DS1 Exam

Rajendran Seetharaman

December 10, 2017

Problem 1: 2016 Election Results Data (25pt)

Use the following datasets to analyze the US 2016 election results (available on canvas in files/data) by county. 1. US_County_Level_Presidential_Results_08-16.csv.bz2 2. county_data.csv.bz2 3. there is also an explanation file for the county data county_data_variables.pdf. Note: the datasets can be merged by FIPS (Federal Information Processing Standards) codes. There are 3-digit county FIPS codes and 2-digit state FIPS codeds, some data use 5 digit FIPS instead: 2 digits for the state followed by 3 digits for the county.

1. Tidy the data. Merge these datasets, retain only more interesting variables, compute additional variables you find interesting, and consider giving these more descriptive names. Explain briefly what did you do.

Ans.

I took the following steps to tidy the data and computed the following additional useful variables-

I only selected the data/variables pertaining to the year 2016 as we are looking at the election results for that particular and how they might have been influenced by factors like the population in a particular county during that year. I selected fips code, county, and votes information from results dataset. I converted the democrat,GOP, other columns to rows by transforming the vote count columns for each party to two columns- party name, vote count for party. I also extracted state and country fips codes from data to facilitate joining with the county data. I retained the region, division, state, county, population, and migration variables from county dataset. I finally joined the 2 datasets on county and state fips codes and computed percent votes for each party in each county as this would be extremely useful for the subsequent questions. I also gave the variables more descriptive names.

In the results dataset, the variables I retained were the fips codes, county and votes information as these are all important as well as interesting variables. The vote counts are the main outcome of the election. In the county dataset, I retained the region, state, county information. I also retained the variables pertaining to population numbers and changes, and migration numbers, as these seem to be interesting and feel that they might have some relationship with election outcome i.e vote counts.

```
library(readr)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.4.3
```

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

```
## v ggplot2 2.2.1     v purrr   0.2.4
## v tibble  1.3.4     v dplyr   0.7.4
## v tidyr   0.7.2     v stringr 1.2.0
## v ggplot2 2.2.1     v forcats 0.2.0
```

```
## Warning: package 'tidyr' was built under R version 3.4.3
```

```
## Warning: package 'purrr' was built under R version 3.4.3
```

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

```
#read the datasets
county_pres_results <-
  read_csv("C:/Users/admin/Downloads/US_County_Level_Presidential_Results_08-16.csv.bz2")
```

```
## Parsed with column specification:
## cols(
##   fips_code = col_character(),
##   county = col_character(),
##   total_2008 = col_integer(),
##   dem_2008 = col_integer(),
##   gop_2008 = col_integer(),
##   oth_2008 = col_integer(),
##   total_2012 = col_integer(),
##   dem_2012 = col_integer(),
##   gop_2012 = col_integer(),
##   oth_2012 = col_integer(),
##   total_2016 = col_integer(),
##   dem_2016 = col_integer(),
##   gop_2016 = col_integer(),
##   oth_2016 = col_integer()
## )
```

```
county_data <- read_csv("C:/Users/admin/Downloads/county_data.csv.bz2")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   STNAME = col_character(),
##   CTYNAME = col_character(),
##   RBIRTH2011 = col_double(),
##   RBIRTH2012 = col_double(),
##   RBIRTH2013 = col_double(),
##   RBIRTH2014 = col_double(),
##   RBIRTH2015 = col_double(),
##   RBIRTH2016 = col_double(),
##   RDEATH2011 = col_double(),
##   RDEATH2012 = col_double(),
##   RDEATH2013 = col_double(),
##   RDEATH2014 = col_double(),
##   RDEATH2015 = col_double(),
##   RDEATH2016 = col_double(),
##   RNATURALINC2011 = col_double(),
##   RNATURALINC2012 = col_double(),
##   RNATURALINC2013 = col_double(),
##   RNATURALINC2014 = col_double(),
##   RNATURALINC2015 = col_double(),
##   RNATURALINC2016 = col_double()
##   # ... with 18 more columns
## )
```

```
## See spec(...) for full column specifications.
```

```
#select fips code, county, and votes information from results dataset
#convert cols to rows by transforming cols for each party to cols- party name, votes
#extract state and country fips codes from data
```

```
county_pres_results <- county_pres_results %>%
  select(fips_code,county,democrats=dem_2016,republicans=gop_2016,
  others=oth_2016,total_votes=total_2016) %>%
  gather(party,votes,3:5) %>% arrange(fips_code) %>%
  mutate(state_fips=as.integer(substr(fips_code,1,2)),
  county_fips=as.integer(substr(fips_code,3,5)))

#select region, division, state, county, population, and migration variables from county dataset
county_data <- county_data %>% select(region=REGION,division=DIVISION,state_fips=STATE,
 county_fips=COUNTY,state_name=STNAME,county_name=CTYNAME,
 resident_population_estimate=POPESTIMATE2016,
 resident_population_change=NPOPCHG_2016,
 international_migration=INTERNATIONALMIG2016,
 domestic_migration=DOMESTICMIG2016,net_migration=NETMIG2016,
 international_migration_rate=RINTERNATIONALMIG2016,
 domestic_migration_rate=RDOMESTICMIG2016,net_migration_rate=RNETMIG2016)

#join the 2 datasets on county and state fips codes
#compute percent votes for each party in each county
voting_data <- merge(x=county_pres_results,y=county_data,
  by.x=c("state_fips","county_fips"),by.y=c("state_fips","county_fips")) %>%
  arrange(fips_code,party) %>% mutate(vote_percent=(votes*100)/total_votes)
#clean county variable by removing extraneous information
voting_data <- voting_data %>% mutate(county_name=
  substr(county,1,regexpr(" ",county)-1))
```

2. describe the data and the more interesting variables. Which variables' relationship to the election outcomes you might want to analyze?

Ans.

The first dataset (county data) contains the following data about each county in the US for different years - population estimate, number of births/deaths, change in population, natural increase in population; net migrations (both domestic and international), group quarters population estimates, birth/death rates, natural increase rates, and net migration rates.

The second dataset (results data) contains the results from from the last 3 presidential elections. It contains information pertaining to the vote counts secured by parties i.e democratic, republican, and other in each of the counties.

I feel that the variables like county, region, state, vote counts for the year 2016, the county population,population migration numbers and rates, for the year 2016 would be helpful to analyze the relationship with the outcomes of the elections for 2016.

```
str(voting_data)
```

```
## 'data.frame':    9333 obs. of  20 variables:
##  $ state_fips                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ county_fips               : int  1 1 1 3 3 3 5 5 5 7 ...
##  $ fips_code                 : chr  "01001" "01001" "01001" "01003" ...
##  $ county                    : chr  "Autauga County" "Autauga County" "Autauga County" "Baldwin Co
##  $ total_votes               : int  24661 24661 24661 94090 94090 94090 10390 10390 10390 8748 ...
##  $ party                     : chr  "democrats" "others" "republicans" "democrats" ...
##  $ votes                     : int  5908 643 18110 18409 2901 72780 4848 111 5431 1874 ...
##  $ region                    : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ division                  : int  6 6 6 6 6 6 6 6 6 6 ...
```

```
##  $ state_name                  : chr  "Alabama" "Alabama" "Alabama" "Alabama" ...
##  $ county_name                 : chr  "Autauga" "Autauga" "Autauga" "Baldwin" ...
##  $ resident_population_estimate: int  55416 55416 55416 208563 208563 208563 25965 25965 25965 22643
##  $ resident_population_change  : int  381 381 381 4873 4873 4873 -305 -305 -305 82 ...
##  $ international_migration     : int  7 7 7 243 243 243 -5 -5 -5 18 ...
##  $ domestic_migration          : int  228 228 228 4046 4046 4046 -248 -248 -248 34 ...
##  $ net_migration               : int  235 235 235 4289 4289 4289 -253 -253 -253 52 ...
##  $ international_migration_rate: num  0.127 0.127 0.127 1.179 1.179 ...
##  $ domestic_migration_rate     : num  4.13 4.13 4.13 19.63 19.63 ...
##  $ net_migration_rate          : num  4.26 4.26 4.26 20.81 20.81 ...
##  $ vote_percent                : num  23.96 2.61 73.44 19.57 3.08 ...
```

```
summary(voting_data)
```

```
##    state_fips     county_fips     fips_code            county
##  Min.   : 1.00   Min.   :  1.0   Length:9333        Length:9333
##  1st Qu.:19.00   1st Qu.: 35.0   Class :character   Class :character
##  Median :29.00   Median : 78.0   Mode  :character   Mode  :character
##  Mean   :30.54   Mean   :103.2
##  3rd Qu.:46.00   3rd Qu.:133.0
##  Max.   :56.00   Max.   :840.0
##   total_votes         party               votes             region
##  Min.   :     64   Length:9333        Min.   :      3   Min.   :1.000
##  1st Qu.:   4816   Class :character   1st Qu.:    635   1st Qu.:2.000
##  Median :  10935   Mode  :character   Median :   2642   Median :3.000
##  Mean   :  40909                      Mean   :  13636   Mean   :2.656
##  3rd Qu.:  28696                      3rd Qu.:   8699   3rd Qu.:3.000
##  Max.   :2314275                      Max.   :1654626   Max.   :4.000
##     division      state_name        county_name
##  Min.   :1.000   Length:9333        Length:9333
##  1st Qu.:4.000   Class :character   Class :character
##  Median :5.000   Mode  :character   Mode  :character
##  Mean   :5.156
##  3rd Qu.:7.000
##  Max.   :9.000
##  resident_population_estimate resident_population_change
##  Min.   :     113             Min.   :-21324.0
##  1st Qu.:   11194             1st Qu.:  -105.0
##  Median :   26027             Median :     3.0
##  Mean   :  103623             Mean   :   715.7
##  3rd Qu.:   67976             3rd Qu.:   247.0
##  Max.   :10137915             Max.   : 81360.0
##  international_migration domestic_migration  net_migration
##  Min.   :  -67.0         Min.   :-75168.00   Min.   :-47810
##  1st Qu.:    2.0         1st Qu.:  -158.00   1st Qu.:  -123
##  Median :   13.0         Median :   -30.00   Median :   -12
##  Mean   :  320.5         Mean   :     1.51   Mean   :   322
##  3rd Qu.:   67.0         3rd Qu.:   109.00   3rd Qu.:   167
##  Max.   :42607.0         Max.   : 43189.00   Max.   : 53377
##  international_migration_rate domestic_migration_rate net_migration_rate
##  Min.   :-1.0433              Min.   :-79.432         Min.   :-67.4442
##  1st Qu.: 0.1449              1st Qu.: -7.291         1st Qu.: -6.2694
##  Median : 0.5722              Median : -1.796         Median : -0.7676
##  Mean   : 1.0985              Mean   : -1.606         Mean   : -0.5072
##  3rd Qu.: 1.3946              3rd Qu.:  4.419         3rd Qu.:  5.4933
```

