

# STT465\_Report\_No\_Code

Sam Isken

December 9, 2019

## STT 465 Final Report:

### Examining a PRIOR Election: A Look into 2000 Elections Using OLS and Bayesian Techniques

This PDF contains the written report (no code visible). The other RMarkdown below contains the code as well as parts of the essay portions of this project (ones that are relevant to the coding scheme). I have also included a supplemental dataset I used to examine the election results at the county level of granularity.

```
knitr::opts_chunk$set(echo = TRUE)
library(faraway)
library(ggplot2)
library(coda)
library(rjags)
```

```
## Linked to JAGS 4.3.0
```

```
## Loaded modules: basemod,bugs
```

```
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
data("gavote")
gavote$undercount<- (gavote$ballots - gavote$votes)/gavote$ballots
```

## Introduction

The 2000 United States Presidential Election was bound to be contentious. George W. Bush, a conservative with a family legacy in the political sphere was running against incumbent Vice President (to Moderate Democrat Bill Clinton) Al Gore. Both ran campaigns focused on domestic issues with some topics of foreign policy sprinkled in. The conservatives emphasized "family values" after Clinton's scandal regarding Monica Lewinsky while Gore criticized Bush's lack of political experience and conservative policy regarding Healthcare.

After election night it was clear this election would not go as smoothly as planned. The biggest story to come out was that of the "hanging chad" which made voters in Florida unsure of for whom they were casting their ballot and lead to the famous Florida recount (which was interrupted when a conservative group stormed the recount offices, led by recently convicted felon Roger Stone). This led to a general skepticism on the part of the American people about the security and fairness of the election results. This was eventually decided in the SCOTUS, which handed the election to Bush.

This paper will focus on another election night controversy, the state of Georgia. We are looking to examine the undercount of votes granularly at the county level. The undercount is defined as the difference between the number of ballots cast and votes recorded for president. Voters may have chosen not to vote for president, voted for more than one candidate (disqualified) or the equipment may have failed to register their choice. (Citation: " `{r} help(gavote)` "). Our goal, is to determine the factors that significantly predict the undercount at the county level, which in turn could be used to make inferences for future elections. Our data contains the following variables to assist with this prediction:

## Data Summary

equip: The voting equipment used: LEVER, OS-CC (optical, central count), OS-PC (optical, precinct count) PAPER, PUNCH

econ: economic status of county: middle poor rich

perAA: percent of African Americans in county

rural: indicator of whether county is rural or urban

atlanta: indicator of whether county is in Atlanta or not: notAtlanta

gore: number of votes for Gore

bush: number of votes for Bush

other: number of votes for other candidates

votes: number of votes

ballots: number of ballots

undercount: (number of ballots - number of votes) / (number of ballots)

(Citation: Meyer M. (2002) Uncounted Votes: Does Voting Equipment Matter? Chance, 15(4), 33-38 & " `{r} help(gavote)` ")

## Surface Level Data Analysis

```
print(paste("The range of the undercount (in percent of county votes) is:",range(gavote$undercount)[1]," , ",range(gavote$undercount)[2]))
```

```
## [1] "The range of the undercount (in percent of county votes) is: 0 , 0.1881205364919"
```

```
print(paste("The mean of undercount (in percent of county votes) is:",mean(gavote$undercount)))
```

```
## [1] "The mean of undercount (in percent of county votes) is: 0.0437918622221233"
```

```
upper <- max(gavote$undercount)
```

```
print(paste("The max value of undercount (in percent of county of county votes) is: " , upper))
```

```
## [1] "The max value of undercount (in percent of county of county votes) is: 0.1881205364919"
```

From a surface level analysis we can see that in some counties, up to 18% of ballots were thrown out in some counties due to voting or machine errors. The average percent of votes thrown out by county was 4%. This quick look proves that this is an important topic as this easily could have swung counties in favor of the other candidate (either Gore or Bush). This provides justification for our research.

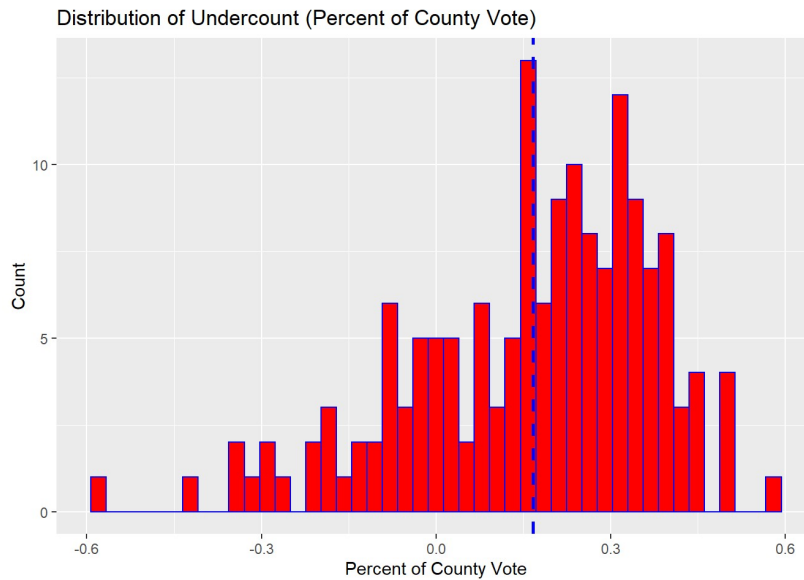
```
Georgia_County_Level_Results <- read.csv("Georgia_County_Level_Results.csv")
Georgia_County_Level_Results[] <- lapply(Georgia_County_Level_Results, as.numeric)
Georgia_County_Level_Results$Percent_Dif <- Georgia_County_Level_Results$Bush_Percent - Georgia_County_Level_Results$Gore_Percent
Georgia_County_Level_Results$within_margin_of_undercount <- ifelse(Georgia_County_Level_Results$Percent_Dif<=upper,1,0)
percent_counties_within_undercount_margin <- sum(Georgia_County_Level_Results$within_margin_of_undercount)/nrow(Georgia_County_Level_Results)
head(Georgia_County_Level_Results)
```

	County <dbl>	Gore_Percent <dbl>	Gore_N... <dbl>	Bush_Percent <dbl>	Bush_N... <dbl>	Others_Percent <dbl>	Others_Num <dbl>	Total <dbl>	Percent_Dif <dbl>
1	1	0.34	58	0.65	101	0.01	66	118	0.31
2	2	0.40	146	0.59	6	0.01	22	31	0.19
3	3	0.32	152	0.67	52	0.01	29	47	0.35
4	4	0.59	150	0.40	141	0.01	11	5	-0.19
5	5	0.49	126	0.50	134	0.02	192	20	0.01
6	6	0.27	10	0.71	82	0.02	111	91	0.44

6 rows | 1-10 of 11 columns

The data set above provides the election results at the county level for Georgia. This will assist in providing a basis / set of background knowledge for our project.

```
# Visualizations
ggplot(Georgia_County_Level_Results, aes(x=Percent_Dif)) + geom_histogram(bins=45,color="blue", fill="red") + geom_vline(aes(xintercept=mean(Percent_Dif)),color="blue", linetype="dashed", size=1)+
  labs(title="Distribution of Undercount (Percent of County Vote)",x="Percent of County Vote", y = "Count")
```



```
print(paste(percent_counties_within_undercount_margin,"were within margin of error max (18%)"))
```

```
## [1] "0.459119496855346 were within margin of error max (18%)"
```

As we can see roughly 45% of county results were within the margin of the undercount (18%). This, coupled with the fact that Bush only won by a relatively narrow margin (54.67% to 42.98%) prompts the analysis of this data.

(Citation: [https://en.wikipedia.org/wiki/2000\\_United\\_States\\_presidential\\_election\\_in\\_Georgia](https://en.wikipedia.org/wiki/2000_United_States_presidential_election_in_Georgia) ([https://en.wikipedia.org/wiki/2000\\_United\\_States\\_presidential\\_election\\_in\\_Georgia](https://en.wikipedia.org/wiki/2000_United_States_presidential_election_in_Georgia)))

This concludes the argument in favor (justification) of conducting this study. We will now discuss the final models and methods used to create the models (OLS and Bayesian) present in this paper.

## Final Models

### OLS Frequentist Regression

Final Model (OLS Frequentist Regression):

$$\text{UNDERCOUNT} = 5.807496 * 10^{-3}x_1 + 2.273752 * 10^{-2}x_2 - 9.044148 * 10^{-3}x_3 + 6.082960 * 10^{-3}x_4 + 2.089696 * 10^{-2}x_5 - 1.894076 * 10^{-2}x_6 \\ - 3.554025 * 10^{-5}x_7 - 3.386005 * 10^{-5}x_8 + 3.282120 * 10^{-5}x_9 + 2.898298 * 10^{-2} + \epsilon_i$$

$$\text{UNDERCOUNT} = \beta_1 + \text{EQUIP}x_1 + \text{ECON}x_2 + \text{GORE}x_3 + \text{BUSH}x_4 + \text{BALLOTS}x_5 + \epsilon_i$$

The above functions show both the specific numerical coefficients within our regression function as well as the generic variables they represent.

### Bayesian Analysis

Within a Bayesian Analysis (applied using a Normal Model) our result is our prior, posteriors and likelihood function:

Final Model (Bayesian Approach):

Likelihood function:

Likelihood Function of Undercount (Y):

$$p(\text{UNDERCOUNT}(Y)|\text{EQUIP},\text{ECON},\text{GORE},\text{BUSH},\text{BALLOTS}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(\text{UNDERCOUNT}_i - \beta_1 - \text{EQUIP}_2x_i - \text{ECON}_3x_i - \right. \\ \left. - \text{BUSH}_5x_i - \text{BALLOTS}_6x_i)^2\right\}$$

Prior:

Flat Prior:

$$p(\beta_1, \dots, \beta_p) \propto 1$$

Posterior Distributions:

Posterior Distribution of Regression Coefficients:

$$p(\text{EQUIP},\text{ECON},\text{GORE},\text{BUSH},\text{BALLOTS}|\text{UNDERCOUNT}(Y)) \propto 1 \times \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(\text{UNDERCOUNT}_i - \beta_1 - \text{EQUIP}_2x_i - \text{ECON}_3x_i - \right. \\ \left. - \text{BUSH}_5x_i - \text{BALLOTS}_6x_i)^2\right\}$$

In the generic formed a posterior is defined in the form of:

$$p(\theta|y) \propto p(\theta) * p(y|\theta)$$

These results will be further explained in both the Methods and Conclusion sections.

```
# Data Cleaning

# Check for na values
gavote <- na.omit(gavote)

#Re-Encode rural and Atlanta
# rural
unique(gavote$rural)
```

```
## [1] rural urban
## Levels: rural urban
```

```
gavote$rural <- ifelse(gavote$rural == 'rural',1,0)

# Atlanta
unique(gavote$atlanta)
```

```
## [1] notAtlanta Atlanta
## Levels: Atlanta notAtlanta
```

```
gavote$atlanta <- ifelse(gavote$atlanta == 'Atlanta',1,0)
```

## Methods

We will be using two methods to examine this data. We will first use a basic Frequentist OLS regression model (lm() in R) along with forward-stepwise variable selections. This will allow us to perform regression diagnostics and make inferences on characteristics that could make counties prone to undercount.

We will then fit a Normal Bayesian model. Here, we will discuss our likelihood function, priors and posterior distributions. Since we do not have any information on priors we will use flat or non-informative priors.

## Exploratory Data Analysis

Let us now apply a more rigorous method to attempt to examine the factors that can predict the occurrence of undercount at the county level. Let us begin our exploratory analysis by examining a few rows of data:

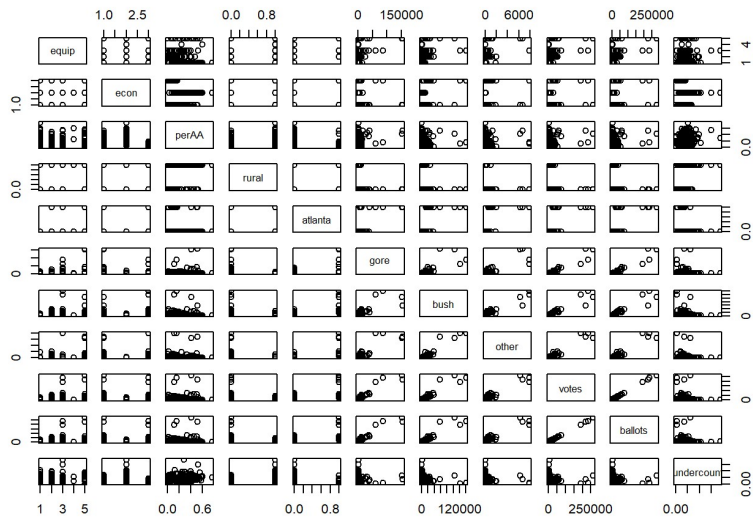
```
head(gavote)
```

	equip <fctr>	econ <fctr>	perAA <dbl>	rural <dbl>	atlanta <dbl>	gore <int>	bush <int>	other <int>	votes <int>
APPLING	LEVER	poor	0.182	1	0	2093	3940	66	6099
ATKINSON	LEVER	poor	0.230	1	0	821	1228	22	2071
BACON	LEVER	poor	0.131	1	0	956	2010	29	2995
BAKER	OS-CC	poor	0.476	1	0	893	615	11	1519
BALDWIN	LEVER	middle	0.359	1	0	5893	6041	192	12126
BANKS	LEVER	middle	0.024	1	0	1220	3202	111	4533

