Homework 5

Kevin Ayala and Aaron Barel, PSTAT 115, Fall 2018

Due on November 18, 2018 at 11:59 pm

Note: If you are working with a partner, please submit only one homework per group with both names and whether you are taking the course for graduate credit or not. Submit your Rmarkdown (.Rmd) and the compiled pdf on Gauchospace.

1. **Frequentist Coverage of The Bayesian Posterior Interval**. In quiz 1 we explored the importance and difficulty of well-calibrated prior distributions by examining the calibration of subjective intervals. Suppose that $y_1, .., y_n$ is an IID sample from a $Normal(\mu, 1)$. We wish to estimate μ.

    (a) For Bayesian inference, we will assume the prior distribution $\mu \sim Normal(0, \frac{1}{\kappa_0})$ for all parts below. State the posterior distribution of μ given $y_1, .., y_n$, and the 95% quantile-based posterior credible interval for μ.

From lecture, we learned that a normal distribution with a normal prior leads to a Posterior which is a normal model with parameters $N \sim (\mu_n, \tau^2)$ where $\mu_n = \frac{\frac{1}{\tau^2}\mu_0 + \frac{n}{\sigma^2}y}{\frac{1}{\tau^2} + \frac{n}{\sigma^2}}$
and $\tau^2 = \frac{1}{\frac{1}{\tau^2} + \frac{n}{\sigma^2}}$

since $y_1, .., y_n \sim N(\mu, 1)$, our prior distribution has the following parameters: $N(0, \frac{1}{\kappa_0})$ then our posterior has these specific parameters $\mu_n = \frac{\frac{\kappa_0}{1}\mu_0 + \frac{n}{1}y}{\frac{\kappa_0}{1} + \frac{n}{1}}$ and $\tau^2 = \frac{1}{\frac{\kappa_0}{1} + \frac{n}{1}}$

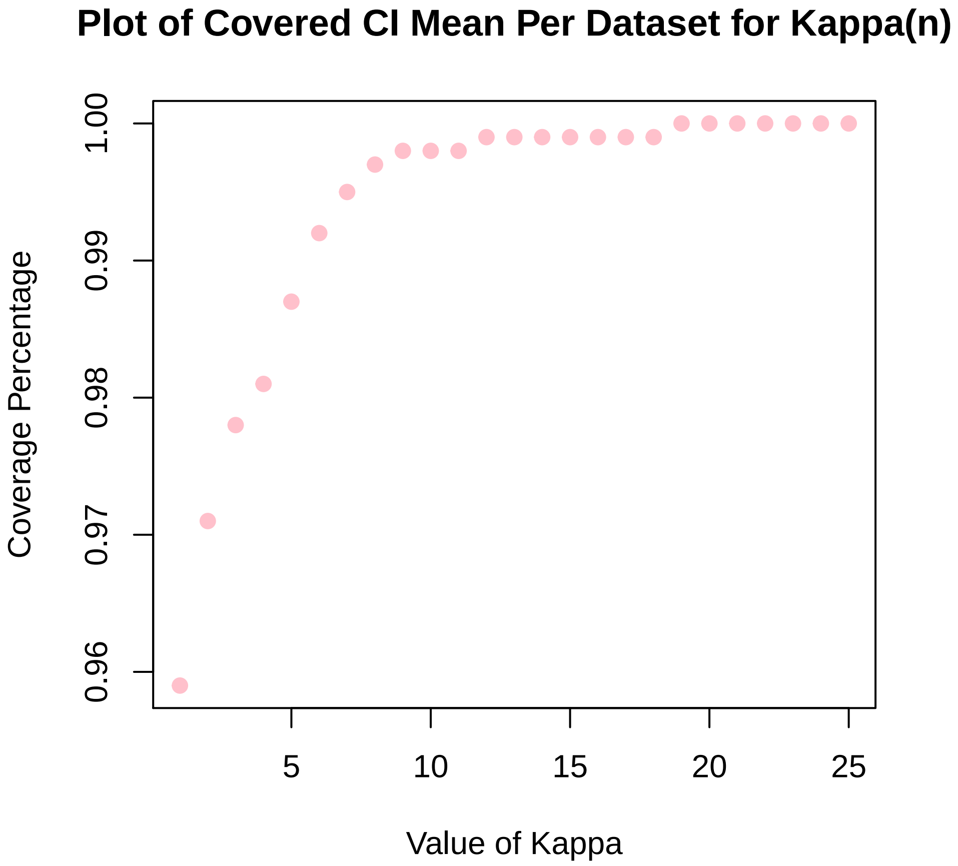
Our 95 percent credible interval would thus be $(\mu_n - 1.96\sqrt{\frac{1}{\kappa_0+n}}, \mu_n + 1.96\sqrt{\frac{1}{\kappa_0+n}})$

```
#. Now we will evaluate the frequentist coverage of the credible interval on simulated data.  Generate 
```

```r
 #The data
    normal.samples<- data.frame(rnorm(10, 0, 1))
  for (k in 1:1000) {
  normal.samples[k] <- data.frame(rnorm(10, 0, 1))
   }




#the averages for each dataset
averages = matrix(NA, nrow=1000, ncol=1)
colnames(averages) <- c("Mean Under Dataset")
rownames(averages) <- c (1:1000)

  for (i in 1:1000){
  averages[i]<-(mean(normal.samples[,i]))
  }
#averages

coverages <- matrix(NA, ncol=1, nrow = 25)
colnames(coverages) <- "coverage"
head(coverages)
```