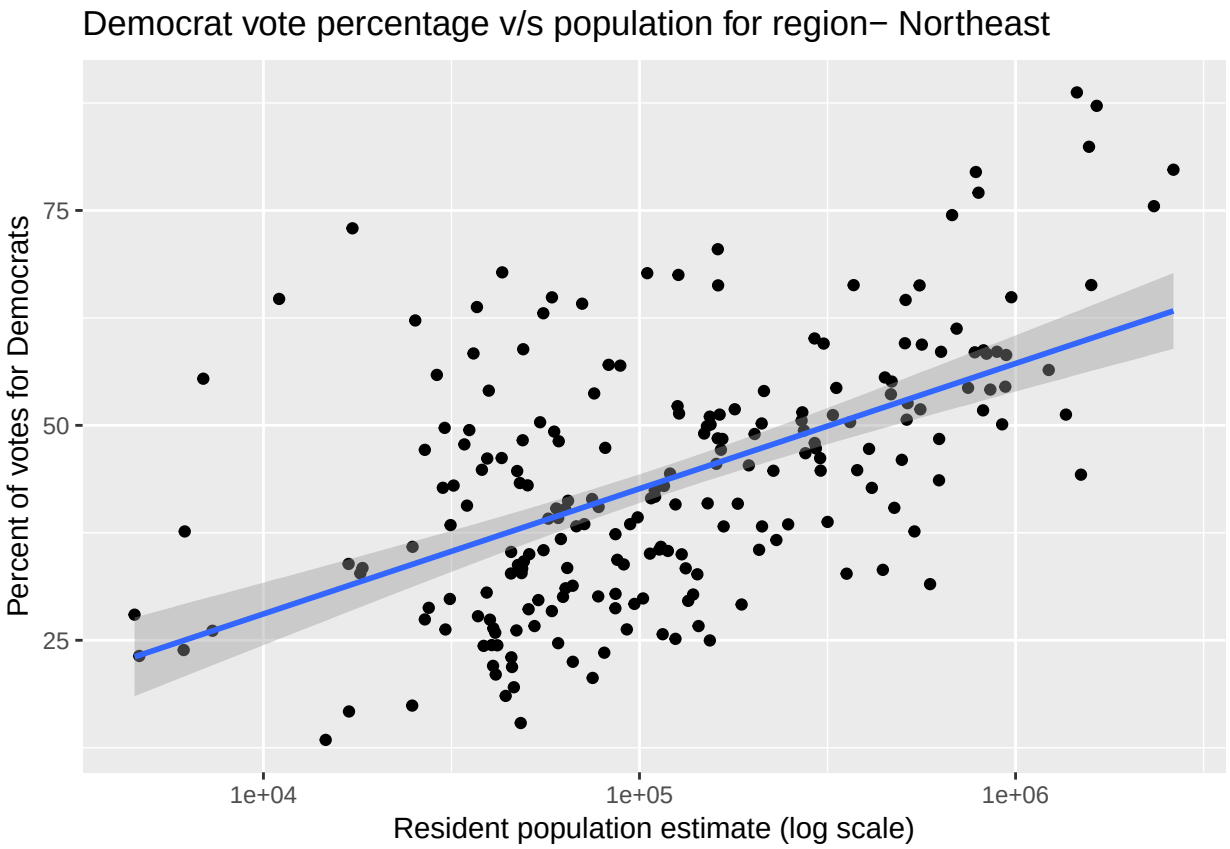
```
##  Max.   :17.1382              Max.   :153.250         Max.   :157.7962
##   vote_percent
##  Min.   : 0.3345
##  1st Qu.: 5.7843
##  Median :27.4143
##  Mean   :33.3333
##  3rd Qu.:58.6429
##  Max.   :95.2727
```

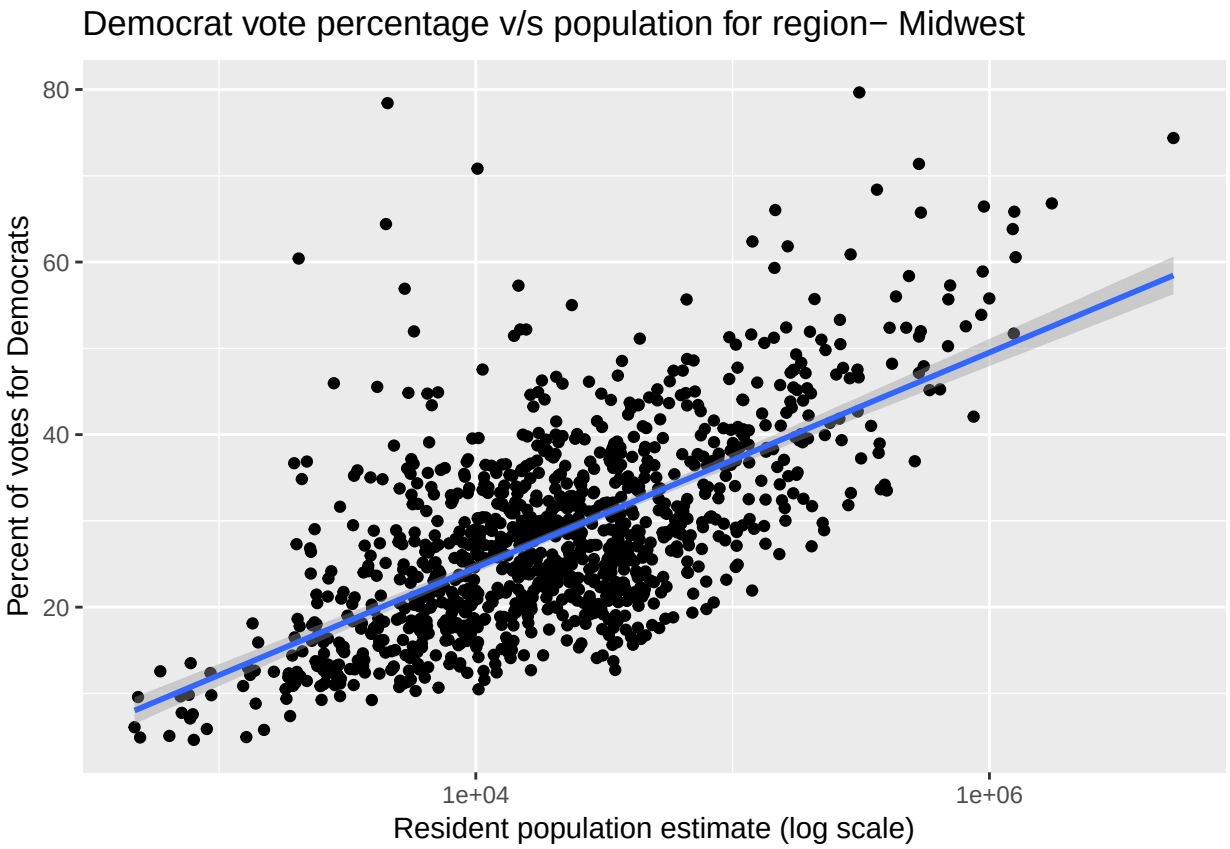
3. plot the percentage of votes for democrats versus the county population. What do you conclude? Use the appropriates labels/scales/colors to make the point clear.