6 rows | 1-10 of 12 columns

Let us now look for any correlations between our variables.

```
pairs(gavote)
```



```
# Data Cleaning

# Check for na values
gavote <- na.omit(gavote)

#Re-Encode rural and Atlanta
# rural
unique(gavote$rural)
```

```
## [1] 1 0
```

```
gavote$rural <- ifelse(gavote$rural == 'rural',1,0)

# Atlanta
unique(gavote$atlanta)
```

```
## [1] 0 1
```

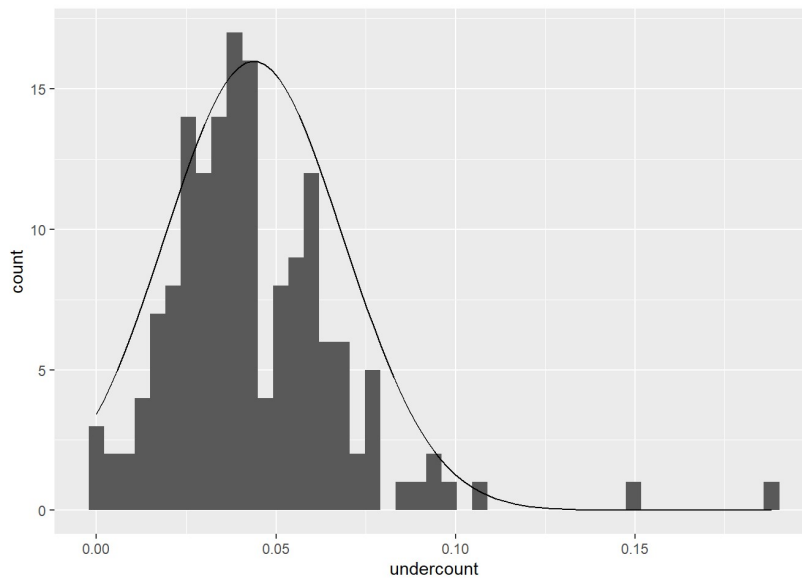
```
gavote$atlanta <- ifelse(gavote$atlanta == 'Atlanta',1,0)
```

We can see that:

1. Undercount is negatively correlated with the level of equipment.
2. Ballots and Votes are essentially co-linear.

Now let us examine our dependent variable. We want to see how undercount is distributed vs. a normal distribution.

```
ggplot(gavote, aes(x=undercount)) + geom_histogram(bins=45) + stat_function(fun = dnorm, args = list(mean = mean(gavote$undercount), sd=sd(gavote$undercount)))
```



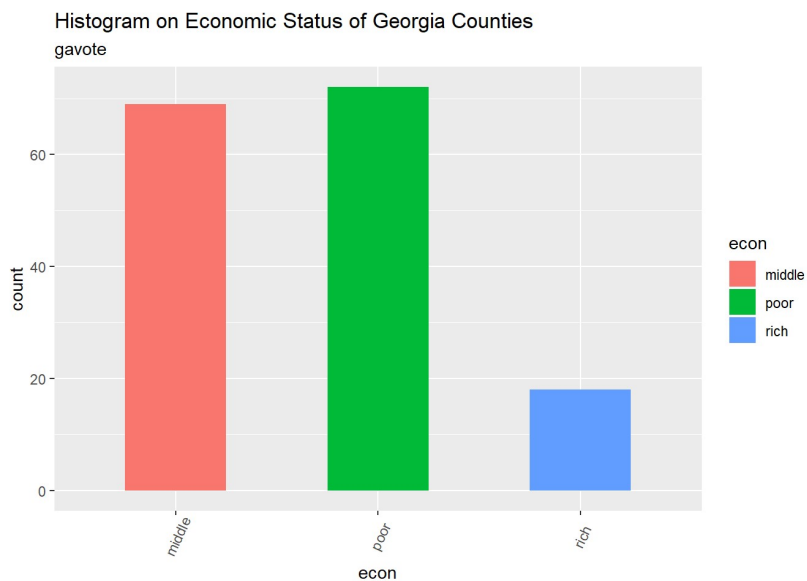
In order to make inference and run necessary analysis (confidence intervals etc.) prior to our Bayesian analysis we will want our dependent variable to be normally distributed. This is close enough (based on the histogram above). Our dependent variable is roughly normal with a few right side outliers.

```
shapiro.test(gavote$undercount)
```

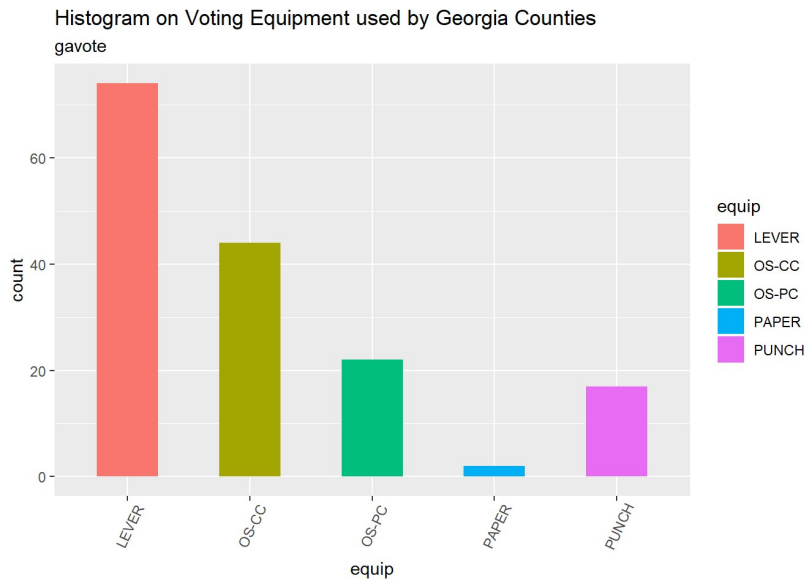
```
##
## Shapiro-Wilk normality test
##
## data:  gavote$undercount
## W = 0.88174, p-value = 6.176e-10
```

Though our sample does not pass the Shapiro-Wilk test this is acceptable as we can see from our histogram that it is roughly normally distributed with some right side outliers. The Shapiro-Wilk test often fails as N becomes large (in contrast to many other statistical tests). Normally, I may remove some of the outliers and re-run this analysis but due to our small data I am only going to remove outliers within the regression diagnostics (when they are shown to be high leverage points).

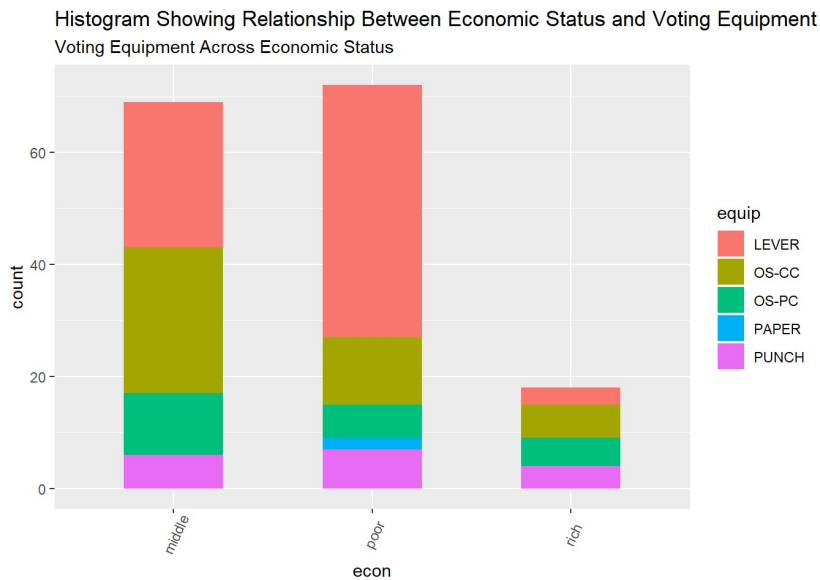
```
# Distribution of Categorical Variables
g <- ggplot(gavote, aes(econ))
g + geom_bar(aes(fill=econ), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Histogram on Economic Status of Georgia Counties",
       subtitle="gavote")
```



```
q <- ggplot(gavote, aes(equip))
q + geom_bar(aes(fill=equip), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Histogram on Voting Equipment used by Georgia Counties",
       subtitle="gavote")
```



```
# Histogram on a Categorical variable
r <- ggplot(gavote, aes(econ))
r + geom_bar(aes(fill=equip), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Histogram Showing Relationship Between Economic Status and Voting Equipment",
        subtitle="Voting Equipment Across Economic Status")
```



From these plots we can see that most counties in Georgia are poor and do not use the most reliable form of voting (paper). It will be interesting to examine whether or not either of these categorical variables end up being significant in a model. I would hypothesize, that they would be (due to knowledge of politics and the history of voting) but they could be statistically insignificant in predicting undercount. We can also see that there is a large increase in the use of lever machines for voting as economic status decreases. This, if lever is found to be an unreliable form of voting, could be used to make inference about which counties have the least accurate voting counts.

Let us now move onto building our models, both within the OLS and Bayesian Frameworks.

## Model Building

### Frequentist OLS Regression Model, Variable Selection and Diagnostics

Now, fit appropriate models for undercount as response variable with all possible predictors in the dataset as described in the "Final Project Instruction" document.

We can begin the Stepwise Variable Selection by first fitting a model including all of the variables.

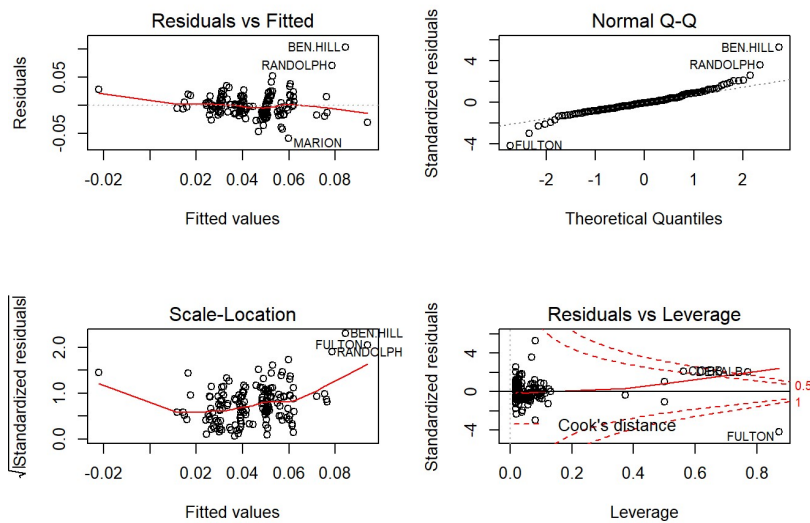
```
##
## Call:
## lm(formula = undercount ~ equip + econ + perAA + rural + atlanta +
##     gore + bush + other + votes + ballots, data = gavote)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.059313 -0.012135 -0.000988  0.008951  0.103068
##
## Coefficients: (3 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.111e-02  4.225e-03   7.365 1.17e-11 ***
## equipOS-CC   8.679e-03  4.185e-03   2.074  0.0398 *
## equipOS-PC   2.523e-02  5.482e-03   4.602 8.99e-06 ***
## equipPAPER  -1.033e-02  1.487e-02  -0.695  0.4884
## equipPUNCH   1.091e-02  6.135e-03   1.779  0.0774 .
## econpoor     1.941e-02  4.337e-03   4.474 1.53e-05 ***
## econrich    -1.325e-02  6.589e-03  -2.011  0.0462 *
## perAA        1.565e-03  1.371e-02   0.114  0.9092
## rural        NA         NA         NA      NA
## atlanta      NA         NA         NA      NA
## gore        -1.267e-05  3.107e-06  -4.079 7.40e-05 ***
## bush        -1.225e-05  2.892e-06  -4.236 3.99e-05 ***
## other        3.355e-06  5.948e-06   0.564  0.5736
## votes        NA         NA         NA      NA
## ballots     1.153e-05  2.769e-06   4.164 5.31e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0206 on 147 degrees of freedom
## Multiple R-squared:  0.3667, Adjusted R-squared:  0.3193
## F-statistic: 7.738 on 11 and 147 DF,  p-value: 1.799e-10
```

As we can see not nearly all of our predictors are significant. Now, let us remove insignificant variables (Performing Stepwise Variable Reduction) resulting in the following model:

```
##
## Call:
## lm(formula = undercount ~ equip + econ + gore + bush + ballots,
##     data = gavote)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.058880 -0.011935 -0.001180  0.008808  0.103648
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.092e-02  3.363e-03   9.192 3.13e-16 ***
## equipOS-CC   8.401e-03  4.095e-03   2.052  0.0420 *
## equipOS-PC   2.472e-02  5.378e-03   4.598 9.05e-06 ***
## equipPAPER  -1.009e-02  1.474e-02  -0.685  0.4947
## equipPUNCH   1.049e-02  6.039e-03   1.737  0.0845 .
## econpoor     1.989e-02  3.689e-03   5.392 2.67e-07 ***
## econrich    -1.300e-02  6.432e-03  -2.020  0.0451 *
## gore        -1.223e-05  2.952e-06  -4.141 5.76e-05 ***
## bush        -1.176e-05  2.722e-06  -4.322 2.82e-05 ***
## ballots     1.119e-05  2.662e-06   4.205 4.48e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02048 on 149 degrees of freedom
## Multiple R-squared:  0.3653, Adjusted R-squared:  0.327
## F-statistic: 9.53 on 9 and 149 DF,  p-value: 2.123e-11
```

```
## Warning in abline(stepwise_lm_2): only using the first two of 10 regression
## coefficients
```

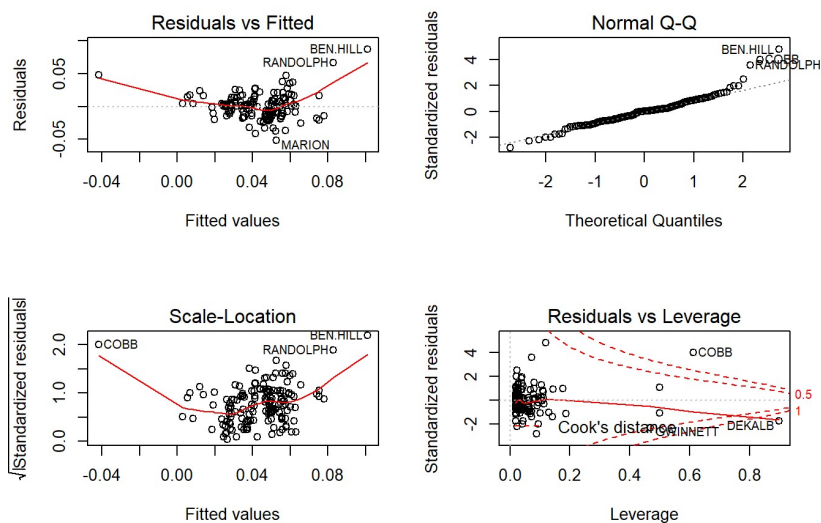




We can now perform Regression Diagnostics. First, we can decide on outliers. I am going to use a standard Rule of Thumb: I am going to only remove outliers that have a very high leverage (greater than Cooks Distance). I attempted to remove more and it caused many errors within the model due to the act of decreasing the dimensions of already small data. I will remove FULTON.