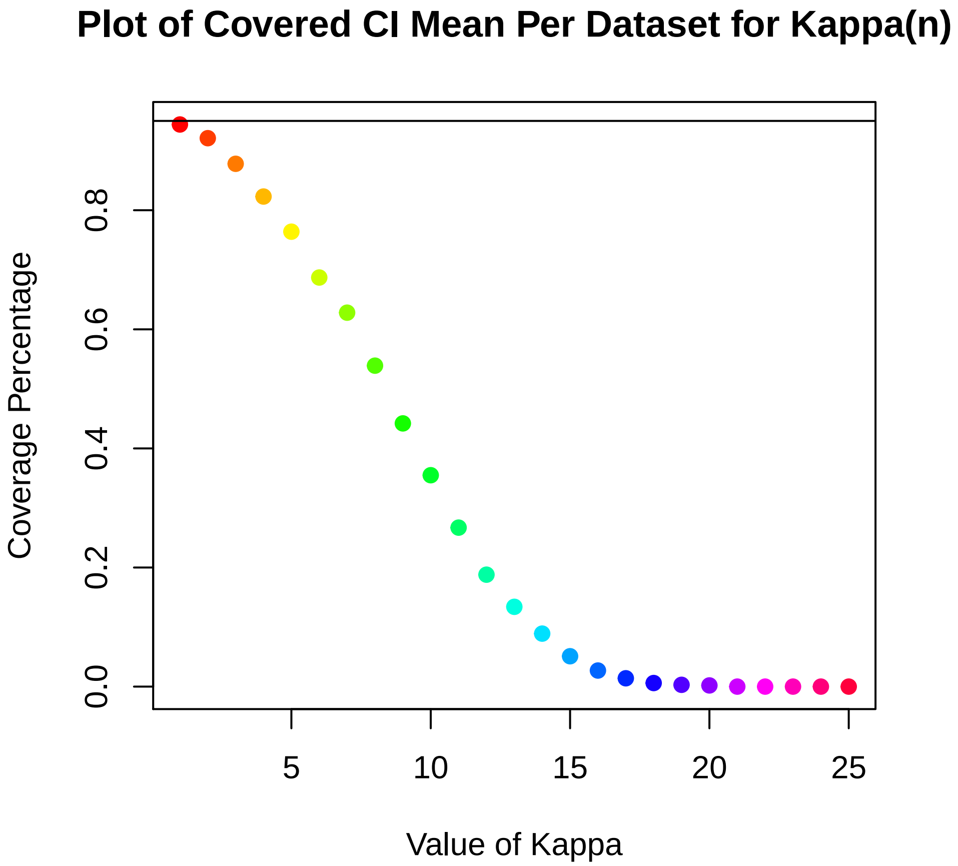
```
##      coverage
## [1,]       NA
## [2,]       NA
## [3,]       NA
## [4,]       NA
## [5,]       NA
## [6,]       NA
```

```
zero.in.intervaltruth<-rep(0,25)
lowerbound <-data.frame(qnorm(.025, mean=averages*10/(5+10), sd=sqrt(1/(5+10))))
#used to get a dataframe for loop
upperbound <-data.frame(qnorm(.975, mean=averages*10/(5+10), sd=sqrt(1/(5+10))))
#used to get a dataframe for loop
set.seed(1)
for (k in 1:25){

lowerbound[k] <-data.frame(qnorm(.025, mean=averages*10/(k+10), sd=sqrt(1/(k+10))))
upperbound[k] <-data.frame(qnorm(.975, mean=averages*10/(k+10), sd=sqrt(1/(k+10))))
zero.in.intervaltruth[k] <-  sum(lowerbound[k]<0 & 0<upperbound[k])
kappas <- zero.in.intervaltruth[1:25]
coverages[k] <- kappas[k]/1000

}

plot(coverages, col="pink", xlab="Value of Kappa",
     ylab="Coverage Percentage",
     main="Plot of Covered CI Mean Per Dataset for Kappa(n)", pch=19)
abline(h=.95)
```