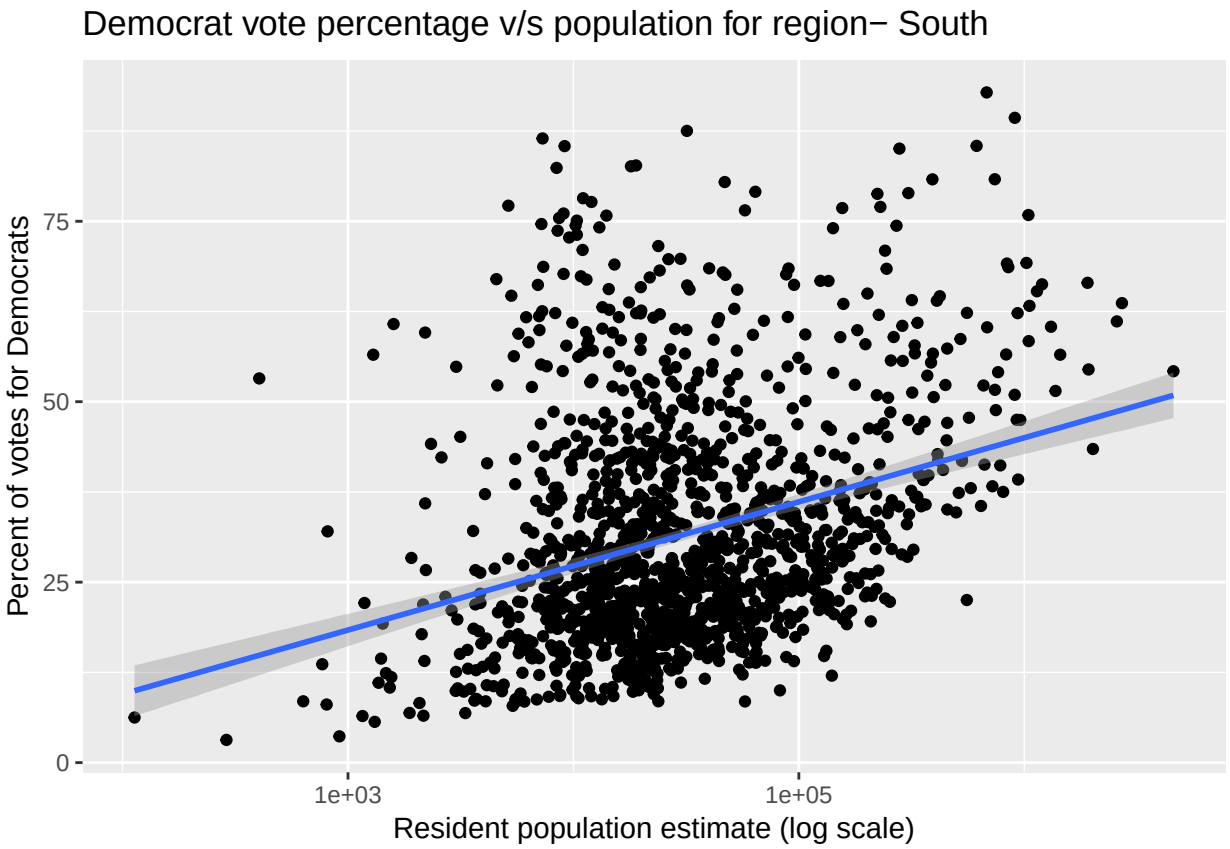
Ans. I am plotting the percentage of votes for democrats for each of the 4 geographic regions across the US to see if the results I get are consistent across all the 4 regions. I am using a logarithmic scale for the population estimate.

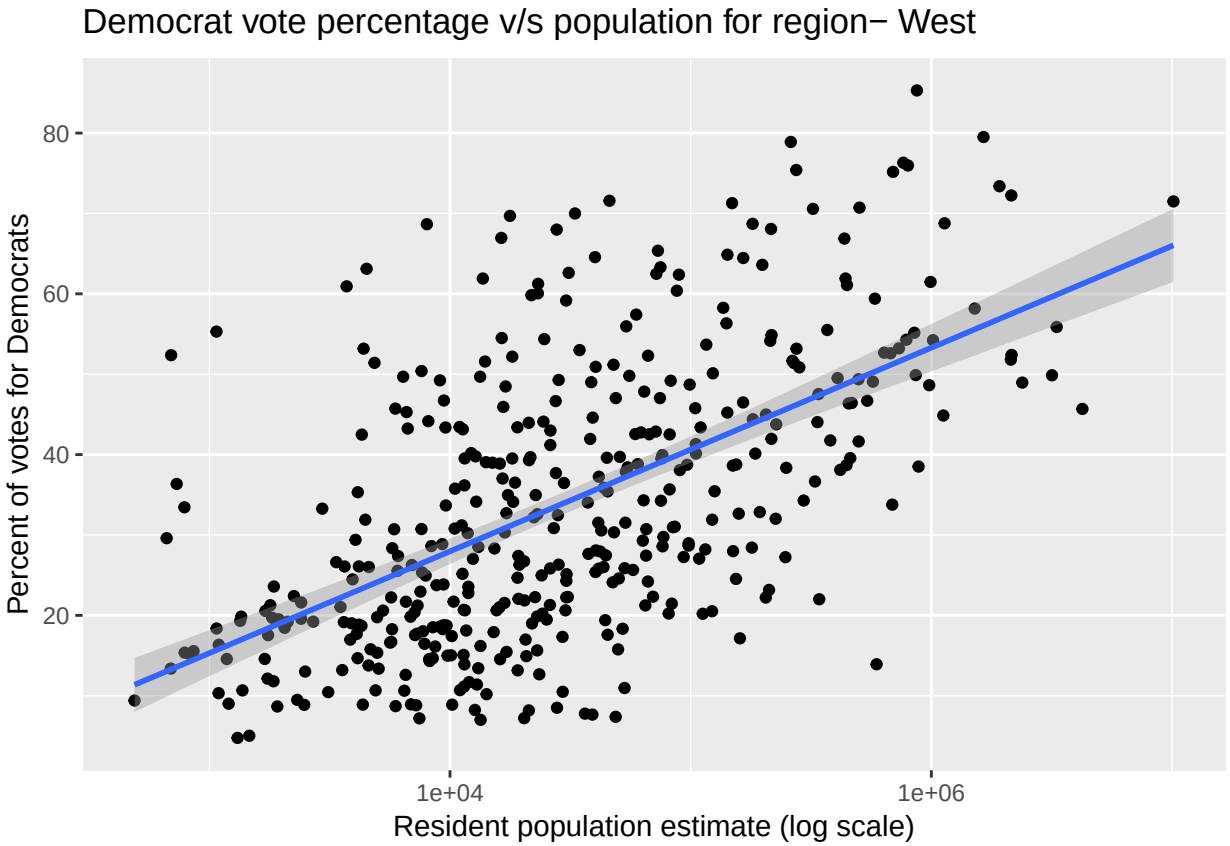
Looking at each of the 4 graphs, I see that across all the regions, an increase in the county's population is associated with an increase in percent vote for democrats in that county. I see a particularly strong trend for the midwest region. The trend might not necessarily be linear.

```
regions <- c("Northeast", "Midwest", "South", "West")
for(i in 1:4)
{
  #plot scatterplot for each of the 4 regions in the US
 print(ggplot(data=voting_data %>%
  filter(party=="democrats" & region==i),aes(x=resident_population_estimate,
  y=vote_percent))+geom_point()+scale_x_log10()+geom_smooth(method='lm')+
  labs(title=paste("Democrat vote percentage v/s population for region-",
  regions[i]),x="Resident population estimate (log scale)",
  y="Percent of votes for Democrats"))
}
```


Democrat vote percentage v/s population for region− Northeast
Percent of votes for Democrats
75
50
25
1e+04
1e+05
1e+06
Resident population estimate (log scale)


Democrat vote percentage v/s population for region− Midwest
Percent of votes for Democrats
80
60
40
20
1e+04
1e+06
Resident population estimate (log scale)


Democrat vote percentage v/s population for region− South
Percent of votes for Democrats
75
50
25
0
1e+03
1e+05
Resident population estimate (log scale)



4. Create a map of percentage of votes for democrats. Do your best to reflect the continuous percentage of votes, and the dierent population sizes across counties and keep county boundaries as well legible as you can. Mark state boundaries on the map. Explain what did you do, and what worked well, what did not work well. Hint: there are many ways to map data in R. You may consider function ggplot::map_data that includes various maps, including US administrative boundaries. However, map_data counties do not include FIPS code. You may rely on merging data by state name and county name, given you a) convert your names to lower case, and b) remove the word " county" from the end of the names. This works for most of the counties, except for Lousiana where counties are called parish.

Ans. I first retrived state and county data for the US using the map_data function. It helped my get the latitudes and longitudes for the states and counties respectively. I then converted the state and county name first letters to lowercase in the voting data to help me join with the US county data retrived earlier using geom map.