We can now recreate the same model:

```
##
## Call:
## lm(formula = undercount ~ equip + econ + gore + bush + ballots,
##     data = gavote)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.051703 -0.012471  0.000477  0.009195  0.087094
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.898e-02  3.200e-03   9.058 7.24e-16 ***
## equipOS-CC    5.807e-03  3.903e-03   1.488  0.1389
## equipOS-PC    2.274e-02  5.088e-03   4.469 1.55e-05 ***
## equipPAPER   -9.044e-03  1.390e-02  -0.651  0.5161
## equipPUNCH    6.083e-03  5.777e-03   1.053  0.2941
## econpoor     2.090e-02  3.484e-03   5.998 1.47e-08 ***
## econrich    -1.894e-02  6.208e-03  -3.051  0.0027 **
## gore        -3.554e-05  5.940e-06  -5.983 1.58e-08 ***
## bush        -3.386e-05  5.597e-06  -6.050 1.13e-08 ***
## ballots      3.282e-05  5.477e-06   5.993 1.50e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0193 on 148 degrees of freedom
## Multiple R-squared:  0.4378, Adjusted R-squared:  0.4036
## F-statistic: 12.81 on 9 and 148 DF,  p-value: 6.066e-15
```



Now we can perform regression diagnostics.

1. Residuals vs. Fitted -Our relationship (even after removing FULTON) is not very linear -This is shown by the curved line -This lowers our confidence in this relationship being of the linear classification
2. Normal Q-Q -Our residuals are roughly normally distributed with a few outliers -This is a positive sign in linearity for our model
3. Scale-Location -Our data is very heteroskedastic (difference in variance across the residuals) -This means our variance is very inconsistent across our data
4. Residuals vs. Leverage -Few points (after FULTON removal) lie outside of Cook's Distance

Overall, the predictors in this model are statistically significant. However, many of our regression diagnostics do not pass so the linearity of this relationship is in question. This is expected, as many situations within the social sciences (in which politics is included) are not very linear in their relationship.

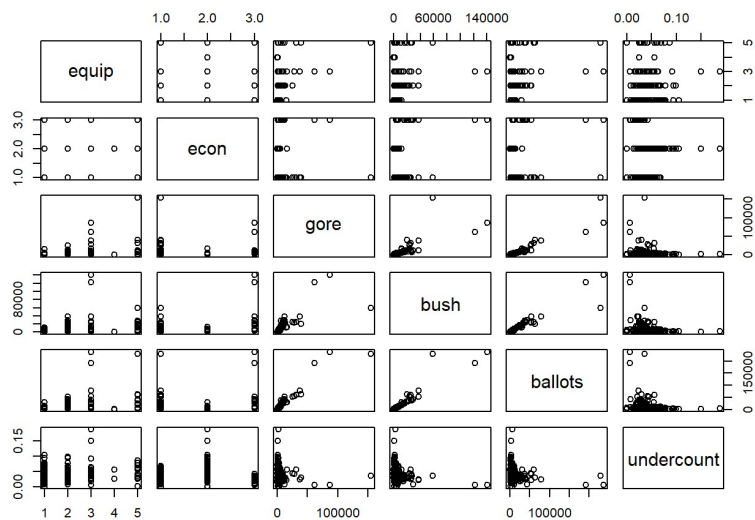
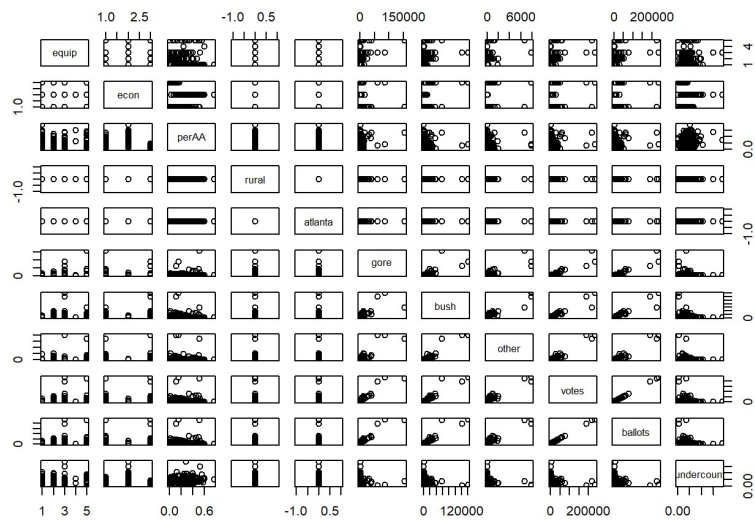
```
## Prediction equation coefficient
bHat <- coef(stepwise_lm_2)
bHat
```

```
## (Intercept)  equipOS-CC  equipOS-PC  equipPAPER  equipPUNCH
## 2.898298e-02  5.807496e-03  2.273752e-02 -9.044148e-03  6.082960e-03
## econpoor    econrich    gore        bush        ballots
## 2.089696e-02 -1.894076e-02 -3.554025e-05 -3.386005e-05  3.282120e-05
```

Thus our regression equation is:

$$\begin{aligned} \text{UNDERCOUNT} = & 5.807496 * 10^{-3}x_1 + 2.273752 * 10^{-2}x_2 - 9.044148 * 10^{-3}x_3 + 6.082960 * 10^{-3}x_4 + 2.089696 * 10^{-2}x_5 - 1.894076 * 10^{-2}x_6 \\ & - 3.554025 * 10^{-5}x_7 - 3.386005 * 10^{-5}x_8 + 3.282120 * 10^{-5}x_9 + 2.898298 * 10^{-2} + \epsilon_i \\ \text{UNDERCOUNT} = & \beta_1 + \text{EQUIP}x_1 + \text{ECON}x_2 + \text{GORE}x_3 + \text{BUSH}x_4 + \text{BALLOTS}x_5 + \epsilon_i \end{aligned}$$

The correlation between only our significant variables is:



We can see that undercount is correlated with the class 'poor' in econ but, suprisingly, not clearly correlated with any single type of election equipment.

Now let us create specific prediction probabilities to model likely scenarios. This code chunk is shown so the reader can clearly see how we developed these predictions.

```
## Prediction equations
bHat1=coef(stepwise_lm_2)
bHat1
```

```
## (Intercept) equipOS-CC equipOS-PC equipPAPER equipPUNCH
## 2.898298e-02 5.807496e-03 2.273752e-02 -9.044148e-03 6.082960e-03
## econpoor econrich gore bush ballots
## 2.089696e-02 -1.894076e-02 -3.554025e-05 -3.386005e-05 3.282120e-05
```

```
# Prediction Equation for: a poor county, with 3000 votes for Gore, 2000 votes for Bush, lever voting (0 for all)
and 5500 ballots
y1=0+2.898298e-02+2.089696e-02+(3000*-3.554025e-05) +(2000*-3.386005e-05)+(5500*3.282120e-05)
print(paste("There is a predicted value",y1*100,"% of votes to be undercounted by in this county"))
```

```
## [1] "There is a predicted value 5.605569 % of votes to be undercounted by in this county"
```

```
# Prediction Equation for: a poor county, with 2000 votes for Gore, 3000 votes for Bush, lever voting (0 for all)
and 5500 ballots
y2=0+2.898298e-02+2.089696e-02+(2000*-3.554025e-05) +(3000*-3.386005e-05)+(5500*3.282120e-05)
print(paste("There is a predicted value",y2*100,"% of votes to be undercounted by in this county"))
```

```
## [1] "There is a predicted value 5.773589 % of votes to be undercounted by in this county"
```

```
# Prediction Equation for: a RICH county, with 2000 votes for Gore, 3000 votes for Bush, lever voting (0 for all)
and 5500 ballots
y3=0+2.898298e-02+-1.894076e-02+(3000*-3.554025e-05) +(2000*-3.386005e-05)+(5500*3.282120e-05)
print(paste("There is a predicted value",y3*100,"% of votes to be undercounted by in this county"))
```

```
## [1] "There is a predicted value 1.621797 % of votes to be undercounted by in this county"
```

We can now see, for example, that the predicted percent of votes to be undercounted drops significantly in rich vs. poor economic status counties when other variables are held fixed. There does not appear to be a drastic change in undercount when the winning candidate is changed.

Overall, the linear model worked relatively well. We were able to isolate a decent number of significant predictors and can run various inputs to predict the demographics of counties that will experience undercount in other elections.

## Normal Bayesian Model

We will now create our normal Bayesian Model.

First we can examine our Frequentist Model:

```
##### OLS fit (Frequentist Inference)
summary(stepwise_lm_2)
```

```
##
## Call:
## lm(formula = undercount ~ equip + econ + gore + bush + ballots,
##     data = gavote)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.051703 -0.012471  0.000477  0.009195  0.087094
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.898e-02  3.200e-03   9.058 7.24e-16 ***
## equipOS-CC   5.807e-03  3.903e-03   1.488  0.1389
## equipOS-PC   2.274e-02  5.088e-03   4.469 1.55e-05 ***
## equipPAPER  -9.044e-03  1.390e-02  -0.651  0.5161
## equipPUNCH   6.083e-03  5.777e-03   1.053  0.2941
## econpoor     2.090e-02  3.484e-03   5.998 1.47e-08 ***
## econrich    -1.894e-02  6.208e-03  -3.051  0.0027 **
## gore        -3.554e-05  5.940e-06  -5.983 1.58e-08 ***
## bush        -3.386e-05  5.597e-06  -6.050 1.13e-08 ***
## ballots     3.282e-05  5.477e-06   5.993 1.50e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0193 on 148 degrees of freedom
## Multiple R-squared:  0.4378, Adjusted R-squared:  0.4036
## F-statistic: 12.81 on 9 and 148 DF,  p-value: 6.066e-15
```

```
## Coefficients of Prediction equations
bHat= coef(stepwise_lm_2)
bHat
```

```
## (Intercept)   equipOS-CC   equipOS-PC   equipPAPER   equipPUNCH
## 2.898298e-02  5.807496e-03  2.273752e-02 -9.044148e-03  6.082960e-03
## econpoor      econrich      gore          bush          ballots
## 2.089696e-02 -1.894076e-02 -3.554025e-05 -3.386005e-05  3.282120e-05
```

I then initialized various functions and packages in order to prep the data to be used to create a Bayesian Normal Model.

Now we can initialize the gibbsMLR function in order to implement a Bayesian ML Regression.

```

### Gibbs sampler for a Multiple Linear Regression Model (MLR) ###
##### Bayesian Inference #####
## y (nx1) response ###
### X (nxp) the covariate matrix for effects ###
### nIter, the number of samples to be collected, set by default to 10,000.
## Hyper-parameters: df0 and S0 (the scale and degree of freedom for the scaled-inverse chi-square prior)
##                               : b0 and varB (the mean and variance of the normal prior assigned to effects).

gibbsMLR=function(y,X,nIter=10000,df0=4,S0=var(y)*0.8*(df0-2),b0=0,varB=1e12,verbose=500){

  ## Objects to store samples
  p=ncol(X); n=nrow(X)
  B=matrix(nrow=nIter,ncol=p,0) # create a matrix to store the gibbs sample for beta
  varE=rep(NA,nIter)           # .. for error variance

  ## Initialize
  B[1,]=0                      # initial values for slopes
  B[1,1]=mean(y)               # initial value for y-intercept
  b=B[1,]
  varE[1]=var(y)               # initial error variance
  resid=y-B[1,1]               # centered y (orthogonal)

  ## Computing sum x'x for each column
  SSx=colSums(X^2)

  for(i in 2:nIter){
    # Sampling regression coefficients
    for(j in 1:p){
      A=SSx[j]/varE[i-1]+1/varB
      Xy= sum(X[,j]*resid)+SSx[j]*b[j] # equivalent to X[,j]'(y-X[, -j]*%*%b[-j])
      rhs=Xy/varE[i-1] + b0/varB      # Numerator of beta^tilda_k
      condMean=rhs/A
      condVar=1/A
      b_old=b[j]
      b[j]=rnorm(n=1,mean=condMean,sd=sqrt(condVar))
      B[i,j]=b[j]
      resid=resid-X[,j]*(b[j]-b_old) # updating residuals
    }
    # Sampling the error variance
    RSS=sum(resid^2)
    DF=n+df0
    S=RSS+S0
    varE[i]=S/rchisq(df=DF,n=1)

    ## if(i%%verbose==0){ cat(' Iter ', i, '\n') }
  }

  out=list(effects=B,varE=varE)
  return(out)
}

```

Before running this let us discuss what a Bayesian MLR is truly doing (behind the scenes of this function). Our goal is predict the undercount at the county level. To do this we have been using a MLR. From a Bayesian perspective, our goal is to get a posterior distribution of the possible model parameters based on the data and prior knowledge (from either a field expert or educated guess).

