decades, the Supreme Court nomination process had been a relatively non-partisan procedure, with the majority of congressional legislators approving a nominee if the nominee was qualified and their background in public service was free of any glaring issues. This analysis, however, reveals that this era of non-partisanship in the Supreme Court nomination process appears to be over.

While these classification methods show that the nominee's percieved qualifications are still important in determining their support in the Congress, they also show that this is certainly not the only factor. Indeed, only when holding all other variables at their mean value does perceived qualifications play the dominant explanatory role that many may have expected in the Supreme Court nomination processes. In both classification approaches, the Euclidean distance between the nominee's ideal point and the legislator's appears to be highly influential in the nomination process. In the Logit Classifier, only percieved qualifications (or lack thereof) of the nominee had a greater coefficient estimate than the Euclidean Distance variable. In the LDA classifier, Euclidean Distance had by far the greatest impact in the ability to correctly predict support for the nominee. This information suggests that ideology and partisanship play a notable role in the nomination process.

Ultimately, our data reveals that the extreme polarization in Congress has made its way to the Supreme Court nomination process.

Question 1.6: Bonus

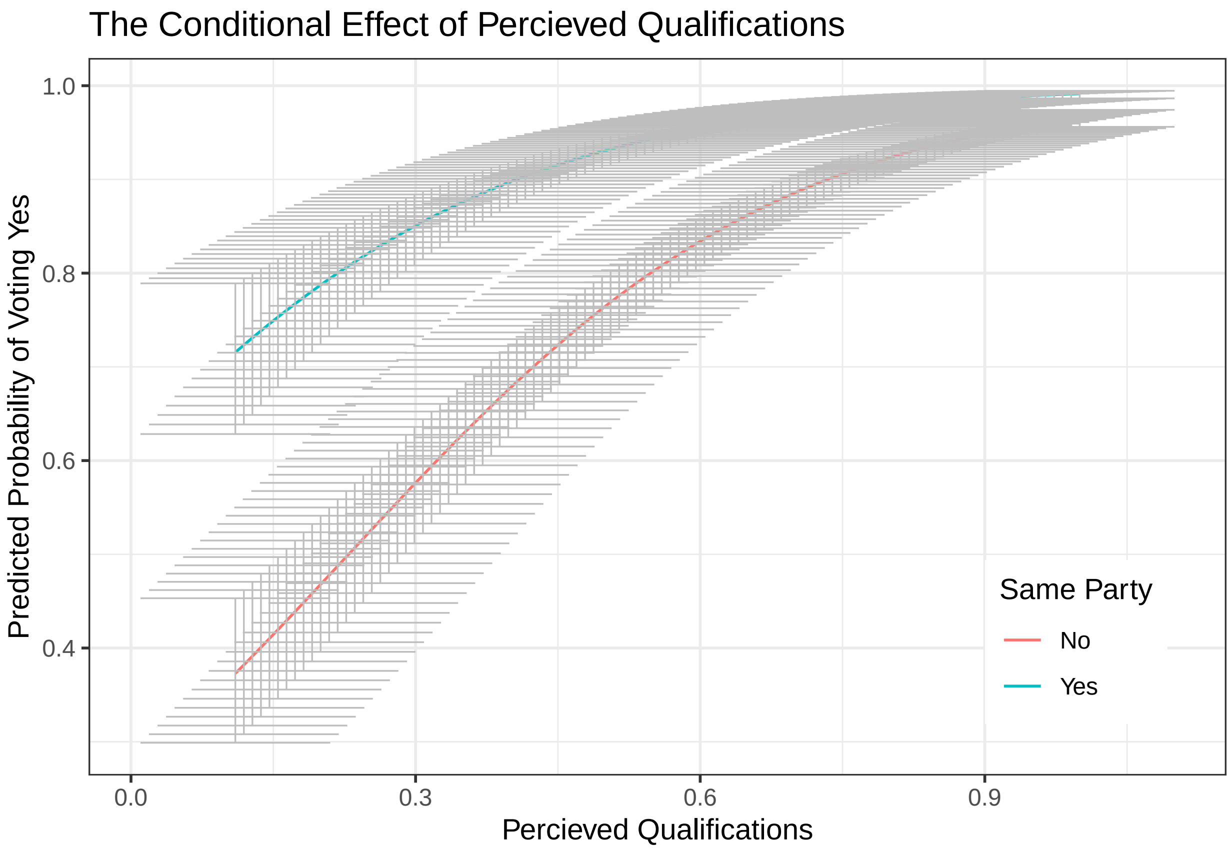
```
library(rminer)
```

```
##
## Attaching package: 'rminer'

## The following object is masked from 'package:lme4':
##
##     factorize
```

```
library(ggplot2)

logitmod7 <- glm(vote ~ sameprty + qual + EuclDist2 + numstrngprs + sameprty*qual,
                 family = binomial(link=logit),
                 data = conf)

# CIs for predicted probabilities. This is creating out synthetic data: let liberal2 range from 20-80,
DATANEW <- with(conf, data.frame(qual = rep(seq(from = 0.11, to = 1, length.out = 100), 2),
                                  sameprty = rep(0:1, each = 100),
                                  numstrngprs = mean(numstrngprs),
                                  EuclDist2 = mean(EuclDist2)))

newdata5 <- cbind(DATANEW, predict(logitmod7,
                                    newdata= DATANEW,
                                    type = "link",
                                    se = TRUE))


newdata3 <- within(newdata5, {
  PredictedProb <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit))
  UL <- plogis(fit + (1.96 * se.fit))
})



# Recode usparty as a factor
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)) +
```

```r
labs(x = "Percieved Qualifications",
     y = "Predicted Probability of Voting Yes",
     color = "Same Party") +
scale_fill_hue(breaks = c("No", "Yes"),
               labels = c("No",  "Yes")) +
ggtitle("The Conditional Effect of Percieved Qualifications") +
theme_bw() +
theme(legend.justification = c(.7,1),
      legend.position = c(.9,.3))
```