I then merged the datasets using the name of the counties and filtered the data for democrats. I used the Rcolorbrewer to plot points on the map for each each county with red representing low democrat votes, yellow representing moderate percent of democrat votes and blue representing high percent democrat votes on a continious scale. I creates state outlines using the geom_polygon function.

What worked well for me was that I was able to effectively show the percent democrat votes as points on the US Map with each state boundary clearly demarcated. What did not work well for me was representing county boundaries and the population estimates in the same plot. I decided to do away with county boundary lines as there are too many counties and the plot was getting too overcrowded with the county boundaries. I also decided not to represent the population for each county on the map (as the size of the points) as it was not appealing visually.

reference: https://uchicagoconsulting.wordpress.com/2011/04/18/how-to-draw-good-looking-maps-in-r/

```
#get state geographic data
united_states1 <- map_data("state")
```

```
## Warning: package 'maps' was built under R version 3.4.3
```

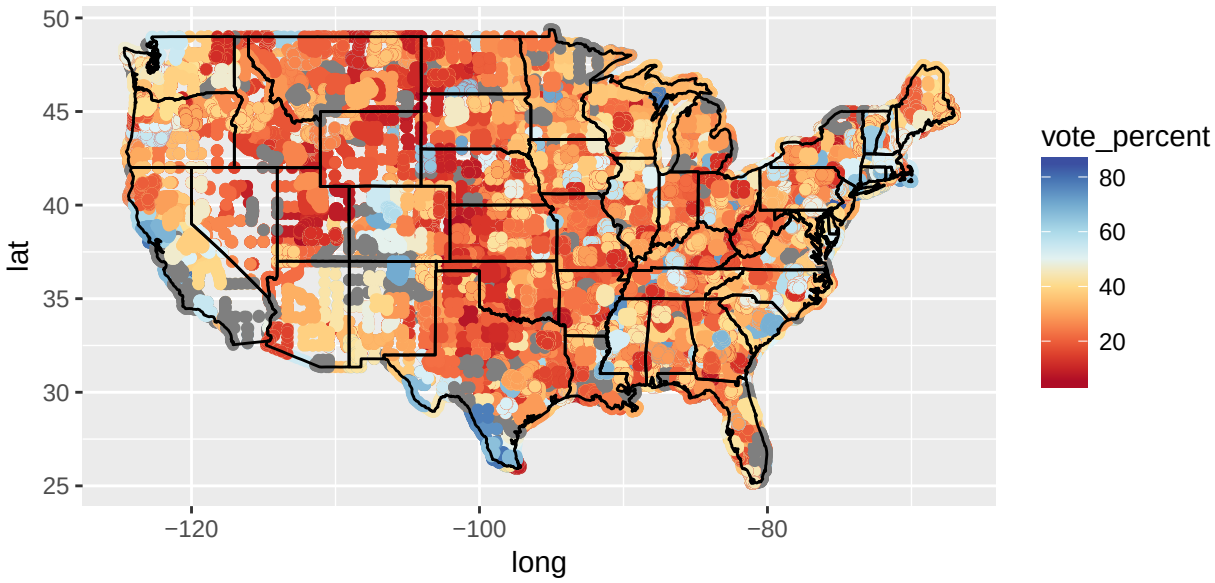
```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
#get county geographic data
united_states <- map_data("county")
#convert county names to lowercase
substr(voting_data$county_name, 1, 1) <-
  tolower(substr(voting_data$county_name, 1, 1))
substr(voting_data$state_name, 1, 1) <-
  tolower(substr(voting_data$state_name, 1, 1))

#merge datasets using county name (subregion) as key
#filter data for democrats
united_states_merged <-
  merge(x=united_states,y=voting_data %>%
  filter(party=="democrats"),by.x=c("subregion"),
  by.y=c("county_name"),all.x = TRUE)
```

```
#using the rcolorbrewer package
library(RColorBrewer)
cols <- rev(brewer.pal(10, 'RdYlBu'))
#plot the map with percent votes as points on map
ggplot(data = united_states_merged) +
  geom_point(aes(x = long, y = lat, col=vote_percent)) +
  coord_fixed(1.3)+scale_colour_gradientn(colours=rev(cols))+
  geom_polygon(data=united_states1,aes(x=long,y=lat,group=group),
  fill=NA,color="black")
```



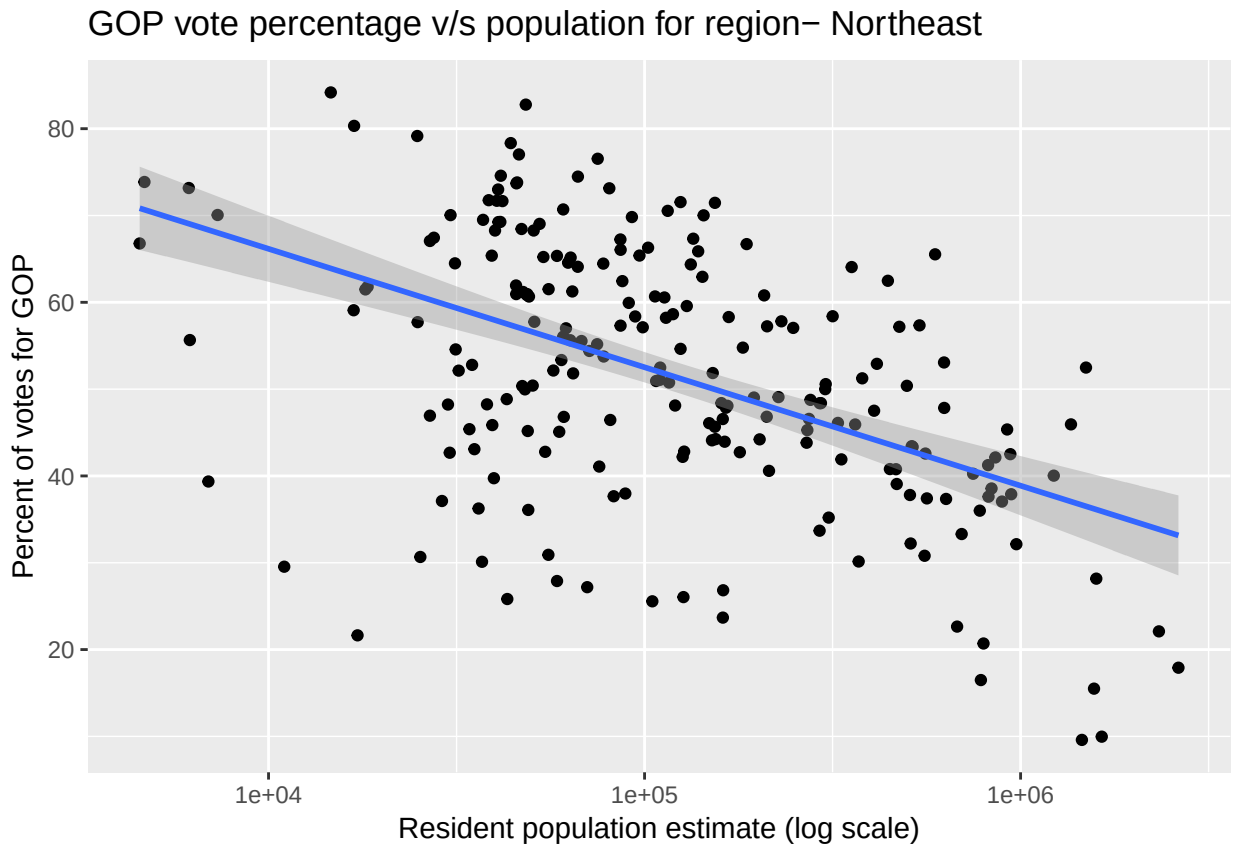
5. Create one more visualization regarding the election results on your choice. The plot should be informative and clear. Use appropriate colors/labels/explanations.