1. Repeat the previous part but now generate data assuming the true $\mu = 1$.

```
   #The data

normal.samples2<-data.frame(rnorm(10, 1, 1))

     for (k in 1:1000) {
  normal.samples2[k] <- data.frame(rnorm(10, 1, 1))
   }


#the averages for each dataset
averages2 = matrix(NA, nrow=1000, ncol=1)
colnames(averages2) <- c("Mean Under Dataset")
rownames(averages2) <- c (1:1000)


  for (i in 1:1000){
  averages2[i]<-(mean(normal.samples2[,i]))
  }
#averages2

coverages2 <- matrix(NA, ncol=1, nrow = 25)
colnames(coverages2) <- "coverage"
#head(coverages2)




lowerbound2 <-data.frame(qnorm(.025, mean=averages*10/(5+10), sd=sqrt(1/(5+10))))
#used to get data frame for loop
upperbound2 <-data.frame(qnorm(.975, mean=averages*10/(5+10), sd=sqrt(1/(5+10))))
#used to get data frame for loop
one.in.intervaltruth<- rep(0,25)

for (k in 1:25){

lowerbound2[k] <-data.frame(qnorm(.025, mean=averages2*10/(k+10), sd=sqrt(1/(k+10))))
upperbound2[k] <-data.frame(qnorm(.975, mean=averages2*10/(k+10), sd=sqrt(1/(k+10))))
one.in.intervaltruth[k] <-  sum(lowerbound2[k]<1 & 1<upperbound2[k])
kappas2 <- one.in.intervaltruth[1:25]
coverages2[k] <- one.in.intervaltruth[k]/1000

}
#head(lowerbound2)
#head(coverages2)
#head(one.in.intervaltruth)


plot(coverages2, col=rainbow(25), xlab="Value of Kappa",
     ylab="Coverage Percentage",
     main="Plot of Covered CI Mean Per Dataset for Kappa(n)", pch=19)
abline(h=.95, col="black")
```



1. Explain the differences between the two plots. For what values of κ_0 do you see closer to nominal coverage (i.e. 95%)? For what values does your posterior interval tend to overcover (the interval covers the true value more than 95% of the time)? Undercover (the interval covers the true value less than 95% of the time)? Why does this make sense?