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



On average, the rate by which an increase of perceived qualifications affects the probability of saying yes differs substantially by party. When the nominee is of the same party as the legislator, the model predicts that the legislator has over a 70% chance of voting in favor of the nominee even if the nominee has the lowest perceived qualifications possible. Indeed, for legislators of the same party as the nominee, the slope of "probability of saying yes" after the nominee surpasses a threshold of 0.45 is very small, further supporting the notion that partisanship is a large factor is the decision making of the legislator when voting on a nominee. Conversely, for legislators of the opposite party as the nominee, when the perceived qualifications of the nominee increase, the predicted probability of the legislators voting yes increase tremendously (eg. when the perceived qualifications of the nominee increase from 0.3 to 0.6, the predicted probability of voting yes for a legislator of the opposite party increases around 30%!). This leads us to the conclusion that perceived qualifications matter much more to legislators of the opposite party than it does for legislators of the same party.

Finally, another interesting observation is that once the perceived qualifications of the nominee surpasses 0.9, the probability of legislators voting yes is nearly identical between parties, indicating that even the most staunchly partisan legislators vote in favor of a nominee that is indisputably qualified for the job.

Part 2: Unfolding

Part 2: Loading Data and Libraries

```
library("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
```

```
library("wnominate")
```

```
##
## ## 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
```

```
house113 <- readKH(
  "~/Downloads/problem-set-2-master/PSET 2 FILES/hou113kh.ord",
  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
```

Question 2.1: Fitting a W-NOMINATE Algorithm

```
wnom_result <- wnominate(house113,
                         dims = 2,
                         minvotes = 20,
                         lop = 0.025,
                         polarity = c(2,2))
```

```
##
## Preparing to run W-NOMINATE...
```