Our MLR Model can be written as:

$$y_i = \sum_{j=1}^p x_{ij}\beta_j + \varepsilon_i$$

for

$$i = 1, \dots, n$$

This can be expressed in matrix form as:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

which can also be written as:

$$\mathbf{y} = \boldsymbol{\beta}^T \mathbf{X} + \boldsymbol{\varepsilon}$$

where

$$\boldsymbol{\beta}^T = (\beta_1 \cdots \beta_p)$$

is the transpose of the vector.

$$\boldsymbol{\beta} = (\beta_1 \cdots \beta_p)^T$$

The goal of OLS is to minimize:

$$S(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

From all of this we can obtain the following estimate that:

$$\hat{\beta}_{OLS} = (X^T X)^{-1} (X^T y)$$

All of this, comes with the normal assumption that

$$\varepsilon_1, \dots, \varepsilon_n \sim N(0, \sigma^2)$$

This can also be expressed in matrix form:

$$\varepsilon \sim \text{Multivariate } N(\mathbf{0}_{n \times 1}, \sigma_\varepsilon^2 \mathbf{I}_{n \times n})$$

which implies:

$$\{y|X, \beta, \sigma^2\} \sim \text{Multivariate } N(X\beta, \sigma^2 I)$$

The posterior distribution (our beliefs after viewing data) can be defined as:

$$p(\theta|y) \propto p(\theta) * p(y|\theta)$$

(y = response data)

Within our model, since we do not have any expert opinion on priors, we are assuming a uniform/flat prior. This means:

$$p(\beta_1, \dots, \beta_p) \propto 1$$

We assume our data comes from a normal population thus:

$$\epsilon_i \sim N(0, \sigma^2)$$

and thus:

$$y_i \sim N(\beta_1 + \beta_2 x_i + \dots + \beta_n x_i)$$

From this we can show that our likelihood function is:

$$p(y_1, \dots, y_n | \beta_1, \dots, \beta_p) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - \beta_1 - \beta_2 x_i - \dots - \beta_p x_i)^2\right\}$$

This can be derived and written in matrix form as:

$$\begin{aligned} p(y_1, y_2, \dots, y_n | \beta, \sigma_\varepsilon^2) &= p(y_1 | \beta, \sigma_\varepsilon^2) \times \dots \times p(y_n | \beta, \sigma_\varepsilon^2) \\ &= \prod_{i=1}^n (2\pi\sigma_\varepsilon^2)^{-\frac{1}{2}} e^{-\frac{(y_i - \sum_{j=1}^p x_{ij}\beta_j)^2}{2\sigma_\varepsilon^2}} \\ &= (2\pi\sigma_\varepsilon^2)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma_\varepsilon^2} \sum_{i=1}^n (y_i - \sum_{j=1}^p x_{ij}\beta_j)^2} \\ &= (2\pi\sigma_\varepsilon^2)^{-\frac{n}{2}} e^{-\frac{1}{2\sigma_\varepsilon^2} (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)} \end{aligned}$$

From this we can see that:

$$\text{RSS}(\mathbf{y}, \mathbf{X}, \beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$$

We can then derive the Maximum Likelihood Estimates (MLE) for  $(\beta, \sigma_\varepsilon^2)$  by minimizing the negative log-likelihood:

$$\{\beta, \sigma_\varepsilon^2\} \stackrel{\text{m}}{=} \frac{n}{2} \log(\sigma_\varepsilon^2) + \frac{1}{2\sigma_\varepsilon^2} \text{RSS}(\mathbf{y}, \mathbf{X}, \beta)$$

Which shows:

$$\begin{aligned} \text{MLE}(\beta) &= \hat{\beta}_{OLS} \\ \text{MLE}(\sigma_\varepsilon^2) &= \hat{\sigma}_\varepsilon^2 = \frac{\text{RSS}(\mathbf{y}, \mathbf{X}, \hat{\beta}_{OLS})}{n} \end{aligned}$$

and that

$$\{\beta, \sigma_\varepsilon^2\} \stackrel{\text{m}}{=} \frac{n}{2} \log(\sigma_\varepsilon^2) + \frac{1}{2\sigma_\varepsilon^2} \text{RSS}(\mathbf{y}, \mathbf{X}, \beta)$$

Commonly people use IID normal priors for  $\beta$ , but, since we have no prior knowledge we will be using uninformed / flat priors.

From this we can draw the conclusion that, assuming a flat prior as mentioned above, our resulting posterior will be:

$$p(\beta_1, \dots, \beta_n | y_1, \dots, y_n) \propto 1 \times \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y_i - \beta_1 - \beta_2 x_i - \dots - \beta_p x_i)^2\right\}$$

which:

$$e^{-\frac{1}{2\sigma^2} \sum (y_i - \beta - \beta_2 x_i - \dots - \beta_p x_i)^2}$$

Specifically, for our regression function our:

Likelihood function:

Likelihood Function of Undercount (Y):

$$p(\text{UNDERCOUNT (Y)}|\text{EQUIP,ECON,GORE,BUSH,BALLOTS}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(\text{UNDERCOUNT}_i - \beta_1 - \text{EQUIP}_2x_i - \text{ECON}_3x_i - \text{BUSH}_5x_i - \text{BALLOTS}_6x_i)^2\right\}$$

Prior:

Flat Prior:

$$p(\beta_1, \dots, \beta_p) \propto 1$$

Posterior Distributions:

Posterior Distribution of Regression Coefficients:

$$p(\text{EQUIP,ECON,GORE,BUSH,BALLOTS}|\text{UNDERCOUNT (Y)}) \propto 1 \times \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(\text{UNDERCOUNT}_i - \beta_1 - \text{EQUIP}_2x_i - \text{ECON}_3x_i - \text{BUSH}_5x_i - \text{BALLOTS}_6x_i)^2\right\}$$

In the generic formed a posterior is defined in the form of:

$$p(\theta|y) \propto p(\theta) * p(y|\theta)$$

We can now move on to working with the data from a Bayesian perspective using R.

Now we can encode variables used in Bayesian MLR function:

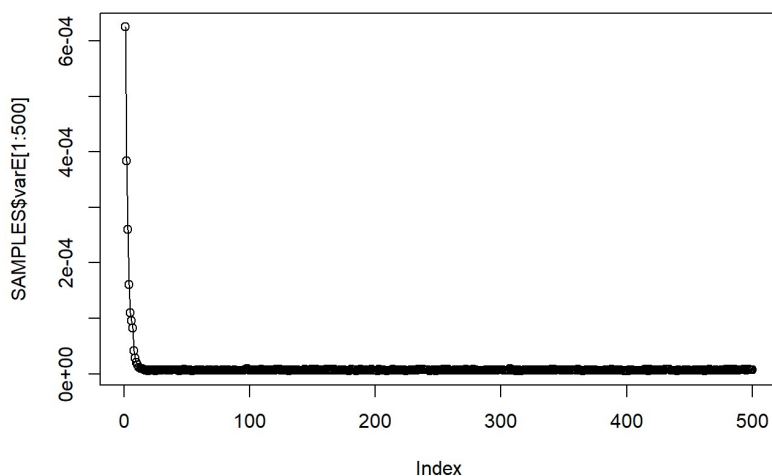
```
undercount=gavote$undercount
X=model.matrix(~.,data=gavote1)
SAMPLES=gibbsMLR(y=undercount,X=X,nIter=15000)
#dim(SAMPLES$effects)# cols are effects, rows are samples
#head(SAMPLES$varE)
```

The GibbsMLR function is now run and we have created 15,000 samples using this function.

We can now conduct post-gibbs analysis (trace plot, auto-correlation, decide on burn-in and thinning, provide posterior means, posterior SDs and posterior credibility regions, estimate and report MC error).

```
# trace plots of a few parameters....

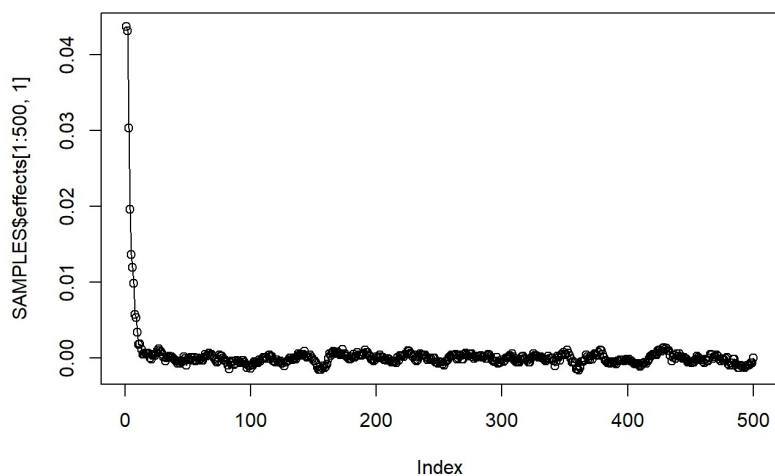
# Trace plot of all variables
plot(SAMPLES$varE[1:500],type='o')
```



```
print("1st 10 iterations should be discared as burn in")
```

```
## [1] "1st 10 iterations should be discared as burn in"
```

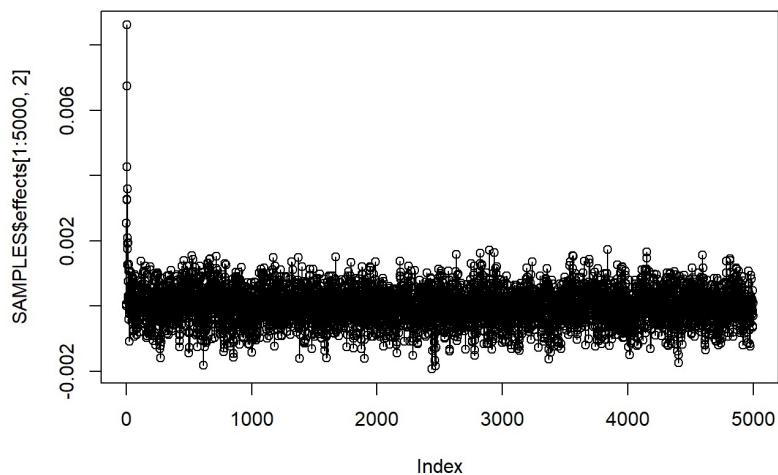
```
# Intercept trace plot
plot(SAMPLES$effects[1:500,1],type='o')
```



```
print("Intercept, first 50 need to be discarded")
```

```
## [1] "Intercept, first 50 need to be discarded"
```

```
# The plot above allows us to decide on burn-in quantity, since the data centers around index = 500 I will burn-n  
(remove) the first 500 samples  
plot(SAMPLES$effects[1:5000,2],type='o') # effect of coefficients Discard 500
```



```
print("The plot above allows us to decide on burn-in quantity, since the data centers around index = 500 I will bu  
rn-n (remove) the first 500 samples")
```

```
## [1] "The plot above allows us to decide on burn-in quantity, since the data centers around index = 500 I will b  
urn-n (remove) the first 500 samples"
```

```
## Discarding 500 iterations as burn-in  
B=SAMPLES$effects[-(1:500),];colnames(B)=colnames(X)  
varE=SAMPLES$varE[-(1:500)]
```

```
# Discarding Burn - In Summary  
summary(B)
```



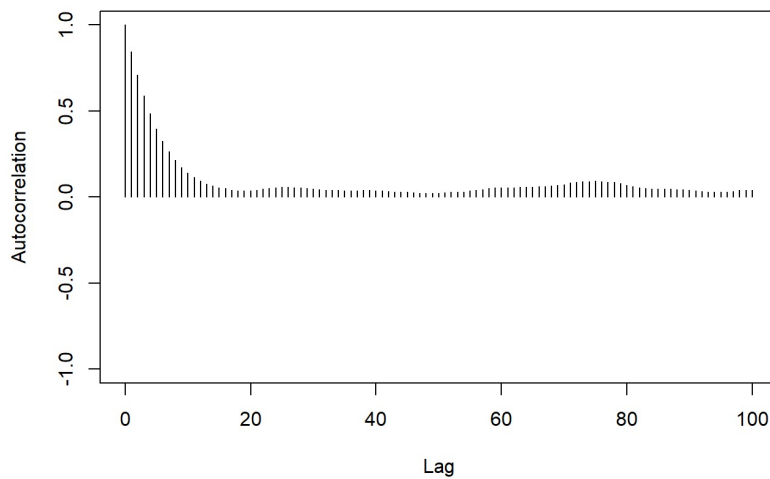
```
##      (Intercept)      equipOS-CC      equipOS-PC
## Min.   :-1.903e-03   Min.    :-2.016e-03   Min.    :-2.677e-03
## 1st Qu.: -2.694e-04   1st Qu.: -3.929e-04   1st Qu.: -4.349e-04
## Median :  9.237e-05   Median : -4.572e-05   Median :  5.436e-05
## Mean   :  8.683e-05   Mean     : -4.440e-05   Mean     :  5.815e-05
## 3rd Qu.:  4.401e-04   3rd Qu.:  3.074e-04   3rd Qu.:  5.434e-04
## Max.   :  2.050e-03   Max.     :  1.965e-03   Max.     :  3.412e-03
##      equipPAPER      equipPUNCH      econpoor
## Min.   :-6.723e-03   Min.    :-3.274e-03   Min.    :-1.995e-03
## 1st Qu.: -1.240e-03   1st Qu.: -6.031e-04   1st Qu.: -2.597e-04
## Median :  2.040e-05   Median : -8.250e-05   Median :  8.227e-05
## Mean   :  3.699e-06   Mean     : -7.926e-05   Mean     :  8.088e-05
## 3rd Qu.:  1.259e-03   3rd Qu.:  4.426e-04   3rd Qu.:  4.250e-04
## Max.   :  6.670e-03   Max.     :  3.044e-03   Max.     :  1.976e-03
##      econrich      gore      bush
## Min.   :-0.0036333   Min.    :-1.847e-06   Min.    :-1.790e-06
## 1st Qu.: -0.0007433   1st Qu.: -1.042e-06   1st Qu.: -9.930e-07
## Median : -0.0001786   Median : -6.074e-07   Median : -5.740e-07
## Mean   : -0.0001819   Mean     : -6.646e-07   Mean     : -6.266e-07
## 3rd Qu.:  0.0003691   3rd Qu.: -2.696e-07   3rd Qu.: -2.529e-07
## Max.   :  0.0030080   Max.     :  3.551e-07   Max.     :  3.377e-07
##      ballots      undercount
## Min.   :-3.124e-07   Min.     :0.9522
## 1st Qu.:  2.500e-07   1st Qu.:0.9894
## Median :  5.614e-07   Median :0.9964
## Mean   :  6.126e-07   Mean     :0.9963
## 3rd Qu.:  9.681e-07   3rd Qu.:1.0033
## Max.   :  1.697e-06   Max.     :1.0350
```

Let's now convert B and varE to mcmc objects in order to examine autocorrelation using the coda library. This will allow us to run more analysis.