We see that when we set$\mu = 0$, that our plot reveals that the coverage increases as κ gets larger and we find 100% complete coverage for when $\kappa = 17, ..., 25$. This is because our confidence interval gets larger, looking at the mean in our confidence interval estimate, we note that

$$\mu_n = \frac{\frac{10}{1}y}{\frac{\kappa_0}{1}+\frac{10}{1}}$$

we notice that the mean gets smaller for for high kappas and gets closer to zero, making it likely for 0 to be within the interval the closer the mean estimate is to 0. Also, we overcover above 95% and the closest we get to 95% perfect coverage is when kappa is equivalent to 1.

For the second, plot we notice that for when true $\mu_0 = 1$, then our mean estimate becomes $\mu_n = \frac{\frac{\kappa_0}{1}*1+\frac{10}{1}y}{\frac{\kappa_0}{1}+\frac{10}{1}}$, we observe a kappa in the numerator and this indicates that for every increasing value of kappa, μ_n increases and gets "farther away" from the value of 1. Since at when kappas increase, our τ^2 estimate gets smaller thus the credible interval gets tighter and centered around an integer larger than one. This explains why at large κ, we observe that μ_n is so far away from one that it leads to zero coverage at a larger kappa.

We mostly tend to undercover as kappa gets larger and get close to true 95% coverage when kappa is 1.

1. **Modeling Election Outcomes**. On November 4, 2014 residents of Kansas voted to elect a member of the United States Senate to represent the state. After the primaries, there were four major contenders in the race: 1) Republican incumbent Pat Roberts, 2) Democrat Chad Taylor, 3) Independent Greg Orman, and 4) Liberatarian Randall Batson.

   For this problem we will reference polling data that can be found here:

http://en.wikipedia.org/wiki/United_States_Senate_election_in_Kansas,_2014#Polling_3

In mid-August 2014 a SurveyUSA poll of 560 people found the following vote preferences:

Pat Roberts	Chad Taylor	Greg Orman	Randall Batson	Undecided
37%	32%	20%	4%	7%

Ignoring the "undecided votes", the maximum likelihood estimate for the true vote shares of each candidate assuming, assuming a multinomial distribution over the 4 candidates, is simply the fraction of people.

(a) Assume that you first interview the 7% of undecided voters. They claim they are equally likely to vote for any of the four candidates. Before reviewing the other survey data, you decide to use this information to construct a prior distribution for the true vote shares of the four candidates. What is the prior distribution and what are it's parameters (think pseudocounts)? Given the survey data above and the prior, specify the posterior distribution for the vote shares of the four candidates and the parameters of this distribution.

We want our prior distribution to be a Dirichlet distribution since it is a conjugate to the multinomial distribution on the condidates.

A Dirichlet distribution has the form $\theta_1, ..., \theta_n \sim Dirichlet(\alpha_1, ..., \alpha_n)$ Considering the information on the candidates and the fact that the voter information follows a multinomial distribution of the form $y_1, ..., y_n \sim multinomial(\theta_1, ..., \theta_n$ the conjugate is useful here, the dirichlet follows the form of $\theta_1, ..., \theta_n \sim Dirichlet(\alpha_1, ..., \alpha_4)$. we chose our prior parameters to be $\theta_1, ..., \theta_4 \sim Dirichlet(9.8, 9.8, 9.8, 9.8)$ since each candidate can possibly inherit equal amount of the undecided voters vote, then we can parameterize the estimated votes for each voter by multiplying $.07 * 560/4$ where .07 is the percentage of undecided voter spread out evenly. In lecture we learned that the posterior distribution is also a Dirishlet distribution. We can parameterize this posterior as $Dirishlet(9.8 + 207, 9.8 + 179, 9.8 + 112, 9.8 + 22)$ where each simplex represents the total amount of votes each candidate should recieve based on the polling data and redistributed votes from undecided voters.

```r
#install.packages("MCMCpack")
library(MCMCpack)
```

```
## Loading required package: coda

## Loading required package: MASS

##
## Attaching package: 'MASS'

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

## ##
## ## Markov Chain Monte Carlo Package (MCMCpack)

## ## Copyright (C) 2003-2018 Andrew D. Martin, Kevin M. Quinn, and Jong Hee Park

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

## ## (Grants SES-0350646 and SES-0350613)
## ##
```

```r
#?rdirichlet
#rdirichlet(1, c(1, 1, 1))
#rdirichlet(1, c(1, 1, 100))
#rdirichlet(1, c(100, 100, 100))
```

```
#. On September 3, 2014 Democratic nominee Chad Taylor withdrew from the
```