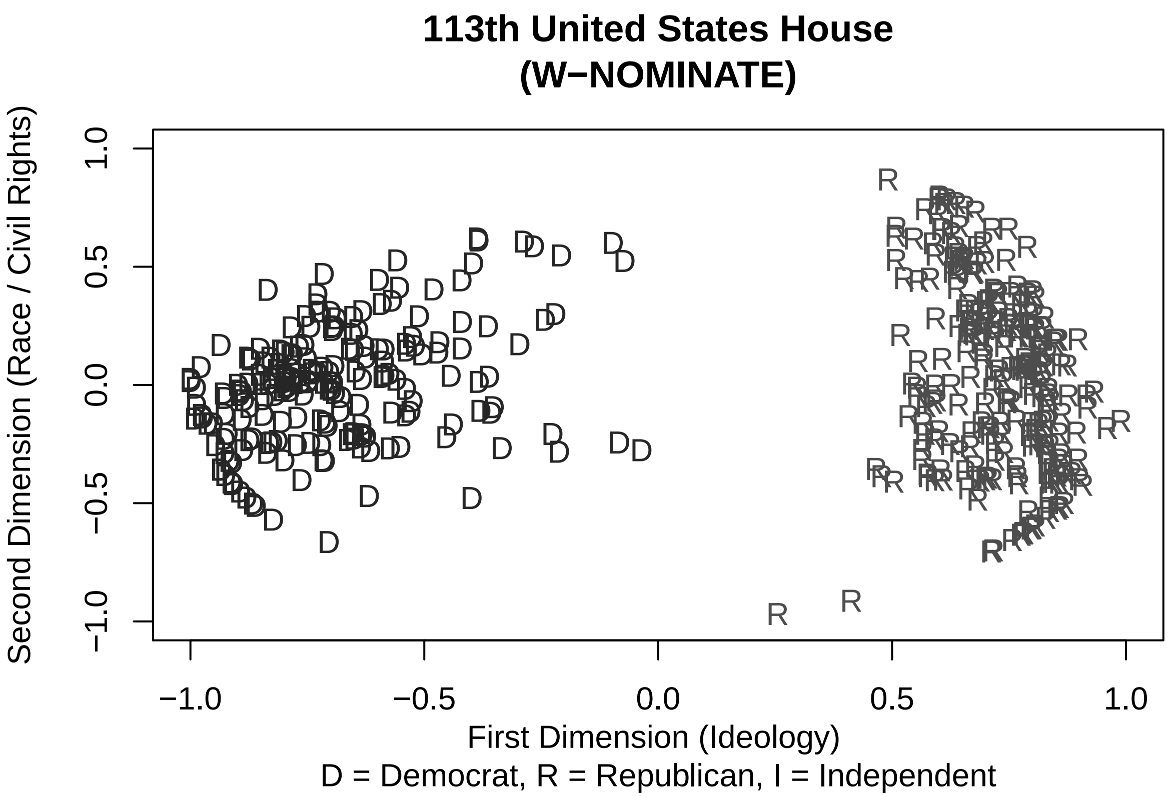
```
##
## Checking data...
##
##      ... 1 of 445 total members dropped.
##
##      Votes dropped:
##      ... 181 of 1202 total votes dropped.
##
## Running W-NOMINATE...
##
##      Getting bill parameters...
##      Getting legislator coordinates...
##      Starting estimation of Beta...
##      Getting 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 223.906 seconds to execute.
```

```
wnom1 <- wnom_result$legislators$coord1D
wnom2 <- wnom_result$legislators$coord2D
party <- house113$legis.data$party
party
```

```
##   [1] D R R R R R R R D R D D D R R R D R D R R R R R D D R D D D R D R D D
##  [36] D D D D D D D D R R R D R D D D D D R D D D D D D D R D D R D D R D D
##  [71] R R R D D D D D R R R R D D D D D D D R R R R D R R R D R R R R R D R
## [106] R R D R R D D D D D R D R R D R D D R R R R R R D D R D D R R D D D D
## [141] D R D D D D D D R R R R D R D R R R R R D R R D D R R R R R R R R D R
## [176] R R R D R R R R R D D R D D D D D D D D D D D D D D D D D R R R R D R
## [211] R R D R R D D D D R R D D R D D R D R R D R R R D R R R R R R R R D R
## [246] R D D D D D R R R R D R D D D R D D R D D R D D D D D D D D R D D D D
## [281] D D D R D D R R D D D R D R R D R R D R R R R D D R R R R D R R R R R
## [316] D R D R D R R R R R R R R D R D D D D D R R R R R R R R R R D D R R D
## [351] R D D R R R R R D R R R R R R D R R R D R R R R R R R R D R R R R R D
## [386] D R D R D R R D R R R R D D D R R D D D R R R R D D R R D R R R R R D
## [421] R R D D D R R R D D R D D R R D R D D D R R R R R
## Levels: D R
```

```
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")
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")
```