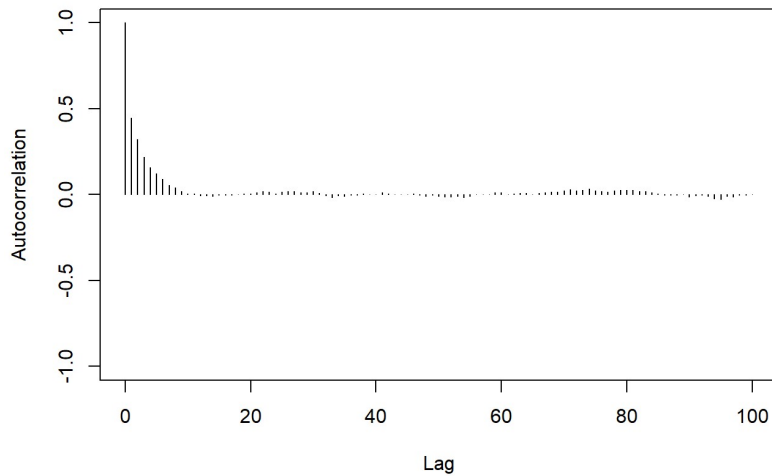
```
# Converting to MCMC object
B=as.mcmc(B)
varE=as.mcmc(varE)
```

Now that B and varE are mcmc objects we can create autocorrelation plots.

```
par(mfrow=c(1,2))
autocorr.plot(B[,1], lag.max=100)
```



```
autocorr.plot(B[,2], lag.max=100)
```



In both instances the autocorrelation approaches 0 as Lag increases to 20. Thus we should sample 1 value every 20 indexes to control for autocorrelation. Thus, we should resample with value 20, 40, ..., to n hundred (end of our data set). This process is called thinning our data.

We have now completed both burn-in and thinning of our data prerequisites for Bayesian work). Let us now create a HPD interval for our B mcmc object. Since this is an MCMC object I read in an additional formula from the cran GitHub for coda:

```
HPDInterval <- function(obj, prob = 0.95, ...) UseMethod("HPDInterval")

HPDInterval.mcmc <- function(obj, prob = 0.95, ...)
{
  obj <- as.matrix(obj)
  vals <- apply(obj, 2, sort)
  if (!is.matrix(vals)) stop("obj must have nsamp > 1")
  nsamp <- nrow(vals)
  npar <- ncol(vals)
  gap <- max(1, min(nsamp - 1, round(nsamp * prob)))
  init <- 1:(nsamp - gap)
  inds <- apply(vals[init + gap, ,drop=FALSE] - vals[init, ,drop=FALSE],
               2, which.min)
  ans <- cbind(vals[cbind(inds, 1:npar)],
              vals[cbind(inds + gap, 1:npar)])
  dimnames(ans) <- list(colnames(obj), c("lower", "upper"))
  attr(ans, "Probability") <- gap/nsamp
  ans
}

HPDInterval.mcmc.list <- function(obj, prob = 0.95, ...)
  lapply(obj, HPDInterval, prob)
# source: https://github.com/cran/coda/blob/master/R/HPDInterval.R
```

```
print("The below shows the posterior mean and standard deviation")
```

```
## [1] "The below shows the posterior mean and standard deviation"
```

```
summary(B)
```

```
##
## Iterations = 1:14500
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 14500
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean          SD Naive SE Time-series SE
## (Intercept)  8.683e-05 5.286e-04 4.390e-06      1.441e-05
## equipOS-CC   -4.440e-05 5.286e-04 4.390e-06      8.915e-06
## equipOS-PC   5.815e-05 7.280e-04 6.046e-06      1.246e-05
## equipPAPER   3.699e-06 1.858e-03 1.543e-05      1.651e-05
## equipPUNCH  -7.926e-05 7.731e-04 6.421e-06      9.907e-06
## econpoor     8.088e-05 5.149e-04 4.276e-06      9.954e-06
## econrich    -1.819e-04 8.315e-04 6.905e-06      1.621e-05
## gore        -6.646e-07 4.990e-07 4.144e-09      3.281e-07
## bush        -6.266e-07 4.727e-07 3.925e-09      3.115e-07
## ballots     6.126e-07 4.603e-07 3.823e-09      3.356e-07
## undercount  9.963e-01 1.046e-02 8.688e-05      2.972e-04
##
## 2. Quantiles for each variable:
##
##              2.5%        25%         50%         75%        97.5%
## (Intercept) -9.463e-04 -2.694e-04  9.237e-05  4.401e-04  1.127e-03
## equipOS-CC  -1.090e-03 -3.929e-04 -4.572e-05  3.074e-04  9.995e-04
## equipOS-PC  -1.351e-03 -4.348e-04  5.436e-05  5.434e-04  1.506e-03
## equipPAPER  -3.691e-03 -1.240e-03  2.040e-05  1.259e-03  3.588e-03
## equipPUNCH  -1.589e-03 -6.031e-04 -8.250e-05  4.426e-04  1.429e-03
## econpoor    -9.309e-04 -2.597e-04  8.227e-05  4.250e-04  1.085e-03
## econrich    -1.802e-03 -7.433e-04 -1.786e-04  3.691e-04  1.453e-03
## gore        -1.636e-06 -1.042e-06 -6.074e-07 -2.696e-07  1.301e-07
## bush        -1.566e-06 -9.930e-07 -5.740e-07 -2.529e-07  1.271e-07
## ballots     -1.175e-07  2.500e-07  5.614e-07  9.681e-07  1.518e-06
## undercount  9.753e-01  9.894e-01  9.964e-01  1.003e+00  1.016e+00
```

```
summary(varE)
```

```
##
## Iterations = 1:14500
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 14500
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean          SD          Naive SE Time-series SE
##              6.707e-06      7.924e-07      6.581e-09      7.246e-09
##
## 2. Quantiles for each variable:
##
##              2.5%        25%         50%         75%        97.5%
##              5.355e-06  6.147e-06  6.641e-06  7.195e-06  8.474e-06
```