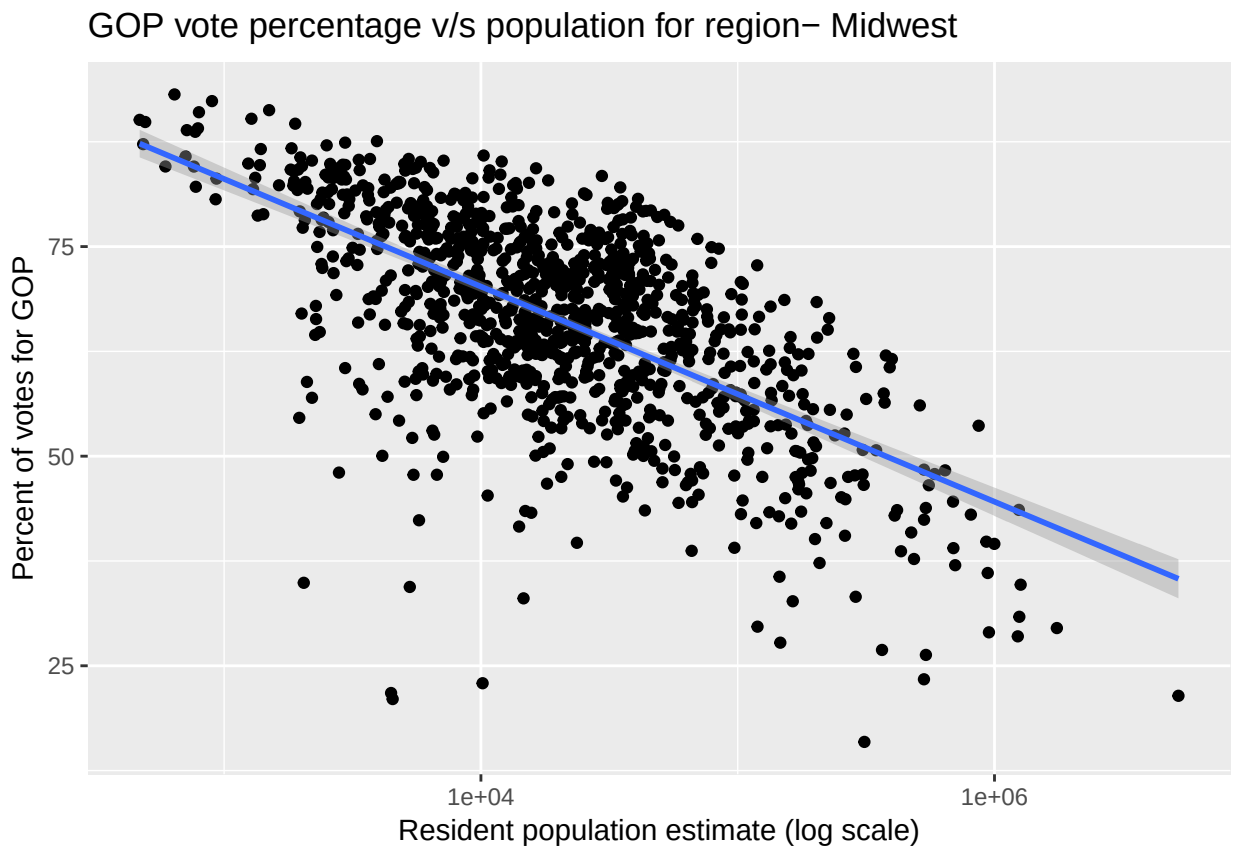
Ans. I am plotting the percentage of votes for GOP (Republicans) for each of the 4 geographic regions across the US to see if the results I get are consistent across all the 4 regions. I am using a logarithmic scale for the population estimate.

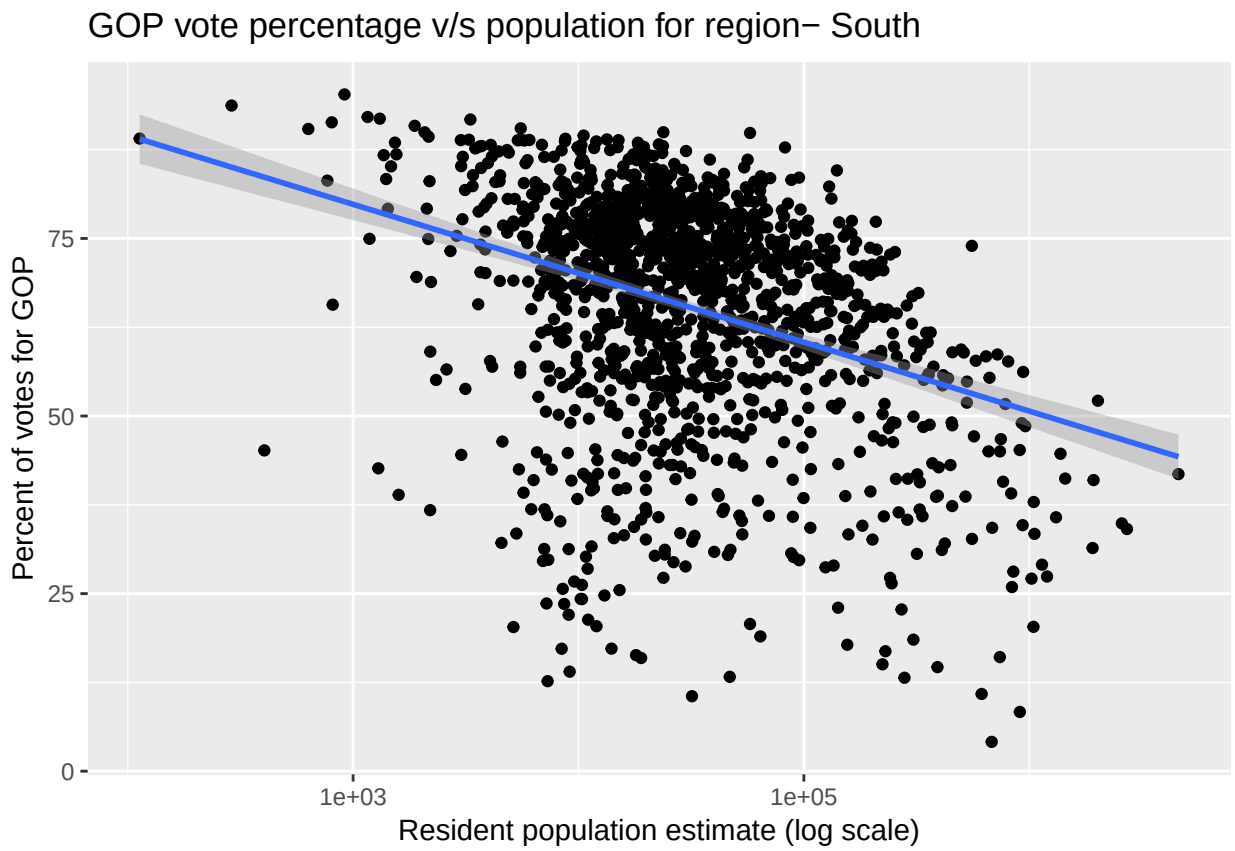
Looking at each of the 4 graphs, I see that across all the regions, an increase in the county's population is associated with an decrease in percent vote for GOP in that county. I see a particularly strong negative trend for the midwest region.

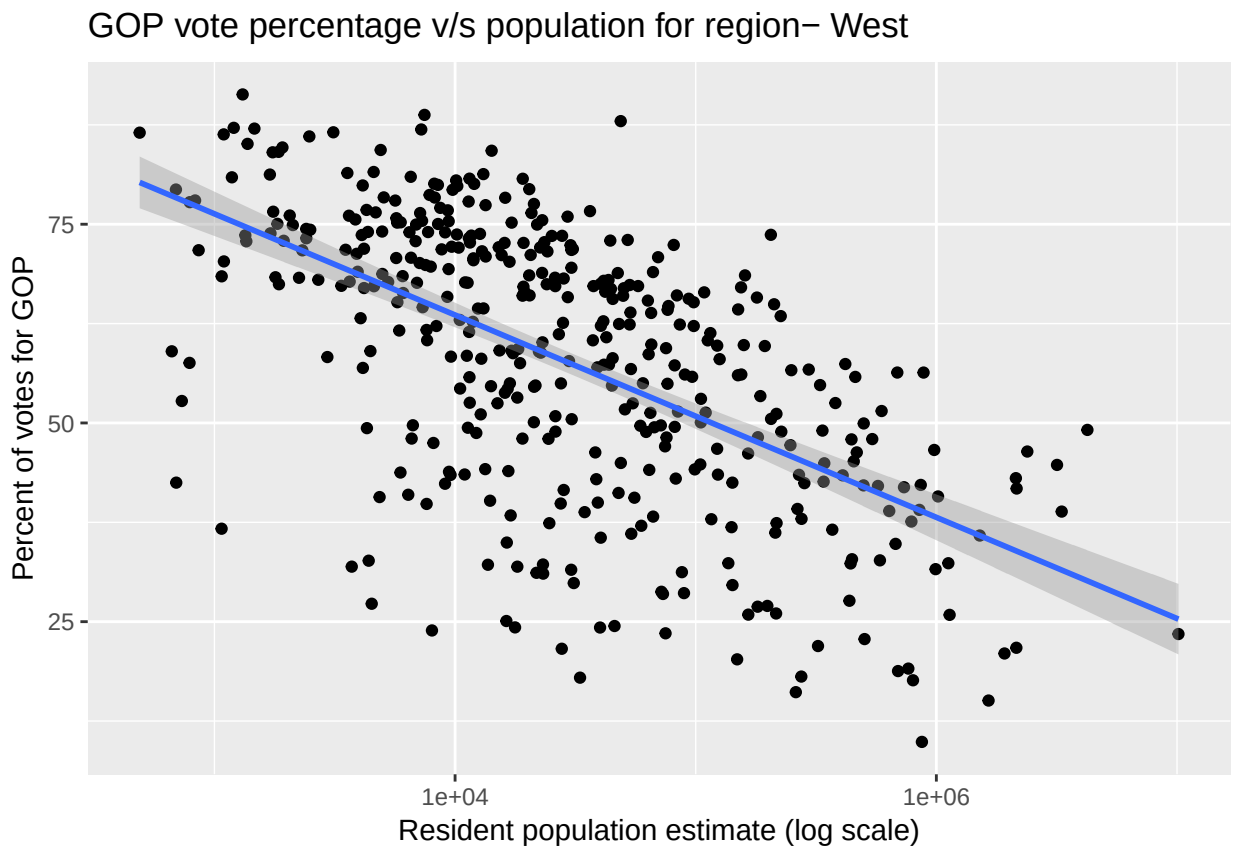
The above result might lead us to believe that states with larger populations were the ones which voted largely in favour of democrats, while states with relatively smaller populations voted largely in favour of republicans.

```
regions <- c("Northeast", "Midwest", "South", "West")
for(i in 1:4)
{
  #plot scatterplot for each of the 4 regions in the US
 print(ggplot(data=voting_data %>%
  filter(party=="republicans" & region==i),
  aes(x=resident_population_estimate,y=vote_percent))+
  geom_point()+scale_x_log10()+geom_smooth(method='lm')+
  labs(title=paste("GOP vote percentage v/s population for region-",
  regions[i]),x="Resident population estimate (log scale)",
  y="Percent of votes for GOP"))
}
```