A few things stand out from this plot. First, Republicans are much more unified on the ideology dimension than democrats. This is evident based on how concentrated the republican legislators are on the ideological dimension, with only a hand-full lying outside the 0.5-1.0 range. Conversely, Democrats have substantially more ideological variation, with over 30 legislators of the Democratic party lying between the -0.5-0.0 range.

Another observation on the ideology dimension that is worth noting is there are only a couple republicans all the way at 1.0, which is the most conservative end of the spectrum, but there are several (nearly a dozen) Democratic legislators that are placed on the liberal extreme (-1.0). This contributes to the notion that there is substantially more ideological variation in the Democratic party than there is the Republican party.

On the dimension of race, Democrats are more unified than the Republican party. Indeed, for the Democrats, all of them appear to be within the range of -0.3 to 0.6, while there are over a dozen Republicans that lie outside that range on issues of race. Interestingly, however, the two most liberal legislators on the topic of race are Republicans. This is not to say that all of the Republican party is highly liberal, however, since the two most conservative legislators on the issue of race are Republicans as well. Ultimately, there is a wide variation of preferences on the issue of race for the Republican party, suggesting that Republican legislators do not appear to classify their party membership based on their preferences on issues of race.

##Question 2.2: Discussing Dimensionality

```
summary(wnom_result)
```

```
##
##
## SUMMARY OF W-NOMINATE OBJECT
## ----------------------------
```

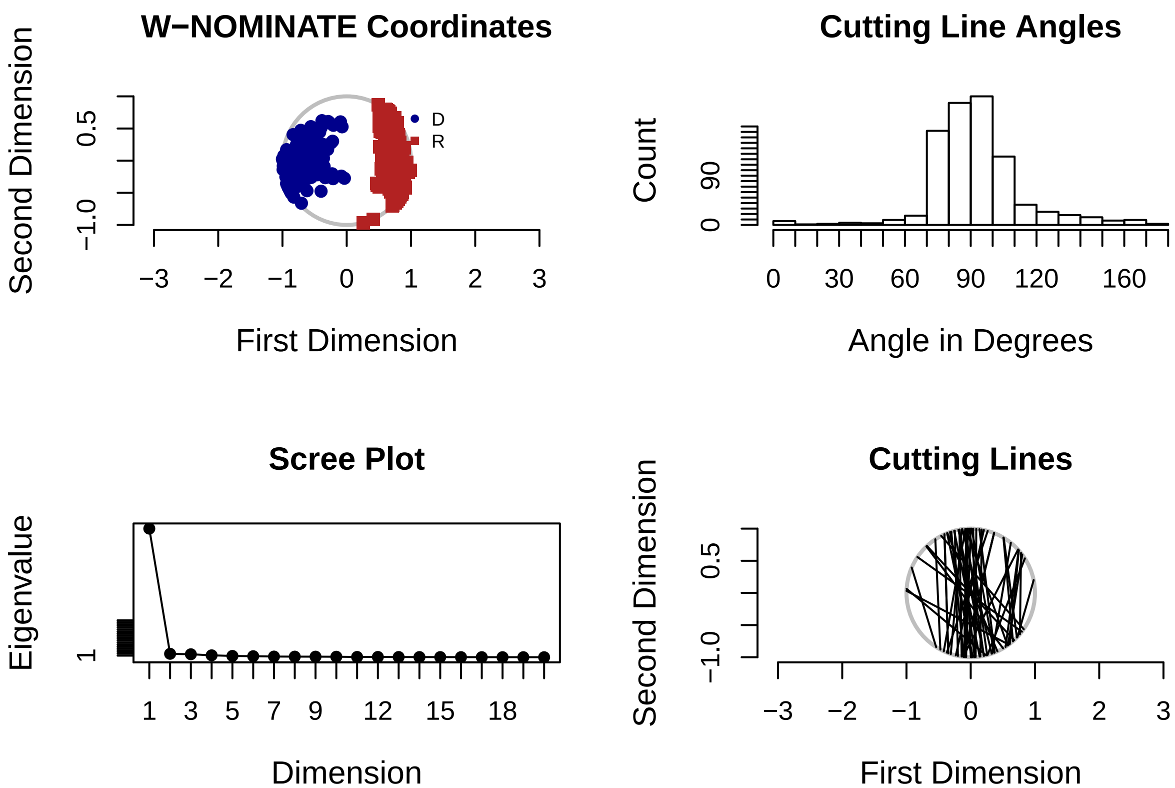
```
##
## Number of Legislators:     444 (1 legislators deleted)
## Number of Votes:   1021 (181 votes deleted)
## Number of Dimensions:      2
## Predicted Yeas:        212927 of 225718 (94.3%) predictions correct
## Predicted Nays:        185010 of 199413 (92.8%) predictions correct
## Correct Classifiction:     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
```

```r
data(wnom_result)
```

```
## Warning in data(wnom_result): data set 'wnom_result' not found
```

```r
plot(wnom_result)
```



```
## NULL
```