race. Assume that amongst those who said they would vote for Taylor in the August survey, 70% of them changed their vote to Orman, 20% to the Libertarian, Baston, and the remaining 10% for Roberts. The above information should be used construct a new prior distribution for the 3-candidate race again assuming that the undecided voters from the August poll will now vote equally among the remaining three candidates. Calculate the new posterior distribution over the vote shares for the 3 remaining candidates. Use Monte Carlo to find the posterior probability that more people in Kansas support Pat Roberts than Greg Orman.

```r
set.seed(1)
CandidatesPercentages <- matrix(NA, nrow=3, ncol=1)
rownames(CandidatesPercentages) <- c("Greg Orman", "Randall Batson", "Pat Roberts")
colnames(CandidatesPercentages) <- "Voter Percentages After Chad Taylor Drop"

CandidatesPercentages[1] <- (32*.7+20)/100
CandidatesPercentages[2] <- (32*.2+4)/100
CandidatesPercentages [3] <- (32*.1+37)/100

CandidatesPercentages
```

```
##                  Voter Percentages After Chad Taylor Drop
## Greg Orman                                          0.424
## Randall Batson                                      0.104
## Pat Roberts                                         0.402
```

```r
#this matrix shows the percentage in shared votes per candidate after Taylor Drops




ExpectedVotes <- matrix(NA, nrow=3, ncol=1)
rownames(ExpectedVotes) <- c("Greg Orman", "Randall Batson", "Pat Roberts")
colnames(ExpectedVotes) <- "Expected Number of Votes For Each Candidate"

ExpectedVotes[1] <- round((32*.7+20)/100 * 560)
ExpectedVotes[2] <- round((32*.2+4)/100 *560)
ExpectedVotes[3] <- round((32*.1+37)/100 * 560)

ExpectedVotes
```

```
##                  Expected Number of Votes For Each Candidate
## Greg Orman                                             237
## Randall Batson                                          58
## Pat Roberts                                            225
```

```r
#expected votes number according to poll of 560 voters.




#since undecided voters are still equally likely to vote equally between the three,
#we can reparameterize in the form of .07*560/3 which equals 13.0666 since its possible all undecided v

#we know from class that the posterior of a multinomial and dirshlet is also a dirishlet
set.seed(1)
dirish.samples <- rdirichlet(1000, c(13+ExpectedVotes[1], 13+ExpectedVotes[2], 13+ExpectedVotes[3]))
colnames(dirish.samples) <- c("Greg Orman","Randal Batson","Pat Roberts")
head(dirish.samples) #col 1 is Greg Orman, 2 is Randal Batson, 3 is Pat Roberts
```

```
##      Greg Orman Randal Batson Pat Roberts
## [1,]  0.4137626     0.1417299   0.4445075
## [2,]  0.4578587     0.1039872   0.4381541
## [3,]  0.4587945     0.1324197   0.4087858
## [4,]  0.4427898     0.1221307   0.4350794
## [5,]  0.4424322     0.1124081   0.4451597
## [6,]  0.4410968     0.1335835   0.4253198
```

```
RobertsoverGregprobability <- sum(dirish.samples[,3] > dirish.samples[,1])/1000
#occurences when Pat Roberts vote share is greater than Greg Orman...
#...divided by the 1000(the election similations)
RobertsoverGregprobability #Probability People in Kansas support Pat Roberts over Greg Orman
```

```
## [1] 0.295
```

```
#with this result, we chose to split the undecided voters evenly between candidates...
#...in part C since they are wildcards
```

```
#. From October 22-26, 2014 SurveyUSA released a poll of 623 found the following preferences:\linebreak
  \begin{table}[h!]
  \centering
  \begin{tabular}{cccccc}
    \hline
   & Pat Roberts & Chad Taylor & Greg Orman & Randall Batson & Undecided \\
    \hline
   & 42\% & -- & 44\% & 4\% & 10\% \\
     \hline
  \end{tabular}
  \end{table}

    Use the posterior from the previous part as the prior for this new survey.  Compute a new posterior

    i.  Greg Orman's team believes that if they can get at least 20000 votes they will win the election
```

```
voter.turnout.prob <- rbeta(10000, 40, 60)
head(voter.turnout.prob) # the amount of voters who vote from population per similuated election
```

```
## [1] 0.4451931 0.4301082 0.4322088 0.4087107 0.3777868 0.4041828
```

```
Candidates <- matrix(NA, nrow=4, ncol = 1)
