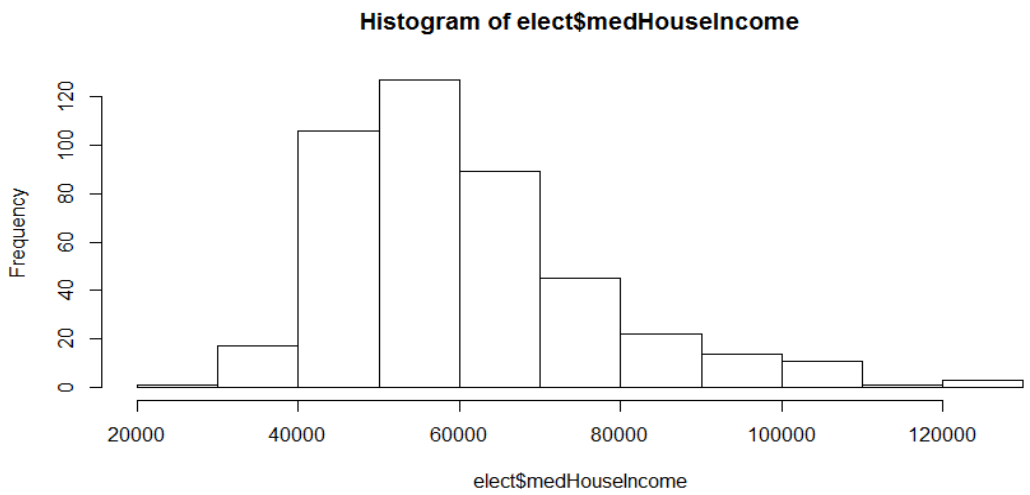
Data Analysis Report

1. From the perspective of background information introduction, U.S. House of Representatives (2018, Wiki), is the lower chamber of the United States Congress, the Senate being the upper chamber. Together they comprise the legislature of the United States. Congressional districts in the United States are electoral divisions for the purpose of electing members of the United States House of Representatives. (2018, Wiki) The 115th U.S. Congress is the current meeting of the legislative branch of the United States federal government, composed of the Senate and the House of Representatives. (2018, Wiki) U.S. political parties, (2018, Wiki) include two main and some other small ones. These two main parties have been the Republican Party and the Democratic Party. The Democratic Party has the most seats in the House of Representatives while the Republicans hold a majority in the senate.

2. The data file given includes five variables in total. They are, respectively, state, district, electedrep, party and medHouseIncome. State means the region divided in order to govern the country. There are in total 50 states in the USA, and this is used as a vote unit when it comes to choose who would be the next president of the country. District is the region divided under the dimension of state, every state has corresponding districts. Some large state has tens of districts, and

small state has several districts. The number of districts depends on population and area involved, and they are distributed with consideration by law. Electedrep means elected representative, the person elected by people in the region to the Congress. Party means the party affiliation of the Representative during the 115^{th} Congress. In specific, there are two major parties in the USA. Here, D stands for Democrat, R stands for Republican, the two main parties in the election. MedHouseIncome means median household income. MedHouseIncome can be the variable with most numeric information. The max median income we have here is 125790 where the representative is Anna Eshoo. The min median income we have here is 29234 where the representative is Jose E. Serrano. The histogram of medHouseIncome is shown below:



3.(a)

model {

for (i in 1:length(res)) {

```
        res[i] ~ dbern(prob[i])

        logit(prob[i]) <- betaintercept + betaincome * incomescaled[i]

        resrep[i] ~ dbern(prob[i])

    }

    betaintercept ~ dt(0, 0.01, 1)

    betaincome ~ dt(0, 0.16, 1)

}
```

(b) I used four chains as usual in the previous projects. Initialize intercept and slope with 10 and -10, respectively. Burn-in step includes 10000 iterations. Number of iterations per chain is 20000. No thinning used. Effective sizes of betaintercept(intercept parameter) and betaincome(slope parameter) are 48995 and 50570, respectively.

(c)

	betaintercept	betaincome
Posterior mean	-0.2145	0.3608
Posterior standard deviation	0.09652	0.19376
95% confidence interval	(-0.40454,-0.02565)	(-0.01946,0.74244)

(d) 0.9689125, it is obvious that as median household income increases, the probability of electing a Democrat increases.

(e)DIC is 600, effective number of parameters are about 2. The effective number of parameters is almost the same as the actual number of parameters.