The scree-plot is very effective at capturing the dimensionality of the space. Based on this plot, one thing is clear– the first dimension (ideology) is by far the most influential dimension in characterizing this space. You can see here that the first dimension is nearly at the very top of the Eigenvalue scale, which conveys its dominant influence in partisanship. After the first dimension, there is a drastic drop-off. Dimension two and three have a minimal effect (each around 1 on the Eigenvalue scale), which suggests that at least ideology is not totally characterizing the dimensionality of Congress.

Question 2.3: Comparing Major Unfolding Approaches

Optimal Classification, NOMINATE and the Bayesian Approach do not have the same core assumptions. NOMINATE and the Bayesian approach are both parametric methods of unfolding, and correspondingly, they treat the vote of a legislator probabilistically. In other words, these parametric methods assume that there are various latent factors that contribute to the decision-making process behind a legislator's vote. On the other hand, the Optimal Classification method is non-parametric, and thus it does not make the same assumptions as the parametric approaches (ie. it does not treat the vote of a legislator probabilistically). Given that politics is a latent space, the assumptions made in parametric assumptions are necessary to better understanding the world of politics (ie. the Optimal Classification approach is not appropriate here).

Between the two parametric approaches discussed above (the NOMINATE and the Bayesian Approach), there is an important distinction in the utility functions assumed. The NOMINATE approach assumes a normal distribution utility function, while the Bayesian approach assumes a quadratic utility function. In other words, the NOMINATE algorithm assumes that the preferences around a legislator's ideal point are highly concentrated, and that moving away from a legislator's ideal point corresponds to a dramatic drop off in utility. This drop-off in utility is far less drastic in the quadratic utility function used by the Bayesian approach. In other words, when moving away from the legislator's ideal point, their change in utility does not drop off as quickly in a quadratic distribution model when compared to the normal distribution model used in NOMINATE algorithms.

Given this information, it becomes clear that certain scenarios favor the use of the NOMINATE algorithm, while others favor the Bayesian approach. On issues that are highly salient to a legislator's voter base, it is fair to conclude that the legislator has less lee way when voting on these issues, since their voter base expects certain legislative behavior and will punish their representative if he or she does not meet these expectations. Thus, when voting on these salient issues, the NOMINATE algorithm is better, since the sharp drop-off in the legislator's utility function as we move away from their ideal point reflects the political pressure to act in a way that appeases their voter base on this salient issue.

The Bayesian approach may be more appropriate for less salient issues and more obscure policies. When a legislator is voting on issues that their voter base either does not care about or is not educated on, such as "niche" or highly specific government regulations, then the legislator is less politically constrained as he or she would on highly salient issues. Thus, on these issues, the Bayesian approach may be more effective at understanding political behavior on these non-salient issues since the quadratic distribution model reflects the greater leeway that legislators have on these issues.