rownames(Candidates) <- c("Pat Roberts","Greg Orman","Randall Batson","Undecided")
colnames(Candidates) <- "Votes Expected Per 623 Votes"
Candidates[1] <- round(.42 * 623)
Candidates[2] <- round(.44 * 623)
Candidates[3] <- round(.04 *623)
Candidates[4] <- round(.10*623)
Candidates #for orginization purposes
```

```
##                 Votes Expected Per 623 Votes
## Pat Roberts                              262
## Greg Orman                               274
## Randall Batson                            25
## Undecided                                 62
```

```
#we split undecided voters as equal for each candidate such tht 62/3 is approx.
#21 votes for each candidate rounded.
```

```
voter.turnout<- 100000*voter.turnout.prob
head(voter.turnout) # each element is a turnout of voters for that similuated election
```

```
## [1] 44519.31 43010.82 43220.88 40871.07 37778.68 40418.28
```

```
sim.share.of.votes <- rdirichlet(10000, c(521, 545, 117))
colnames(sim.share.of.votes)<- c("Pat Roberts","Greg Orman","Randall Patson")
head(sim.share.of.votes) #amount of vote percentage share for candidates, each row is a similated elect
```

```
##      Pat Roberts Greg Orman Randall Patson
## [1,]   0.4384610  0.4605816     0.10095737
## [2,]   0.4138200  0.4561606     0.13001940
## [3,]   0.4492123  0.4559098     0.09487792
## [4,]   0.4488207  0.4571195     0.09405973
## [5,]   0.4268927  0.4739813     0.09912603
## [6,]   0.4591981  0.4498180     0.09098382
```

```
#how I got the parameters above for rdirichlet?
#We added both data from part b and part c together, the following matrix shows how we arrived at these
Our.Parameters <- matrix(NA, nrow = 4, ncol=1)
row.names(Our.Parameters) <- c("Pat Roberts", "Greg Orman", "Randall Batson", "Total")
Our.Parameters[1] <- (13+225)+(21+262)
Our.Parameters[2] <- (13+237)+(21+274)
Our.Parameters[3] <- (13+58)+(21+25)
Our.Parameters[4]<- sum(Our.Parameters[1]+Our.Parameters[2]+Our.Parameters[3])
Our.Parameters
```

```
##                 [,1]
## Pat Roberts      521
## Greg Orman       545
## Randall Batson   117
## Total           1183
```

```
#We added both datasets together for the new updated prior, 560+623 total votes
#from the first parenthesis, we equally spread undecided votes to the candidates which led to 13 (round
#plus the redistributed votes after Chad Taylor dropped from the race, then we added the new informatio
#added undecided votes equally amongst candidates which led to plus 21 votes (rounded) for each candida
#then calculated percentages in new data in part c into counts which were rounded to integers...
#... calculations of these counts in matrix "Candidates"

Total.Votes.PerCandidate.Perelection <- matrix(NA, nrow=10000, ncol=3)
colnames(Total.Votes.PerCandidate.Perelection) <- c("Pat Roberts", "Greg Orman", "Randall Batson")

Total.Votes.PerCandidate.Perelection[,1] <- sim.share.of.votes[,1]*voter.turnout
Total.Votes.PerCandidate.Perelection[,2] <- sim.share.of.votes[,2]*voter.turnout
Total.Votes.PerCandidate.Perelection[,3] <- sim.share.of.votes[,3]*voter.turnout

head(Total.Votes.PerCandidate.Perelection)
```

```
##      Pat Roberts Greg Orman Randall Batson
## [1,]    19519.98   20504.78       4494.553
## [2,]    17798.74   19619.84       5592.241
## [3,]    19415.35   19704.82       4100.707
## [4,]    18343.78   18682.97       3844.322
## [5,]    16127.44   17906.39       3744.851
## [6,]    18560.00   18180.87       3677.410
```

```
sum(Total.Votes.PerCandidate.Perelection[,2] > 20000 &
      (Total.Votes.PerCandidate.Perelection[,2] > (Total.Votes.PerCandidate.Perelection[,1] )))/10000
```

```
## [1] 0.2019
```

```
#probability of Orman getting at least 20,000 votes and winnnig.
#assuming person who has most votes out of all three wins election.
#Winner between Orman and Roberts, Batson irrelev.
```

```
    ii.  Both leading candidates fear that the third party vote is taking away potential supporters.  Wh
```

```
set.seed(1)
voteshare<-data.frame(Total.Votes.PerCandidate.Perelection)

votesharediff<- data.frame(abs(voteshare[,1]-voteshare[,2]))
diff.vs.randall<- cbind(votesharediff, Total.Votes.PerCandidate.Perelection[,3])

sum(diff.vs.randall[,1] < diff.vs.randall[,2])/10000
```

```
## [1] 0.9968
```