4.(a)

```
model {
    for (i in 1:length(res)) {
        res[i] ~ dbern(prob[i])
        logit(prob[i]) <- betaintercept + betastate[state[i]] + betaincome *
incomescaled[i]
        resrep[i] ~ dbern(prob[i])
    }
    for (j in 1:max(state)) {
        betastate[j] ~ dnorm(0, 1/sigmastate[j]^2)
        sigmastate[j] ~ dunif(0, 1000)
    }
    betaintercept ~ dt(0, 0.01, 1)
    betaincome ~ dt(0, 0.16, 1)
}
```

(b) I used four chains as usual in the previous projects. Initialize intercept and slope with 10 and -10, respectively. Burn-in step includes 10000 iterations. Number of iterations per chain is 20000. No thinning used.

```
> effectiveSize(x2[,1:54])
    betaincome  betaintercept    betastate[1]
    22101.0824           514.4019          2138.9014
```

betastate[2]	betastate[3]	betastate[4]
12151.0923	1017.5236	12724.5728
betastate[5]	betastate[6]	betastate[7]
613.0937	1192.4288	13501.7262
betastate[8]	betastate[9]	betastate[10]
12102.5173	11672.7863	622.3312
betastate[11]	betastate[12]	betastate[13]
845.0911	13004.3195	12193.1992
betastate[14]	betastate[15]	betastate[16]
729.2299	1226.9855	2359.8617
betastate[17]	betastate[18]	betastate[19]
11732.2055	2454.0024	2158.5104
betastate[20]	betastate[21]	betastate[22]
4370.5451	2048.1829	12337.9692
betastate[23]	betastate[24]	betastate[25]
805.6643	1026.7342	2310.4370
betastate[26]	betastate[27]	betastate[28]
1260.8094	11925.5283	12904.3440
betastate[29]	betastate[30]	betastate[31]
2386.7891	12053.3730	923.7962
betastate[32]	betastate[33]	betastate[34]
2877.3082	691.5578	996.1072

```
betastate[35]  betastate[36]  betastate[37]
   12018.5966       817.1567     11849.4395
betastate[38]  betastate[39]  betastate[40]
    2302.5174       748.4923     11432.7346
betastate[41]  betastate[42]  betastate[43]
    2227.8284     12223.6166      1220.1849
betastate[44]  betastate[45]  betastate[46]
     644.2265     12026.5556     11817.1071
betastate[47]  betastate[48]  betastate[49]
    1005.8876       963.0579     12088.1189
betastate[50]  betastate[51]
    1054.7602     11334.5769
```

(c)

	betaintercept	betaincome
Posterior mean	-0.7362	-1.0747
Posterior standard deviation	0.8041	0.2880
95% confidence interval	(-2.236,1.0615)	(-1.651,-0.5211)

(d)

```
> mean(as.numeric(unlist(x2[,1]))>0)
[1] 3.75e-05
```

As median household income increases, after adjustment for state, there is little probability of electing a Democrat.

(e) The 40th state, Rhode Island has the largest posterior mean random effect. The 37th state, Oklahoma has the smallest posterior mean random effect. Rhode Island apparently supports Democrat, Oklahoma apparently supports Republican.

(f)

penalty NaN

Penalized deviance: NaN

Both parameters are not numbers. So the second model is worse than the first one.

5. The results show that the vote preference depends on the random effect of state, which means different states have different preferences on the party support. The median house income behaves oppositely between with states and without states random effect. It means, within the country, generally, higher house income people tends to support Democrat, while, within different states, higher house income people tends to support Republican. In another word, rich states tend to support Democrat, while rich people in every state tend to support Republican. In the end, the Bayesian statistical generation for the second model with random effect of states cannot produce effective results.