Now that we have derived our posterior let's sample again using Gibbs and plot the posterior distribution of undercount.

```
##### Gibbs Sampler for Normal Model #####

#Outputs the plot of Posterior

mu=mean(gavote$undercount)
y=gavote$undercount

## Hyper-parameters (i.e., parameters of the prior distribution(s))
mu_0 =1
sigma_0 =1

a =1
b = 1
lambd_init = 1

## Gibbs Sampler

# Initialization

ybar=mean(y)
# input <- list(y,mu_0,sigma_0, a, b, lambd_init)
# Sampling function

gibbs_N <- function(input, S, seed){
  set.seed(seed)
  ybar <- mean(input$y)
  n <- length(input$y)
  paral <- matrix(NA,S, 2)
  lambd <- input$lambd_init

  for(i in 1:S){
    mu1 = (input$mu_0/input$sigma_0^2+n*lambd*ybar)/(1/input$sigma_0^2+n*lambd)
    sigma1 =sqrt(1/(1/input$sigma_0^2)+n*lambd)
    mu <-rnorm(1,mean=mu1,sd=sigma1)
    a1 <-input$a +n/2
    b1 <- input$b +sum(input$y - mu)^2/2
    lambd <- rgamma(1, shape = a1, rate = b1)
    paral[i,] <- c(mu,lambd)
  }
  print(paral)
}

input <- list(y=y,mu_0=5,sigma_0 = 1, a = 1, b =1, lambd_init =1)
output<- gibbs_N(input, S=1000, seed = 10212019)
```

```
##      [,1]      [,2]
## [1,] 0.4512597 0.0468786023
## [2,] 2.6669157 0.0010683432
## [3,] 3.6136967 0.0004177096
## [4,] 3.9826598 0.0004123258
## [5,] 4.7908745 0.0002909764
## [6,] 4.7431181 0.0003253758
## [7,] 3.4272030 0.0005447643
## [8,] 3.8380829 0.0003894240
## [9,] 5.5322344 0.0001841151
## [10,] 3.7863764 0.0004411474
## [11,] 3.9972724 0.0004566614
## [12,] 3.1554536 0.0006545216
## [13,] 2.6345901 0.0010280299
## [14,] 5.4996245 0.0002410190
## [15,] 5.0796378 0.0002772579
## [16,] 5.6110678 0.0001646064
## [17,] 4.7983060 0.0002623779
## [18,] 6.7252341 0.0001426539
## [19,] 4.5706677 0.0002899769
## [20,] 4.5796297 0.0003324454
## [21,] 5.8169907 0.0001968019
## [22,] 4.6490511 0.0003848511
## [23,] 6.9308078 0.0001153541
## [24,] 5.1344595 0.0003039664
## [25,] 2.5559991 0.0009358596
## [26,] 4.6500285 0.0003595413
## [27,] 4.7259823 0.0002983712
## [28,] 4.7567237 0.0002521750
## [29,] 3.4821481 0.0006206917
## [30,] 4.7172178 0.0002781374
## [31,] 5.6569191 0.0002291093
## [32,] 5.4325466 0.0001946961
## [33,] 4.6299738 0.0002912349
## [34,] 4.5656256 0.0002736231
## [35,] 3.3853398 0.0006248538
## [36,] 2.1699609 0.0013556837
## [37,] 3.4503823 0.0005777598
## [38,] 6.0433395 0.0001702614
## [39,] 3.5255460 0.0006240557
## [40,] 5.1241905 0.0001945065
## [41,] 5.3681798 0.0002137444
## [42,] 5.0201847 0.0002251114
## [43,] 6.3554964 0.0001697460
## [44,] 5.5524466 0.0002186460
## [45,] 4.1739980 0.0004480736
## [46,] 5.7031570 0.0002091703
## [47,] 4.2257569 0.0004382143
## [48,] 4.3436285 0.0002967667
## [49,] 5.7837323 0.0001707944
## [50,] 3.9181494 0.0003695859
## [51,] 6.0627712 0.0002014398
## [52,] 5.1166959 0.0002562167
## [53,] 6.2491942 0.0001524043
## [54,] 4.9992226 0.0002333432
## [55,] 4.2398012 0.0003369495
## [56,] 3.2603361 0.0005778155
## [57,] 3.8236789 0.0004093078
## [58,] 4.6545544 0.0003255146
## [59,] 6.3418639 0.0001688140
## [60,] 5.8692940 0.0001950264
## [61,] 3.9347353 0.0004066703
## [62,] 4.7121825 0.0003081240
## [63,] 4.6048482 0.0002174714
## [64,] 4.3095460 0.0003689553
## [65,] 4.0557819 0.0003720557
## [66,] 2.9813589 0.0007054167
## [67,] 3.8216030 0.0004536890
## [68,] 5.7810058 0.0001618106
## [69,] 4.7080937 0.0003448247
## [70,] 5.9430609 0.0001856881
## [71,] 4.8845760 0.0003188900
## [72,] 2.9435860 0.0007870900
## [73,] 3.8837817 0.0003946008
## [74,] 5.6675393 0.0001943272
## [75,] 5.4501043 0.0001875337
## [76,] 4.4709132 0.0003178100
## [77,] 5.9714725 0.0001782391
## [78,] 5.9645848 0.0001727611
## [79,] 7.2003121 0.0001087963
## [80,] 4.8238394 0.0002838241
## [81,] 4.9281566 0.0002756659
## [82,] 5.9641309 0.0002199846
```

```
## [83,] 4.2073096 0.0003851953
## [84,] 4.8336080 0.0002436896
## [85,] 4.5154317 0.0002989406
## [86,] 3.8904963 0.0003240842
## [87,] 4.8673651 0.0002636628
## [88,] 1.6580512 0.0024543416
## [89,] 2.6146902 0.0008465181
## [90,] 7.4544781 0.0001238913
## [91,] 3.9337455 0.0003873126
## [92,] 5.7882589 0.0001781339
## [93,] 5.1924337 0.0002488487
## [94,] 5.7192651 0.0002214497
## [95,] 4.9168818 0.0002526363
## [96,] 6.3470956 0.0001609197
## [97,] 5.3211163 0.0002168534
## [98,] 5.2443852 0.0002119721
## [99,] 6.2734485 0.0001829928
## [100,] 6.5354078 0.0001432009
## [101,] 5.2189129 0.0001945068
## [102,] 4.8467104 0.0003204228
## [103,] 2.5908221 0.0008888759
## [104,] 2.9665489 0.0006552323
## [105,] 4.1358050 0.0003498977
## [106,] 5.2146294 0.0002382189
## [107,] 4.0788986 0.0004501927
## [108,] 7.5288780 0.0001163040
## [109,] 4.4643172 0.0003185302
## [110,] 3.9728275 0.0004478042
## [111,] 3.6900299 0.0005047568
## [112,] 2.7622072 0.0010293328
## [113,] 3.8089658 0.0004930870
## [114,] 3.0918002 0.0007123100
## [115,] 4.0371667 0.0004509296
## [116,] 3.3799129 0.0006425662
## [117,] 5.0168606 0.0002353035
## [118,] 5.2331106 0.0002287770
## [119,] 5.7440217 0.0001953889
## [120,] 4.3863805 0.0003658518
## [121,] 3.7699444 0.0005556015
## [122,] 4.6071605 0.0003607404
## [123,] 5.2361425 0.0001970698
## [124,] 4.3016725 0.0003910398
## [125,] 4.8171991 0.0002675557
## [126,] 3.8828996 0.0004308014
## [127,] 4.2294892 0.0003612721
## [128,] 3.4794075 0.0004496363
## [129,] 4.8900512 0.0003173136
## [130,] 6.4726641 0.0001460540
## [131,] 4.8462629 0.0002969381
## [132,] 4.9121870 0.0002355972
## [133,] 3.3194524 0.0005820262
## [134,] 4.6876837 0.0003209143
## [135,] 4.6703632 0.0003121984
## [136,] 5.6341263 0.0002355021
## [137,] 4.7063704 0.0002577066
## [138,] 4.5769943 0.0002952620
## [139,] 3.7188335 0.0005312858
## [140,] 4.6511452 0.0002728585
## [141,] 3.5558473 0.0005251570
## [142,] 5.0315021 0.0002559599
## [143,] 3.5753896 0.0005027916
## [144,] 5.7874916 0.0001874141
## [145,] 6.1457094 0.0001328701
## [146,] 6.5871751 0.0001334126
## [147,] 4.5116339 0.0003752063
## [148,] 3.5371960 0.0006469762
## [149,] 5.6214131 0.0001950782
## [150,] 5.8045040 0.0002316237
## [151,] 4.9646363 0.0002711728
## [152,] 3.6211286 0.0005124181
## [153,] 4.1138280 0.0003377872
## [154,] 6.0058545 0.0001665220
## [155,] 4.4296357 0.0003707792
## [156,] 4.7092539 0.0002763941
## [157,] 7.5674155 0.0001200440
## [158,] 4.9640587 0.0002585705
## [159,] 4.3114546 0.0002814539
## [160,] 4.2041982 0.0003914858
## [161,] 3.7676545 0.0005055249
## [162,] 5.4985318 0.0002132875
## [163,] 3.8990721 0.0004132798
## [164,] 3.9876992 0.0004106438
## [165,] 5.8392055 0.0002078572
## [166,] 5.8811663 0.0001833265
```

```
## [167,] 3.7713370 0.0004111842
## [168,] 5.2072105 0.0002748106
## [169,] 2.6966228 0.0008461101
## [170,] 4.4575243 0.0002838554
## [171,] 5.2088258 0.0002675241
## [172,] 4.0489607 0.0003129673
## [173,] 4.3993053 0.0003533247
## [174,] 4.2160731 0.0004261782
## [175,] 3.4094724 0.0005408213
## [176,] 5.5533394 0.0001736420
## [177,] 4.7296082 0.0003012687
## [178,] 5.2877190 0.0002099861
## [179,] 5.3849526 0.0002406256
## [180,] 4.8333388 0.0002935926
## [181,] 4.5487354 0.0003178179
## [182,] 4.5458629 0.0002948441
## [183,] 4.8122950 0.0003138216
## [184,] 4.7533165 0.0002442915
## [185,] 4.9638080 0.0002634151
## [186,] 5.0168643 0.0002586137
## [187,] 7.1610877 0.0001031679
## [188,] 5.2525409 0.0002457170
## [189,] 4.9974869 0.0002322802
## [190,] 5.7573848 0.0001952117
## [191,] 5.3879745 0.0002220103
## [192,] 4.0065470 0.0003689079
## [193,] 5.2711855 0.0001651395
## [194,] 4.8168720 0.0002724329
## [195,] 5.1277508 0.0002331000
## [196,] 3.9062376 0.0004480124
## [197,] 4.6037345 0.0003331165
## [198,] 4.8440795 0.0002631489
## [199,] 3.6249117 0.0004724967
## [200,] 5.5560399 0.0002054231
## [201,] 6.1747885 0.0001450170
## [202,] 4.3673735 0.0003441679
## [203,] 5.4945023 0.0002075586
## [204,] 4.1166897 0.0004814969
## [205,] 6.1473722 0.0001880814
## [206,] 6.7576842 0.0001583248
## [207,] 5.4585775 0.0002206255
## [208,] 6.4680117 0.0001566210
## [209,] 4.3357500 0.0003638010
## [210,] 2.4665169 0.0009544878
## [211,] 4.6224020 0.0002851917
## [212,] 5.5923437 0.0001933099
## [213,] 4.8992142 0.0002400502
## [214,] 4.0942198 0.0003872405
## [215,] 6.2576291 0.0001776799
## [216,] 5.7877748 0.0002006714
## [217,] 5.5746873 0.0002163737
## [218,] 6.0067192 0.0001720141
## [219,] 5.2339356 0.0002312565
## [220,] 4.4368745 0.0003100303
## [221,] 3.9131218 0.0005776879
## [222,] 3.5519691 0.0005848872
## [223,] 5.9281462 0.0001483610
## [224,] 7.2133474 0.0001403583
## [225,] 6.5279679 0.0001437225
## [226,] 5.7118967 0.0001914120
## [227,] 4.8400863 0.0003265589
## [228,] 5.5632760 0.0001755749
## [229,] 4.9524474 0.0002530633
## [230,] 5.4705215 0.0001974182
## [231,] 4.5650334 0.0003175363
## [232,] 4.5523985 0.0003652448
## [233,] 4.8449800 0.0002904953
## [234,] 5.1380367 0.0002157245
## [235,] 4.6917585 0.0003086773
## [236,] 4.6881285 0.0002540448
## [237,] 3.3586300 0.0005254888
## [238,] 3.1893298 0.0005241452
## [239,] 4.1682430 0.0003647268
## [240,] 6.3317596 0.0001743309
## [241,] 4.8475624 0.0002971849
## [242,] 5.7338633 0.0002079580
## [243,] 5.6310524 0.0002345144
## [244,] 3.9427197 0.0004060094
## [245,] 4.3980048 0.0003538934
## [246,] 4.3296103 0.0003746205
## [247,] 4.3101601 0.0002983073
## [248,] 3.3170823 0.0005763835
## [249,] 2.6476541 0.0007698911
## [250,] 5.1256366 0.0002631628
```

```
## [251,] 4.4783374 0.0003811954
## [252,] 4.0044530 0.0004253261
## [253,] 4.3308929 0.0003517832
## [254,] 6.3392853 0.0001567658
## [255,] 5.9162014 0.0002475396
## [256,] 4.0135761 0.0004317992
## [257,] 4.3419901 0.0003462691
## [258,] 2.7322930 0.0009003935
## [259,] 5.2782340 0.0002272977
## [260,] 4.6396538 0.0003101387
## [261,] 6.9186288 0.0001187405
## [262,] 6.7572838 0.0001398862
## [263,] 3.0572489 0.0008351247
## [264,] 4.3063249 0.0003332667
## [265,] 4.5329605 0.0003042020
## [266,] 5.3107221 0.0001836526
## [267,] 5.1015700 0.0002211803
## [268,] 4.9110596 0.0002777771
## [269,] 6.3427637 0.0001677926
## [270,] 4.4615069 0.0003182686
## [271,] 4.8582489 0.0002161806
## [272,] 4.4120903 0.0003310985
## [273,] 5.5305771 0.0001940013
## [274,] 3.4340254 0.0005675257
## [275,] 5.9650593 0.0001716862
## [276,] 4.0879204 0.0003966494
## [277,] 5.4846908 0.0001817391
## [278,] 4.9163596 0.0002491872
## [279,] 4.7004839 0.0002709802
## [280,] 5.0462874 0.0003000397
## [281,] 4.1918023 0.0003795921
## [282,] 6.5979687 0.0001794062
## [283,] 3.0373992 0.0007105141
## [284,] 5.1423827 0.0002704393
## [285,] 4.3483056 0.0003172078
## [286,] 5.2916161 0.0001876818
## [287,] 2.6156890 0.0010490164
## [288,] 2.8630790 0.0009051200
## [289,] 3.4074637 0.0005833247
## [290,] 4.4327510 0.0003336104
## [291,] 5.4280501 0.0002278381
## [292,] 5.2795149 0.0002516487
## [293,] 3.2080775 0.0006577021
## [294,] 4.8453615 0.0002878070
## [295,] 4.7103006 0.0002863198
## [296,] 3.0825465 0.0007516637
## [297,] 7.4532624 0.0001226768
## [298,] 3.9214843 0.0004166134
## [299,] 4.9772605 0.0002431802
## [300,] 2.0417520 0.0016426412
## [301,] 5.8363648 0.0002093049
## [302,] 6.3569127 0.0001519454
## [303,] 4.3644195 0.0003001331
## [304,] 7.7070940 0.0001074290
## [305,] 5.6025222 0.0002042282
## [306,] 5.2450320 0.0003043101
## [307,] 3.5606703 0.0004721212
## [308,] 5.4126107 0.0001807886
## [309,] 5.9937476 0.0001494209
## [310,] 4.8932078 0.0002624564
## [311,] 4.1743008 0.0003660481
## [312,] 5.1993830 0.0002297153
## [313,] 6.2003580 0.0001851097
## [314,] 5.1590339 0.0002458268
## [315,] 5.9563520 0.0001952770
## [316,] 5.4524557 0.0002048124
## [317,] 5.9847340 0.0001621854
## [318,] 5.5518484 0.0001888588
## [319,] 5.0975537 0.0002452256
## [320,] 5.1250459 0.0002282030
## [321,] 4.1724275 0.0002998525
## [322,] 4.4410454 0.0004062221
## [323,] 4.5344364 0.0003173712
## [324,] 6.1835953 0.0001782568
## [325,] 4.3354637 0.0003181461
## [326,] 3.3028666 0.0005501596
## [327,] 4.4875884 0.0003464142
## [328,] 4.0860649 0.0003854826
## [329,] 3.6708764 0.0005451742
## [330,] 5.2270239 0.0002387426
## [331,] 4.2592699 0.0003651808
## [332,] 4.1730536 0.0004479922
## [333,] 4.9329837 0.0003066487
## [334,] 6.3609925 0.0001526224
```



```
## [335,] 3.6979891 0.0005108551
## [336,] 5.7057248 0.0002172379
## [337,] 5.4512463 0.0002186454
## [338,] 4.1339901 0.0003379917
## [339,] 4.7421063 0.0003259452
## [340,] 6.3924367 0.0001479546
## [341,] 6.2033281 0.0001859529
## [342,] 6.7894026 0.0001168340
## [343,] 4.5397844 0.0003150086
## [344,] 4.6735490 0.0002941105
## [345,] 3.8778390 0.0004421606
## [346,] 5.8272881 0.0002038527
## [347,] 3.7762314 0.0004448482
## [348,] 4.8902662 0.0003256838
## [349,] 3.7431446 0.0004306425
## [350,] 3.3114481 0.0005494833
## [351,] 4.2190403 0.0004399736
## [352,] 4.7155167 0.0002901316
## [353,] 5.9127225 0.0001896833
## [354,] 4.6988852 0.0003679486
## [355,] 5.8838799 0.0002350358
## [356,] 3.8942414 0.0004584211
## [357,] 5.8909257 0.0001839718
## [358,] 5.9028480 0.0002114793
## [359,] 4.6232834 0.0003186483
## [360,] 6.6832687 0.0001394287
## [361,] 5.4773677 0.0002616251
## [362,] 4.9263290 0.0002286748
## [363,] 4.2592506 0.0003424782
## [364,] 5.2879032 0.0002282226
## [365,] 4.5809951 0.0002799778
## [366,] 3.8042894 0.0004911481
## [367,] 3.0911867 0.0006898863
## [368,] 5.2781437 0.0002298969
## [369,] 3.3819344 0.0005998469
## [370,] 4.6336087 0.0002578587
## [371,] 6.0227814 0.0001812603
## [372,] 4.5503194 0.0002259687
## [373,] 4.0552862 0.0004591804
## [374,] 3.2590539 0.0007268265
## [375,] 4.9687479 0.0002283471
## [376,] 4.0413787 0.0004586551
## [377,] 4.8407328 0.0002790962
## [378,] 6.1777994 0.0001665977
## [379,] 3.9627347 0.0004580768
## [380,] 5.5412859 0.0002281426
## [381,] 5.4912720 0.0002059212
## [382,] 3.4598888 0.0005879532
## [383,] 4.6797292 0.0003894400
## [384,] 4.7361446 0.0002838496
## [385,] 4.3902824 0.0003033239
## [386,] 5.2224366 0.0002589489
## [387,] 4.9876886 0.0002376554
## [388,] 6.3556186 0.0001478099
## [389,] 5.3057987 0.0002357137
## [390,] 4.7023647 0.0002508593
## [391,] 6.5338725 0.0001405255
## [392,] 4.6686388 0.0002740631
## [393,] 5.6782916 0.0002198512
## [394,] 4.8445556 0.0002440655
## [395,] 4.3751808 0.0002879116
## [396,] 4.9589328 0.0002624353
## [397,] 3.2457609 0.0007073354
## [398,] 4.4233968 0.0003334521
## [399,] 4.9120160 0.0002326608
## [400,] 6.0734775 0.0001908190
## [401,] 5.3324283 0.0002487716
## [402,] 4.2551478 0.0003534145
## [403,] 4.5773906 0.0002918858
## [404,] 6.2298512 0.0001586440
## [405,] 5.4716597 0.0002435124
## [406,] 5.1480598 0.0002587057
## [407,] 4.2551252 0.0005117636
## [408,] 5.5635129 0.0002086952
## [409,] 4.0689553 0.0003928045
## [410,] 4.9802517 0.0002874599
## [411,] 5.2693692 0.0001676775
## [412,] 6.6193697 0.0001678178
## [413,] 7.3125521 0.0001318106
## [414,] 6.6371843 0.0001349439
## [415,] 5.2065475 0.0002931288
## [416,] 6.6178056 0.0001620289
## [417,] 5.8564892 0.0001985827
## [418,] 2.9826665 0.0007250475
```