GOP vote percentage v/s population for region− Northeast
Percent of votes for GOP
80
60
40
20
1e+04
1e+05
1e+06
Resident population estimate (log scale)


GOP vote percentage v/s population for region− Midwest
Percent of votes for GOP
75
50
25
1e+04
1e+06
Resident population estimate (log scale)


GOP vote percentage v/s population for region− South
Percent of votes for GOP
75
50
25
0
1e+03
1e+05
Resident population estimate (log scale)



Problem 2: 2016 Election Model (25pt)

Use the data from the previous problem. Your task is to estimate the probability that a county voted for democrats in 2016 elections (ie the probability that democrats received more votes than GOP). Note: you may want to include more/different variables than what you did in the previous problem.

1. List the variables you consider relevant, and explain why do you think these may matter for the election results.

Ans.

I think that that the variables which would be relevant to predict the probability that a county voted for democrats in the 2016 elections are the resident population estimate, the resident population change, the domestic and total migration numbers, and the total number of votes placed in a county.

Looking at the results of the previous problem, the data seems to suggest that a huge proportion of votes for democrats came from counties which had large populations. Hence, I feel that resident population, as well as resident population change could be a good predictor to estimate the probability of a county voting for democrats in the election. I also feel that migration patterns into a county, especially domestic migration (Migration of people eligible) to vote could influence the probability of democrats winning as that changes the demographic of voters in a county. Total votes cast in a county are also a factor which could indicate the probability of democrats winning. We earlier saw that population seemed to have a positive correlation with democrat vote percent. I would like to explore the relationship of democrats winning with votes cast as not everyone in a county votes in an election.

```
library(readr)
library(tidyverse)
#read the datasets
county_pres_results <-
```

```
  read_csv("C:/Users/admin/Downloads/US_County_Level_Presidential_Results_08-16.csv.bz2")
```

```
## Parsed with column specification:
## cols(
##   fips_code = col_character(),
##   county = col_character(),
##   total_2008 = col_integer(),
##   dem_2008 = col_integer(),
##   gop_2008 = col_integer(),
##   oth_2008 = col_integer(),
##   total_2012 = col_integer(),
##   dem_2012 = col_integer(),
##   gop_2012 = col_integer(),
##   oth_2012 = col_integer(),
##   total_2016 = col_integer(),
##   dem_2016 = col_integer(),
##   gop_2016 = col_integer(),
##   oth_2016 = col_integer()
## )
```

```
county_data <- read_csv("C:/Users/admin/Downloads/county_data.csv.bz2")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   STNAME = col_character(),
##   CTYNAME = col_character(),
##   RBIRTH2011 = col_double(),
##   RBIRTH2012 = col_double(),
##   RBIRTH2013 = col_double(),
##   RBIRTH2014 = col_double(),
##   RBIRTH2015 = col_double(),
##   RBIRTH2016 = col_double(),
##   RDEATH2011 = col_double(),
##   RDEATH2012 = col_double(),
##   RDEATH2013 = col_double(),
##   RDEATH2014 = col_double(),
##   RDEATH2015 = col_double(),
##   RDEATH2016 = col_double(),
##   RNATURALINC2011 = col_double(),
##   RNATURALINC2012 = col_double(),
##   RNATURALINC2013 = col_double(),
##   RNATURALINC2014 = col_double(),
##   RNATURALINC2015 = col_double(),
##   RNATURALINC2016 = col_double()
##   # ... with 18 more columns
## )
```

```
## See spec(...) for full column specifications.
```

```
#select fips code, county, and votes information from results dataset
#extract state and country fips codes from data
county_pres_results1 <- county_pres_results %>%
  select(fips_code,county,democrats=dem_2016,republicans=gop_2016,
  others=oth_2016,total_votes=total_2016)  %>% arrange(fips_code) %>%
```

```
  mutate(state_fips=as.integer(substr(fips_code,1,2)),
  county_fips=as.integer(substr(fips_code,3,5)))

#select region, division, state, county, population, and migration variables from county dataset
county_data1 <- county_data %>% select(region=REGION,division=DIVISION,
 state_fips=STATE,county_fips=COUNTY,state_name=STNAME,
 county_name=CTYNAME,resident_population_estimate=POPESTIMATE2016,
 resident_population_change=NPOPCHG_2016,
 international_migration=INTERNATIONALMIG2016,
 domestic_migration=DOMESTICMIG2016,net_migration=NETMIG2016,
 international_migration_rate=RINTERNATIONALMIG2016,
 domestic_migration_rate=RDOMESTICMIG2016,
 net_migration_rate=RNETMIG2016)

#join the 2 datasets on county and state fips codes
#compute percent votes for each party in each county
voting_data1 <- merge(x=county_pres_results1,y=county_data1,
 by.x=c("state_fips","county_fips"),by.y=c("state_fips","county_fips")) %>%
 arrange(fips_code)
#clean county variable by removing extraneous information
voting_data1 <- voting_data1 %>%
  mutate(county_name=substr(county,1,regexpr(" ",county)-1))
voting_data1 <- voting_data1 %>%
  mutate(winner=as.factor(as.integer(democrats>republicans)))
```

2. Estimate a logistic regression model where you explain the probability of voting democratic as a function of the variables you considered relevant. Show the results (summary).

```
#create logistic model
m <-glm(winner~resident_population_estimate+
  resident_population_change+net_migration+domestic_migration+
  total_votes, data=voting_data1, family=binomial(link = "logit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(m)
```

```
##
## Call:
## glm(formula = winner ~ resident_population_estimate + resident_population_change +
##     net_migration + domestic_migration + total_votes, family = binomial(link = "logit"),
##     data = voting_data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.4136  -0.4766  -0.4579  -0.4457   2.4045
##
## Coefficients:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -2.241e+00  6.820e-02 -32.859  < 2e-16 ***
## resident_population_estimate -3.847e-06  1.927e-06  -1.996  0.04596 *
## resident_population_change    2.230e-04  1.212e-04   1.841  0.06563 .
## net_migration                 1.048e-03  3.065e-04   3.419  0.00063 ***
## domestic_migration           -1.592e-03  2.617e-04  -6.085 1.16e-09 ***
## total_votes                   1.316e-05  4.082e-06   3.224  0.00126 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2155.2  on 3105  degrees of freedom
## AIC: 2167.2
##
## Number of Fisher Scoring iterations: 7
```

3. Experiment with a few different specifications and report the best one you got. Explain what did you do. Hint: we did not talk about choosing between different logistic regression models. You may use a pseudo-R2 value in a similar fashion as you use R2 for linear models. For instance, pscl::pR2 will provide a number of different pseudo-R2 values for estimated glm models, you may pick McFadden's version.

Ans.

pR2 function from the pscl library provides a pseudo-R2 values for estimated logistic regression models, I would be using McFadden's version for tesing the goodness of fit. The pseudo R2 value for the first model (m) came out to be 0.20068, indicating that the model is not a very good fit for the observed data.

In the second model (m2), I replaced the migration numbers with rates, and added the international migration rate as well to check if adding migration rates instead of numbers might improve the goodness of fit. I also felt that taking the log value of resident population estimate might improve the model. The goodness of fit improved for the model with the pseudo R2 value going up to 0.2303363, which indicates a better fit to the observed data.

For my last model (m3), I used the migration numbers instead of the rates and removed international migration, as I felt that international immigrants generally do not have voting rights, and an influx/outflow of international migrants to a county might not impact the probability of democrats winning there as much as an influx or ouflow of domestic migrants might. I also tried to experiment by only keeping the resident population change and not the resident population number as one of my predictors. This lowered my pseudo R squared value to be even lower than my first value. The R2 value for this model came out as 0.1990827. This model seems to have the worst fit to the data.