REFERENCES

United States House of Representatives. (2018, December 7). In Wikipedia, The Free Encyclopedia. Retrieved December 16, 2018 from https://en.wikipedia.org/wiki/United_States_House_of_Representatives

List of United States congressional districts. (2018, December 15). In Wikipedia, The Free Encyclopedia. Retrieved December 16, 2018 from https://en.wikipedia.org/wiki/List_of_United_States_congressional_districts

115th United States Congress, (2018, December 15). In Wikipedia, The Free Encyclopedia. Retrieved December 16, 2018 from https://en.wikipedia.org/wiki/115th_United_States_Congress

Political parties in the United States, (2018, November 9). In Wikipedia, The Free Encyclopedia. Retrieved December 16, 2018 from https://simple.wikipedia.org/wiki/Political_parties_in_the_United_States

APPENDIX

First Model

```
> elect = read.csv(file="party115cong.csv", header = TRUE)
> unclass(elect$party)
> res = unclass(elect$party)
> res = 2 - res
> logmed = log(elect$medHouseIncome, exp(1))
> d1 <- list(res = res, incomescaled = as.vector(scale(logmed-mean(l
ogmed),scale=2*sd(logmed))))
> inits1 <- list(list(betaintercept=10,betaincome=10),list(betaintercept
=10,betaincome=-10),list(betaintercept=-10,betaincome=10),list(betaint
ercept=-10,betaincome=-10))
> library(rjags)
> m1 <- jags.model("elect.bug", d1, inits1, n.chains=4, n.adapt = 10
000)
> update(m1, 10000)
> x1 <- coda.samples(m1, c("betaintercept","betaincome"), n.iter=20
000)
> gelman.diag(x1, autoburnin=FALSE)
> x1 <- coda.samples(m1, c("betaintercept","betaincome","prob","res
rep"),n.iter=20000)
> summary(x1[,1:2])
```

Iterations = 40001:60000

Thinning interval = 1

Number of chains = 4

Sample size per chain = 20000

1. Empirical mean and standard deviation for each variable,

   plus standard error of the mean:

	Mean	SD	Naive SE
betaincome	0.3608	0.19376	0.0006850
betaintercept	-0.2145	0.09652	0.0003412

	Time-series SE
betaincome	0.0008755
betaintercept	0.0004296

2. Quantiles for each variable:

	2.5%	25%	50%
betaincome	-0.01946	0.2304	0.3601
betaintercept	-0.40454	-0.2796	-0.2144

	75%	97.5%
betaincome	0.4912	0.74244

```
betaintercept  -0.1494  -0.02565


>  effectiveSize(x1[,1:2])
    betaincome  betaintercept
       48995.05          50570.36
>  mean(as.numeric(unlist(x1[,1]))>0)
[1]  0.9689125
>  dic.samples(m1,200000)
  |**************************************************|  100%
Mean  deviance:    598
penalty  1.991
Penalized  deviance:  600
```

Second Model

```
>  state  =  unclass(elect$state)
>  d2  <-  list(res  =  res,  state  =  state,  incomescaled  =  as.vector(scale
(logmed-mean(logmed),scale=2*sd(logmed))))
>  inits2  <-  list(list(betaintercept=10,  betastate=sample(x  =  c(-10,10),
  size=51,  replace=TRUE),betaincome=10),list(betaintercept=10,betastat
e=sample(x  =  c(-10,10),  size=51,  replace=TRUE),betaincome=-10),list
(betaintercept=-10,betastate=sample(x  =  c(-10,10),  size=51,  replace=T
```

```
RUE),betaincome=10),list(betaintercept=-10,betastate=sample(x = c(-1
0,10), size=51, replace=TRUE),betaincome=-10))
> m2 <- jags.model("elect.bug",d2,inits2,n.chains=4,n.adapt=10000)
> update(m2,10000)
> x2 <- coda.samples(m2, c("betaintercept","betastate","betaincome
"),n.iter=20000)
> gelman.diag(x2, autoburnin=FALSE)
> x2 <- coda.samples(m2,c("betaintercept","betastate","betaincome",”
prob”,”resrep”),n.iter=20000)
> summary(x2[,1:2])

Iterations = 60001:80000
Thinning interval = 1
Number of chains = 4
Sample size per chain = 20000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                 Mean     SD  Naive SE
betaincome     -1.0747  0.2880  0.001018
betaintercept  -0.7362  0.8041  0.002843
```

```
                     Time-series SE
betaincome                0.001939
betaintercept             0.036079

2. Quantiles for each variable:

                 2.5%     25%     50%
betaincome     -1.651  -1.267  -1.0707
betaintercept  -2.236  -1.234  -0.8031
                  75%    97.5%
betaincome     -0.8781  -0.5211
betaintercept  -0.2997   1.0615

> dic.samples(m2,200000)
  |**************************************************| 100%
> dic.samples(m2,200000)
  |**************************************************| 100%
Mean deviance:  475.9
penalty NaN
Penalized deviance: NaN
```