```
## [419,] 2.4753571 0.0010345448
## [420,] 4.3424192 0.0003361044
## [421,] 4.9594887 0.0002621767
## [422,] 3.6656845 0.0005129062
## [423,] 4.0106271 0.0004458744
## [424,] 5.0585320 0.0002468690
## [425,] 1.1379730 0.0053701641
## [426,] 1.7924312 0.0022089331
## [427,] 2.9926266 0.0008289202
## [428,] 5.5795403 0.0001963352
## [429,] 5.9158780 0.0001463383
## [430,] 5.5069408 0.0002261215
## [431,] 3.2804318 0.0008118316
## [432,] 2.7650009 0.0008535608
## [433,] 4.0700437 0.0004052104
## [434,] 6.3068848 0.0001421514
## [435,] 5.4990023 0.0001798852
## [436,] 4.4444221 0.0003078690
## [437,] 4.5280018 0.0002817936
## [438,] 5.4939048 0.0002172456
## [439,] 5.5697523 0.0002090846
## [440,] 6.0086677 0.0001862599
## [441,] 3.7660107 0.0004933074
## [442,] 4.0959936 0.0003699954
## [443,] 3.4133330 0.0005808777
## [444,] 5.9048417 0.0001759600
## [445,] 3.9288742 0.0004517148
## [446,] 3.4241446 0.0005820351
## [447,] 4.8799446 0.0002704120
## [448,] 4.1396153 0.0004244939
## [449,] 6.5090268 0.0001511326
## [450,] 4.3818187 0.0003265183
## [451,] 3.2496500 0.0005845312
## [452,] 5.5760079 0.0002128874
## [453,] 6.8394299 0.0001290126
## [454,] 7.0073609 0.0001219943
## [455,] 4.5297806 0.0003042734
## [456,] 4.1917728 0.0003937752
## [457,] 5.4836852 0.0002377226
## [458,] 3.5783493 0.0005536666
## [459,] 5.0183863 0.0002623701
## [460,] 5.3357061 0.0002575484
## [461,] 4.3828063 0.0003999108
## [462,] 3.7733025 0.0005292986
## [463,] 5.2007976 0.0002172364
## [464,] 6.2464921 0.0001521143
## [465,] 5.8111865 0.0002034949
## [466,] 6.1593189 0.0001393885
## [467,] 6.0406579 0.0001732126
## [468,] 4.1010812 0.0003929629
## [469,] 4.4952777 0.0002764665
## [470,] 6.8629890 0.0001454978
## [471,] 4.5954631 0.0003319772
## [472,] 4.1540036 0.0003762105
## [473,] 5.5043649 0.0002025838
## [474,] 4.7751172 0.0003654524
## [475,] 2.2140297 0.0011598447
## [476,] 4.1561317 0.0003911749
## [477,] 3.9663197 0.0004148225
## [478,] 4.7127805 0.0003104538
## [479,] 4.1564600 0.0003571519
## [480,] 4.4231454 0.0003445100
## [481,] 6.0415622 0.0001946413
## [482,] 4.7014259 0.0002991610
## [483,] 6.0597684 0.0001642865
## [484,] 5.0728571 0.0002843872
## [485,] 4.3128320 0.0004086574
## [486,] 7.1483957 0.0001228659
## [487,] 3.5157503 0.0004677045
## [488,] 4.6742617 0.0003035651
## [489,] 3.9983678 0.0003627946
## [490,] 4.7408510 0.0003233612
## [491,] 4.1086942 0.0003873085
## [492,] 4.6847614 0.0003091181
## [493,] 5.1214012 0.0002447631
## [494,] 5.6438122 0.0002096741
## [495,] 5.3428548 0.0002596655
## [496,] 4.1065631 0.0002960301
## [497,] 5.3399722 0.0002045505
## [498,] 6.8336457 0.0001440132
## [499,] 5.3454012 0.0002268770
## [500,] 3.9207059 0.0004959400
## [501,] 5.4391612 0.0001849257
## [502,] 3.3804444 0.0005320171
```

```
## [503,] 6.2507432 0.0001563816
## [504,] 5.5747889 0.0001981906
## [505,] 5.3259471 0.0002364302
## [506,] 4.2313195 0.0003786622
## [507,] 4.0256086 0.0003927089
## [508,] 5.1705582 0.0002671275
## [509,] 5.1666949 0.0002659043
## [510,] 6.2095286 0.0001612919
## [511,] 5.0922506 0.0002860647
## [512,] 4.1944840 0.0003543289
## [513,] 3.9646719 0.0004180227
## [514,] 4.0061653 0.0003860874
## [515,] 3.9691330 0.0004056356
## [516,] 3.7116526 0.0004964298
## [517,] 6.0761130 0.0001923099
## [518,] 4.8602870 0.0002697219
## [519,] 3.6225019 0.0004019464
## [520,] 5.3757334 0.0002661963
## [521,] 4.4819837 0.0002879751
## [522,] 3.4730667 0.0005887465
## [523,] 3.8759301 0.0004483633
## [524,] 3.3756019 0.0006322537
## [525,] 3.8103144 0.0004448245
## [526,] 3.8097044 0.0003447038
## [527,] 4.9381305 0.0003190981
## [528,] 3.7786419 0.0003795720
## [529,] 4.8189431 0.0002752105
## [530,] 3.0630054 0.0008589802
## [531,] 1.9696465 0.0016745515
## [532,] 3.8657676 0.0003571288
## [533,] 4.9406861 0.0002780742
## [534,] 2.7099813 0.0008177562
## [535,] 3.8508261 0.0004945904
## [536,] 5.1649096 0.0002422804
## [537,] 4.0522320 0.0004059510
## [538,] 5.1299880 0.0002202549
## [539,] 5.0881834 0.0002295842
## [540,] 5.2796246 0.0002267059
## [541,] 4.5884705 0.0002789012
## [542,] 6.2532427 0.0001310505
## [543,] 5.9019300 0.0001858507
## [544,] 5.2664328 0.0002686031
## [545,] 2.8040194 0.0007918442
## [546,] 4.8916575 0.0002740409
## [547,] 5.1322361 0.0002813148
## [548,] 4.7338425 0.0002925835
## [549,] 3.5161722 0.0004842168
## [550,] 4.6640420 0.0003112195
## [551,] 4.1151823 0.0003807973
## [552,] 6.1635884 0.0001661322
## [553,] 4.4409980 0.0003199103
## [554,] 3.9444907 0.0005357785
## [555,] 6.2029656 0.0001588923
## [556,] 4.7202667 0.0002972747
## [557,] 4.9174290 0.0003298593
## [558,] 3.6066959 0.0005158518
## [559,] 3.8778869 0.0004514571
## [560,] 5.8387143 0.0001813927
## [561,] 5.7122256 0.0001715775
## [562,] 3.6113870 0.0004868249
## [563,] 4.9711093 0.0002681827
## [564,] 3.5230252 0.0005571457
## [565,] 5.6434248 0.0001819507
## [566,] 3.7286235 0.0004926907
## [567,] 3.2788719 0.0006383367
## [568,] 5.4146797 0.0002279559
## [569,] 5.3516518 0.0002394541
## [570,] 4.6213026 0.0003027421
## [571,] 6.0551155 0.0001960518
## [572,] 4.2885335 0.0003807046
## [573,] 5.2827242 0.0002368971
## [574,] 4.5811795 0.0003617197
## [575,] 6.2204511 0.0001744406
## [576,] 5.3465515 0.0002673306
## [577,] 4.0279503 0.0004316042
## [578,] 3.5263609 0.0004857429
## [579,] 3.8189299 0.0004633629
## [580,] 5.6176763 0.0002237642
## [581,] 5.0731111 0.0002710577
## [582,] 4.6825967 0.0003649184
## [583,] 5.6811049 0.0002151425
## [584,] 4.5010661 0.0003032606
## [585,] 3.2486808 0.0005992944
## [586,] 4.2887540 0.0003859041
```

```
## [587,] 5.6349277 0.0002335441
## [588,] 4.6367924 0.0002951009
## [589,] 4.8215748 0.0003186476
## [590,] 5.0598791 0.0002643687
## [591,] 5.1059641 0.0002662357
## [592,] 5.1679811 0.0002530230
## [593,] 6.7339540 0.0001424921
## [594,] 5.3957918 0.0002310436
## [595,] 5.9730972 0.0001875023
## [596,] 4.8169992 0.0003350591
## [597,] 5.5411804 0.0002296818
## [598,] 4.4121825 0.0003091057
## [599,] 3.7753893 0.0004943617
## [600,] 4.1482630 0.0004205540
## [601,] 4.1715048 0.0004308260
## [602,] 4.4190660 0.0003792920
## [603,] 3.7347373 0.0004934762
## [604,] 4.8744610 0.0002519890
## [605,] 6.0650340 0.0001750871
## [606,] 4.9534105 0.0002457953
## [607,] 5.4109519 0.0002133391
## [608,] 3.3580928 0.0006048200
## [609,] 1.5606773 0.0020663332
## [610,] 3.1483301 0.0007491048
## [611,] 4.6040889 0.0003023659
## [612,] 4.7166983 0.0002323415
## [613,] 4.9453824 0.0002507816
## [614,] 3.3134952 0.0005423570
## [615,] 4.8052068 0.0002894248
## [616,] 4.4409731 0.0003507360
## [617,] 4.2470975 0.0003892615
## [618,] 4.8457526 0.0002334530
## [619,] 6.3139746 0.0001776689
## [620,] 5.7672605 0.0002358485
## [621,] 4.6913848 0.0002776905
## [622,] 7.2724164 0.0001216312
## [623,] 3.7419085 0.0004092199
## [624,] 4.5345418 0.0003630315
## [625,] 5.2157536 0.0002049331
## [626,] 3.8801379 0.0003815361
## [627,] 3.6909691 0.0004414104
## [628,] 6.7410355 0.0001858004
## [629,] 4.9199754 0.0002351679
## [630,] 4.9177022 0.0002528815
## [631,] 4.8909709 0.0002962166
## [632,] 4.3458352 0.0002779708
## [633,] 6.1121788 0.0002071983
## [634,] 5.4288043 0.0002340596
## [635,] 4.6552184 0.0003061879
## [636,] 3.9340916 0.0004617949
## [637,] 6.3026530 0.0001657256
## [638,] 4.5683872 0.0003182160
## [639,] 3.4117192 0.0005789018
## [640,] 5.3941342 0.0002240545
## [641,] 4.0077064 0.0003908050
## [642,] 5.0177856 0.0002774190
## [643,] 3.6967481 0.0004457935
## [644,] 3.0728065 0.0008064687
## [645,] 4.3692992 0.0003977968
## [646,] 3.8517163 0.0003941354
## [647,] 4.8592512 0.0002414959
## [648,] 3.8079100 0.0005364719
## [649,] 5.5483875 0.0002173692
## [650,] 3.6919035 0.0004717548
## [651,] 4.2446821 0.0004302392
## [652,] 2.7024485 0.0007545733
## [653,] 2.7727775 0.0008514564
## [654,] 5.2082539 0.0002044347
## [655,] 6.4640285 0.0001467022
## [656,] 3.6726080 0.0005279324
## [657,] 6.3836611 0.0001653431
## [658,] 3.9399388 0.0004806482
## [659,] 6.8859066 0.0001662227
## [660,] 4.1344539 0.0003589786
## [661,] 4.5403951 0.0003428673
## [662,] 3.8115942 0.0004256146
## [663,] 5.8713667 0.0001541554
## [664,] 3.5271177 0.0005307442
## [665,] 5.6408423 0.0002084353
## [666,] 3.6111303 0.0005952266
## [667,] 5.1212795 0.0002634922
## [668,] 5.5508741 0.0001991495
## [669,] 3.8794517 0.0004790562
## [670,] 5.0976682 0.0002677981
```

```
## [671,] 5.5614182 0.0002322215
## [672,] 5.7686929 0.0002218404
## [673,] 4.0593325 0.0004381906
## [674,] 3.3884916 0.0005779225
## [675,] 4.7949586 0.0002718963
## [676,] 3.6781208 0.0004851860
## [677,] 5.1857737 0.0002692014
## [678,] 4.9243910 0.0002292072
## [679,] 3.9714397 0.0003768612
## [680,] 3.7264696 0.0005044152
## [681,] 4.0898344 0.0003184515
## [682,] 5.8513547 0.0001703959
## [683,] 5.1967605 0.0002802174
## [684,] 5.0249730 0.0002744680
## [685,] 4.7611413 0.0002739324
## [686,] 5.0711161 0.0002346616
## [687,] 4.3981352 0.0003801028
## [688,] 5.8347326 0.0001917863
## [689,] 4.3446709 0.0003149208
## [690,] 4.7861521 0.0002768543
## [691,] 3.5988126 0.0004981207
## [692,] 4.6754386 0.0002795399
## [693,] 6.4904551 0.0001654512
## [694,] 4.3079955 0.0003596975
## [695,] 6.9179179 0.0001291681
## [696,] 5.5702147 0.0002213771
## [697,] 6.2679663 0.0001892834
## [698,] 5.9314151 0.0002228155
## [699,] 4.8857854 0.0003341417
## [700,] 4.0249299 0.0004178152
## [701,] 4.0028391 0.0003706140
## [702,] 2.7040001 0.0011168890
## [703,] 2.7322115 0.0010769260
## [704,] 2.9778692 0.0007046812
## [705,] 5.6217004 0.0002367388
## [706,] 5.2128702 0.0002566402
## [707,] 5.6467390 0.0002204011
## [708,] 4.6212846 0.0003414147
## [709,] 3.9533549 0.0003796994
## [710,] 5.8598918 0.0001916449
## [711,] 3.6439532 0.0004948944
## [712,] 5.2380137 0.0002700003
## [713,] 5.5103900 0.0002326501
## [714,] 4.9787794 0.0002429463
## [715,] 2.7777594 0.0008478741
## [716,] 5.4585322 0.0001847879
## [717,] 3.1529723 0.0006795654
## [718,] 4.8781531 0.0002802297
## [719,] 4.8644815 0.0002664297
## [720,] 6.0493587 0.0001494915
## [721,] 6.5339783 0.0001705435
## [722,] 3.1765918 0.0006606607
## [723,] 5.3558102 0.0002147564
## [724,] 4.4697805 0.0003032347
## [725,] 6.0222436 0.0001636091
## [726,] 3.8813090 0.0004288598
## [727,] 4.5855396 0.0003331117
## [728,] 3.8790243 0.0005110432
## [729,] 4.8948417 0.0002925521
## [730,] 4.0318266 0.0005077879
## [731,] 6.0416685 0.0001709393
## [732,] 5.3689789 0.0001847726
## [733,] 4.9438427 0.0002414599
## [734,] 4.6345794 0.0002567386
## [735,] 5.1974339 0.0002828356
## [736,] 6.0163698 0.0001715109
## [737,] 3.7766128 0.0005251093
## [738,] 5.0049991 0.0002794050
## [739,] 4.2218000 0.0003312511
## [740,] 5.0884822 0.0002347647
## [741,] 4.4246238 0.0003872887
## [742,] 3.9212116 0.0004257769
## [743,] 5.0463974 0.0002792039
## [744,] 4.3937675 0.0003424076
## [745,] 4.7423088 0.0002797337
## [746,] 4.6408478 0.0002921784
## [747,] 5.4191553 0.0002137907
## [748,] 5.2306200 0.0002518997
## [749,] 6.0062222 0.0001878741
## [750,] 4.8156507 0.0002890650
## [751,] 6.9772057 0.0001326987
## [752,] 5.8834332 0.0001786703
## [753,] 5.6563670 0.0002311479
## [754,] 5.5074283 0.0002345220
```

```
## [755,] 4.2308481 0.0003316130
## [756,] 4.7110785 0.0002951765
## [757,] 3.4487466 0.0005378783
## [758,] 4.5856670 0.0003191786
## [759,] 4.7379851 0.0002815124
## [760,] 4.5971373 0.0003148460
## [761,] 5.0909925 0.0002828113
## [762,] 4.9694916 0.0002629954
## [763,] 3.6332883 0.0004044027
## [764,] 3.5229509 0.0005795892
## [765,] 4.1067448 0.0004053074
## [766,] 4.1369589 0.0004344171
## [767,] 3.9930051 0.0004605999
## [768,] 5.0841181 0.0002175035
## [769,] 5.6541738 0.0001807143
## [770,] 5.4405134 0.0001934353
## [771,] 2.5161856 0.0009182821
## [772,] 3.7945880 0.0003442516
## [773,] 3.9382937 0.0004193350
## [774,] 6.1478138 0.0001757526
## [775,] 3.2716679 0.0006823649
## [776,] 5.0441259 0.0002510548
## [777,] 5.2811886 0.0002394767
## [778,] 3.7324591 0.0004974158
## [779,] 6.2917150 0.0001676080
## [780,] 6.5993967 0.0001512343
## [781,] 5.6036152 0.0001741255
## [782,] 2.7982278 0.0009054190
## [783,] 4.7816922 0.0002420094
## [784,] 5.6832249 0.0001890194
## [785,] 4.2976142 0.0003608821
## [786,] 6.1810296 0.0001576046
## [787,] 5.9555444 0.0001978005
## [788,] 2.9940881 0.0005745963
## [789,] 4.6581798 0.0003249779
## [790,] 5.7191838 0.0001821007
## [791,] 5.3309406 0.0002520277
## [792,] 5.8824972 0.0001859309
## [793,] 4.5393778 0.0002978318
## [794,] 3.8818303 0.0003573701
## [795,] 5.9038103 0.0001970399
## [796,] 3.9018556 0.0004372453
## [797,] 6.4796502 0.0001410744
## [798,] 5.6448697 0.0001864339
## [799,] 1.8091358 0.0022284854
## [800,] 2.1422019 0.0013044712
## [801,] 3.1074024 0.0007676573
## [802,] 3.7436377 0.0004978649
## [803,] 4.7575540 0.0003451770
## [804,] 4.4309857 0.0003314439
## [805,] 6.9539578 0.0001109097
## [806,] 5.7083478 0.0001855574
## [807,] 4.9481807 0.0002848230
## [808,] 5.4392144 0.0001889267
## [809,] 4.6482706 0.0002781054
## [810,] 3.6968438 0.0004377752
## [811,] 5.8625893 0.0001826396
## [812,] 5.7072338 0.0002121798
## [813,] 5.7619391 0.0002430383
## [814,] 4.4995770 0.0003403262
## [815,] 4.7630695 0.0002745807
## [816,] 5.1331683 0.0002969106
## [817,] 3.6951372 0.0004823397
## [818,] 5.4872649 0.0002182364
## [819,] 5.2441638 0.0002299038
## [820,] 6.1725996 0.0001447164
## [821,] 6.1969004 0.0001799990
## [822,] 2.9136660 0.0007478126
## [823,] -0.6430584 0.0145010099
## [824,] 2.4451701 0.0010626781
## [825,] 2.8893200 0.0009758628
## [826,] 4.8540364 0.0003542060
## [827,] 4.5573651 0.0002641487
## [828,] 3.5619010 0.0005739889
## [829,] 4.2100007 0.0003100380
## [830,] 4.1079168 0.0004129002
## [831,] 4.4865015 0.0003597496
## [832,] 3.5260155 0.0005082815
## [833,] 5.6094399 0.0002207298
## [834,] 5.9479627 0.0001783286
## [835,] 7.3363210 0.0001217806
## [836,] 5.3937337 0.0002075279
## [837,] 4.6965490 0.0003213734
## [838,] 4.3147659 0.0003616906
```