We can conclude from this probability that both candidates are correct in fearing the third party. The race between the leading candidates seems close enough that the voters casting votes for Randall Batson will decide who wins, meaning the candidate who does not win over enough Randall Batson voters will likley lose.

```
#. Discuss the assumptions that were made to generate your
```

predictions. If you think some assumptions were poor, how might you change the model to improve upon them? This is an open-ended question with no right or wrong answers and will be graded on thoughtfullness and effort only.

Considering that it is an election, I think it is safe to assume that we made an assumptin that the candidates did not suffer from any scandal on their personal life or political life as is usually the case in todays media world, usually when stuff like that happens a candidate's public profile goes down and they go down in the polls as time goes on. Essentailly, we are making assumptions based on a strict point in time before the election, something could have happend day after the polls were taken and thus the votes do not reflect that.

Another assumption that was made was that since the target population is Kansas, the polling information perhaps was limited only to those voters in huge cities where it is easy to poll data and thus alieniating voters who live in the country side or in rural areas. In fact, this is one of the many reasons that the last presidential election was so poorly forecasted, because polls did pull in enough data from rural areas or from areas where people are away from big cities.

We are also assuming that the polling data was adequatley randomized from polling voters. One big assumption we made was that the voters in the polls were of perhaps of different races, genders, and the big one, age. We do not know if there is bias or a large chunk of polling where the average age was 60 or something, also race may play a role so we assumed it doesnt because it is in poor ethnic judgment to assume that it does and is subjective.