Therefore my best fit model in this case was m2 with a pseudo R2 value of 0.2303363.

```
library(pscl)
```

```
## Warning: package 'pscl' was built under R version 3.4.3
```

```
## 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
```

```
#create logistic model 2
m2 <-glm(winner~log(resident_population_estimate)+
  resident_population_change+net_migration_rate+
  domestic_migration_rate+international_migration_rate+
  total_votes, data=voting_data1, family=binomial(link = "logit"))
summary(m2)
```

```
##
## Call:
```

```
## glm(formula = winner ~ log(resident_population_estimate) + resident_population_change +
##     net_migration_rate + domestic_migration_rate + international_migration_rate +
##     total_votes, family = binomial(link = "logit"), data = voting_data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0712  -0.4852  -0.3905  -0.3104   2.7695
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -5.680e+00  6.690e-01  -8.491  < 2e-16
## log(resident_population_estimate)  2.942e-01  6.589e-02   4.465 8.00e-06
## resident_population_change      -8.222e-05  2.432e-05  -3.380 0.000724
## net_migration_rate               1.029e+07  3.490e+07   0.295 0.768115
## domestic_migration_rate         -1.029e+07  3.490e+07  -0.295 0.768115
## international_migration_rate    -1.029e+07  3.490e+07  -0.295 0.768115
## total_votes                      7.295e-06  1.459e-06   5.001 5.71e-07
##
## (Intercept)                     ***
## log(resident_population_estimate) ***
## resident_population_change      ***
## net_migration_rate
## domestic_migration_rate
## international_migration_rate
## total_votes                     ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2075.2  on 3104  degrees of freedom
## AIC: 2089.2
##
## Number of Fisher Scoring iterations: 5
```

```
#create logistic model 3
m3 <-glm(winner~resident_population_change+net_migration+
 domestic_migration+total_votes, data=voting_data1,
 family=binomial(link = "logit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(m3)
```

```
##
## Call:
## glm(formula = winner ~ resident_population_change + net_migration +
##     domestic_migration + total_votes, family = binomial(link = "logit"),
##     data = voting_data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.5682  -0.4777  -0.4574  -0.4464   2.3556
##
```

```
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -2.252e+00  6.799e-02 -33.123  < 2e-16 ***
## resident_population_change  4.373e-05  8.543e-05   0.512 0.608762
## net_migration               1.227e-03  2.947e-04   4.163 3.14e-05 ***
## domestic_migration         -1.590e-03  2.571e-04  -6.186 6.16e-10 ***
## total_votes                 5.615e-06  1.662e-06   3.379 0.000727 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2159.5  on 3106  degrees of freedom
## AIC: 2169.5
##
## Number of Fisher Scoring iterations: 7
```

```
#estimate pseudo R2 for models
pR2(m)[4]
```

```
## McFadden
##  0.20068
```

```
pR2(m2)[4]
```

```
##  McFadden
## 0.2303363
```

```
pR2(m3)[4]
```

```
##  McFadden
## 0.1990827
```

4. Explain the meaning of statistical significance. What does it mean that an estimated coefficient is statistically significant (at 5% confidence level)?

Ans.

Statistical significance of a coefficient in an an estimated model basically tests if the predicted relationship in the model betweenthe predictor and response variables is not due to random chance.

Statistical significance of a variable in a model is indicated by its p-value. The p-value for each term is the result of a hypothesis test for the model which tests if the null hypothesis that the coefficient is equal to zero. A low p-value (< 0.05) indicates that you can reject the null hypothesis that the relationship we see between the predictor and outcome is due to random chance.

A larger p-value suggests that the data does not present enough evidence to reject the null hypothesis that the relationship seen between the predictor and outcome might be due to random chance.

For 95% confidence, a p-value of less than 0.05 for a variable indicates that the relationship described between it and the response variable in the model might not be due to random chance.

5. Indicate which results are statistically significant in your preferred model.

Ans. In my prefered model, the results which are statistically significant are the following as their p-values are lower than the critical p value of 0.05.

log(resident_population_estimate) with a p value of- 8.00e-06

resident_population_change with p value of - 0.000724

total_votes with p value of - 5.71e-07

Net migration rate, domestic migration rate, and internationa migration rate do not seen statistically significant as their p values are not lower than 0.05.

6. Interpret the results. Provide correct interpretable explanations about what the most important effect are and what do the particular numeric results mean. Hint: you may use either odds ratios or marginal effects.

Ans.

The following are the interpretations of the model coefficients:

For a unit increase in log(resident_population_estimate), the log odds of democrats winning in a county are likely to change by 2.942e-01. (Log odds likely to increase)

For a unit increase in resident_population_change, the log odds of democrats winning in a county are likely to change by -8.222e-05. (Log odds likely to decrease)

For a unit increase in resident_population_change, the log odds of democrats winning in a county are likely to change by 5.615e-06. (Log odds likely to increase)

Problem 3: Simulate the Effect of Additional Random Coefficients (25pt)

Here your task is to simulate the logit coeffcients of irrelevant input variables. You may either pick your favorite model from above, or use a different specification.

1. Choose a distribution. Poisson is fine, but you may pick something else as well.

(a) Create a vector of random numbers, exactly as long as many observations you have in your data.

```
#create vector of random numbers from poisson distribution
rand_nos <- rpois(nrow(voting_data1),lambda=100)
```

(b) Estimate the logistic regression model using your former specification, but adding the random number as an additional explanatory variable.

```
#create glm with random number as explanatory variable
m4 <-glm(winner~log(resident_population_estimate)+
  resident_population_change+net_migration_rate+
  domestic_migration_rate+international_migration_rate+total_votes+
  rand_nos, data=voting_data1, family=binomial(link = "logit"))
summary(m4)
```

```
##
## Call:
## glm(formula = winner ~ log(resident_population_estimate) + resident_population_change +
##     net_migration_rate + domestic_migration_rate + international_migration_rate +
##     total_votes + rand_nos, family = binomial(link = "logit"),
##     data = voting_data1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0558  -0.4890  -0.3914  -0.3096   2.7838
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -5.005e+00  8.921e-01  -5.610 2.02e-08
## log(resident_population_estimate)  2.924e-01  6.589e-02   4.438 9.09e-06
## resident_population_change      -8.298e-05  2.429e-05  -3.416 0.000635
## net_migration_rate               1.158e+07  3.495e+07   0.331 0.740436
```

```
## domestic_migration_rate           -1.158e+07  3.495e+07  -0.331 0.740436
## international_migration_rate      -1.158e+07  3.495e+07  -0.331 0.740436
## total_votes                        7.331e-06  1.460e-06   5.023 5.09e-07
## rand_nos                          -6.602e-03  5.803e-03  -1.138 0.255277
##
## (Intercept)                       ***
## log(resident_population_estimate) ***
## resident_population_change        ***
## net_migration_rate
## domestic_migration_rate
## international_migration_rate
## total_votes                       ***
## rand_nos
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2073.9  on 3103  degrees of freedom
## AIC: 2089.9
##
## Number of Fisher Scoring iterations: 5
```

(c) store the coefficient for the random variable. Hint: function coef gives you the estimated coefficients of the model. It is a named vector, you can extract the coefficient of interest as coef(m)["varname"] where m is the estimated model and "varname" is the name of the variable of interest.

```
#Extract coefficient of random variable
rand_coeff <- coef(m4)['rand_nos']
print(rand_coeff)
```

```
##     rand_nos
## -0.006601886
```

(d) repeat these steps a large number R > 1000 times. Now you have R estimates of the coefficent for pure carbage features.

```
start_time <- Sys.time()
R <- 2000
coefficients_list <- c()
#repeat the estimation of coefficients step R times for different sets of poisson random variable
for(i in 1:R)
{
  rand_nos1 <- rpois(nrow(voting_data1),lambda=100)
  m5 <-glm(winner~log(resident_population_estimate)+
    resident_population_change+net_migration_rate+
    domestic_migration_rate+international_migration_rate+
    total_votes+rand_nos1, data=voting_data1,
    family=binomial(link = "logit"))
  rand_coeff1 <- coef(m5)['rand_nos1']
  coefficients_list <- append(coefficients_list,rand_coeff1)
}

end_time <- Sys.time()
#compute time taken for sequential calculations
```

```
print(end_time-start_time)
```

```
## Time difference of 36.15507 secs
```

2. What are the (sample) mean and (sample) standard deviation of the estimated coefficients?

Ans. The sample mean of the estimated coefficients is -0.0001977636 The sample standard deviation of the estimated coefficients is 0.005710202

```
#compute mean and sd of coefficients
coef_mean <- mean(coefficients_list)
coef_sd <- sd(coefficients_list)
print(coef_mean)
```

```
## [1] -4.426189e-05
```

```
print(coef_sd)
```

```
## [1] 0.005690284
```

3. Find the 95% confidence interval of the coefficient based on your simulations.

Ans. The 95% confidence interval of the coefficents is (-0.01115788 ,0.01092059)

```
#compute 95% confidence interval of coefficients
quantile(coefficients_list,0.025)
```

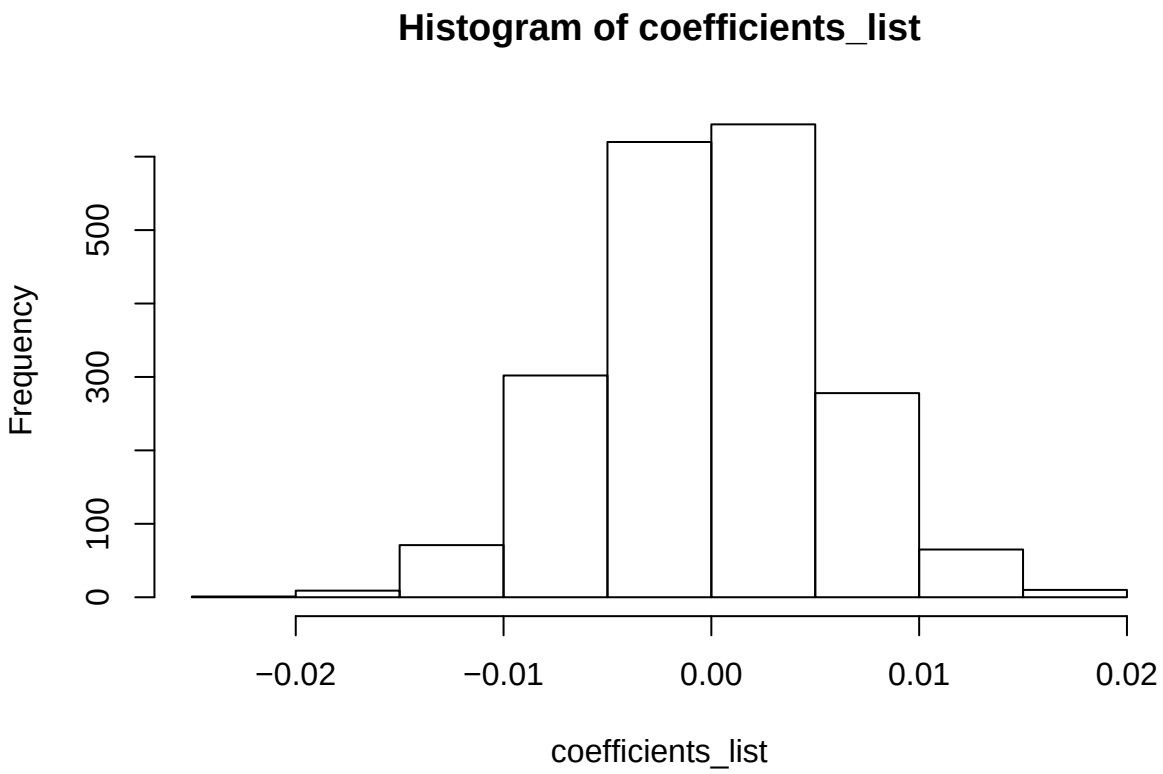
```
##       2.5%
## -0.0112067
```

```
quantile(coefficients_list,0.975)
```

```
##      97.5%
## 0.01090139
```

4. Plot the distribution of the estimates (histogram, or another density plot).

```
#plot histogram of coefficients
hist(coefficients_list)
```



5. Assume the estimates are randomly distributed with mean and standard deviation as you found above. What are the theoretical 95% confidence intervals for the results?

Ans. The 95% theoretical confidence interval of the coefficents would be (-0.01138976 ,0.01099423) as they would be distributed around the mean with 95% confidence interval of: mean +/- (1.96 * SD)

```
#compute theoretical 95% confidence intervals
zscore <- 1.96
upper <- coef_mean + zscore*coef_sd
lower <- coef_mean - zscore*coef_sd
print(lower)
```

```
## [1] -0.01119722
```

```
print(upper)
```

```
## [1] 0.0111087
```

6. Extra credit (2pt): run the simulations in parallel. Report how much faster did it go compared to sequential processing.

Ans. Sequential processing took approximately 35.71104 secs. Parallel processing took about 24.74642 secs. Parallel processing had a performance improvement of about 11 secs.

Reference: https://www.r-bloggers.com/5-ways-to-measure-running-time-of-r-code/

```
#use doparallel and foreach libraries
library(doParallel)
```

```
## Warning: package 'doParallel' was built under R version 3.4.3
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.4.3
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loading required package: iterators
```

```
## Loading required package: parallel
```

```
library(foreach)
registerDoParallel(cores=20)
start_time1 <- Sys.time()
R <- 2000
#repeat the estimation of coefficients step R times for different sets of poisson random variable
coeffval <-foreach(i <- 1:R, .combine=append)%dopar%
  {
    rand_nos2 <- rpois(nrow(voting_data1),lambda=100)
  m6 <-glm(winner~log(resident_population_estimate)+
   resident_population_change+net_migration_rate+
   domestic_migration_rate+international_migration_rate+
   total_votes+rand_nos2, data=voting_data1,
   family=binomial(link = "logit"))
  rand_coeff1 <- coef(m6)['rand_nos2']
    }

end_time1 <- Sys.time()
#compute time taken  for parallel execution
print(end_time1-start_time1)
```

```
## Time difference of 24.4574 secs
```

Problem 4:

1(a-b) and 2(a-d) Solved on paper and attached to this exam

2(e)

```
library(maxLik)
```

```
## Warning: package 'maxLik' was built under R version 3.4.3
```

```
## Loading required package: miscTools
```

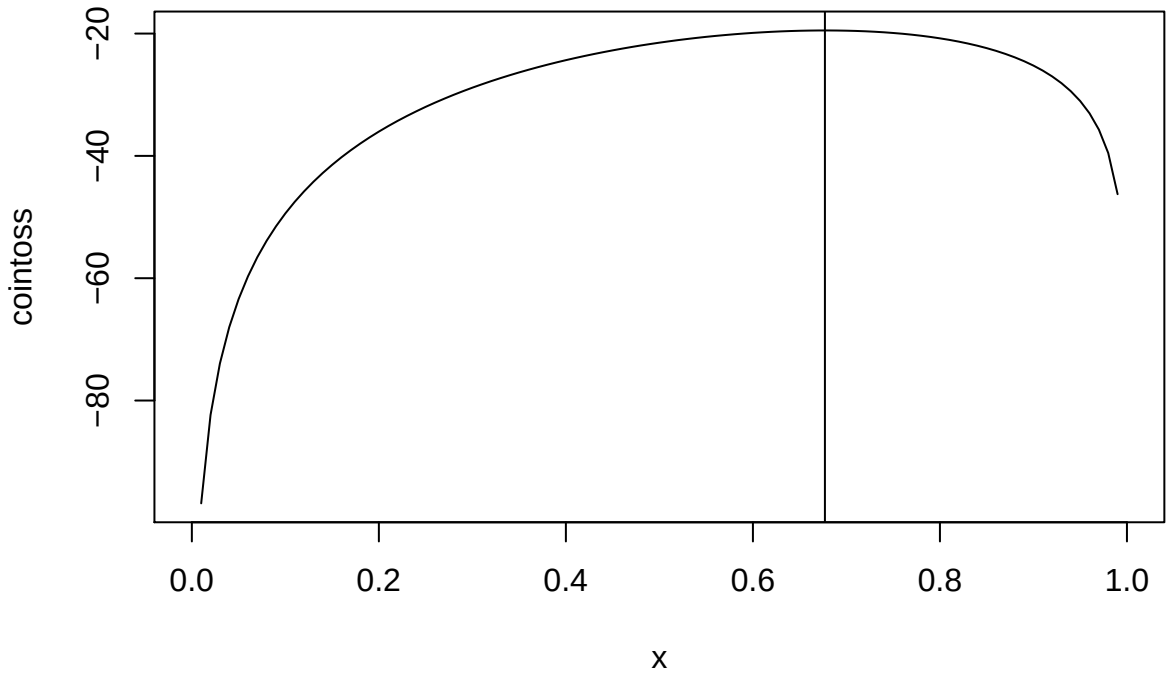
```
## Warning: package 'miscTools' was built under R version 3.4.3
```

```
##
## Please cite the 'maxLik' package as:
## Henningsen, Arne and Toomet, Ott (2011). maxLik: A package for maximum likelihood estimation in R. C
##
## If you have questions, suggestions, or comments regarding the 'maxLik' package, please use a forum o
## https://r-forge.r-project.org/projects/maxlik/
```

```
cointoss <- function(p) {21*log(p)+ 10*log(1-p)}
maxLik::maxLik(cointoss,start=0.1)
```

```
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 6 iterations
## Return code 1: gradient close to zero
## Log-Likelihood: -19.49278 (1 free parameter(s))
## Estimate(s): 0.6774194
```

```
plot(cointoss)
abline(v=0.677)
```



Statement of Compliance

Please copy and sign the following statement. You may do it on paper (and include the image file), or add the following text with your name and date in the rmarkdown document.

I affirm that I have had no conversation regarding this exam with any persons other than the instructor or the teaching assistant. Further, I certify that the attached work represents my own thinking. Any information, concepts, or words that originate from other sources are cited in accordance with University of Washington guidelines as published in the Academic Code (available on the course website). I am aware of the serious consequences that result from improper discussions with others or from the improper citation of work that is not my own.

Rajendran Seetharaman 11th December, 2017