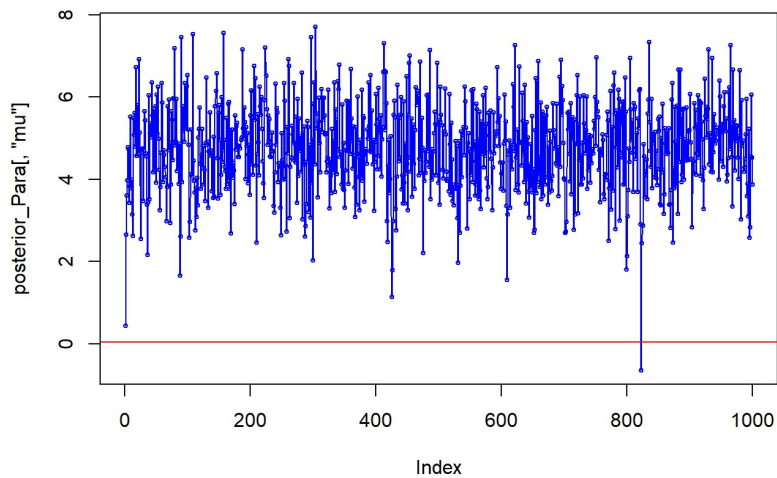
```
## [839,] 5.9580400 0.0001574224
## [840,] 3.7924734 0.0004174549
## [841,] 4.8785265 0.0002940978
## [842,] 5.1107714 0.0002593433
## [843,] 5.1189667 0.0002479142
## [844,] 4.0633300 0.0004268974
## [845,] 5.1585964 0.0002431130
## [846,] 4.6909588 0.0002812513
## [847,] 4.0233614 0.0003800267
## [848,] 3.8587499 0.0003973078
## [849,] 4.2503816 0.0003279198
## [850,] 5.4623534 0.0002164136
## [851,] 5.1794229 0.0002118068
## [852,] 5.5319122 0.0002259251
## [853,] 3.4042642 0.0005414244
## [854,] 3.3772066 0.0005780199
## [855,] 5.0051692 0.0002447538
## [856,] 4.1774372 0.0003904388
## [857,] 5.8401901 0.0001875761
## [858,] 4.2370406 0.0003970172
## [859,] 3.1673879 0.0006621699
## [860,] 3.6462650 0.0004748422
## [861,] 5.2458740 0.0002036839
## [862,] 4.8680555 0.0002964316
## [863,] 4.6184342 0.0003366867
## [864,] 5.1791792 0.0003079925
## [865,] 4.6538498 0.0003038357
## [866,] 5.2519510 0.0002142629
## [867,] 3.3844497 0.0006207602
## [868,] 3.3999620 0.0005958826
## [869,] 2.8267027 0.0007189673
## [870,] 3.7480753 0.0003856829
## [871,] 5.9371534 0.0001894339
## [872,] 6.3460706 0.0001673662
## [873,] 2.4725104 0.0009844052
## [874,] 4.2823037 0.0003666138
## [875,] 3.7909752 0.0005587717
## [876,] 5.7658095 0.0001630761
## [877,] 4.9225132 0.0003057836
## [878,] 3.8321485 0.0004346994
## [879,] 4.2392637 0.0003250745
## [880,] 5.8031647 0.0002528985
## [881,] 3.1596271 0.0006748206
## [882,] 6.6769490 0.0001861959
## [883,] 4.4228658 0.0003432248
## [884,] 5.0379288 0.0002577873
## [885,] 6.6661595 0.0001603912
## [886,] 5.5500435 0.0002451214
## [887,] 3.9404041 0.0003889840
## [888,] 4.1115510 0.0003895892
## [889,] 5.6910783 0.0001995368
## [890,] 5.7172071 0.0001388254
## [891,] 3.9981504 0.0003686461
## [892,] 4.9722072 0.0002900508
## [893,] 4.7409034 0.0003026301
## [894,] 4.4577293 0.0003956730
## [895,] 5.7501476 0.0002112976
## [896,] 4.4836649 0.0003438520
## [897,] 4.9359897 0.0002495127
## [898,] 4.4758052 0.0003167054
## [899,] 5.1666503 0.0002285279
## [900,] 5.1953756 0.0002481959
## [901,] 6.0724498 0.0001711835
## [902,] 4.9111715 0.0002557637
## [903,] 4.5063705 0.0003628237
## [904,] 2.8446175 0.0008959555
## [905,] 4.4469447 0.0003229910
## [906,] 4.1412592 0.0004300222
## [907,] 5.8543182 0.0001553894
## [908,] 5.2578808 0.0002443736
## [909,] 5.6851434 0.0002513162
## [910,] 4.1167694 0.0003338309
## [911,] 4.6377100 0.0003475624
## [912,] 5.3287005 0.0002555264
## [913,] 6.0556040 0.0001899781
## [914,] 4.5250166 0.0003441968
## [915,] 5.9103489 0.0001755102
## [916,] 5.8883998 0.0001647301
## [917,] 4.0932249 0.0003926105
## [918,] 5.8532165 0.0001996330
## [919,] 5.4063367 0.0002477444
## [920,] 4.9822504 0.0002634977
## [921,] 3.9795816 0.0003823836
## [922,] 3.2896309 0.0005717832
```

```
## [923,] 4.2858855 0.0004210555
## [924,] 6.4544673 0.0001401391
## [925,] 5.5422016 0.0002426271
## [926,] 5.7669505 0.0002123047
## [927,] 5.0172242 0.0002480984
## [928,] 5.4657656 0.0002293078
## [929,] 5.7240580 0.0001722932
## [930,] 7.1646742 0.0001202635
## [931,] 4.3633899 0.0003233221
## [932,] 5.1213656 0.0002207209
## [933,] 5.0336942 0.0002733359
## [934,] 4.1869121 0.0004025269
## [935,] 4.3165928 0.0004337066
## [936,] 6.9500933 0.0001424254
## [937,] 4.1727181 0.0004593004
## [938,] 4.6595597 0.0003201007
## [939,] 5.5892481 0.0002199986
## [940,] 5.3292487 0.0002426473
## [941,] 4.8767753 0.0002407609
## [942,] 4.9479420 0.0002574247
## [943,] 5.1001313 0.0001831138
## [944,] 5.2671939 0.0002161191
## [945,] 5.1788019 0.0002529816
## [946,] 4.0003800 0.0003986999
## [947,] 6.3984870 0.0001402225
## [948,] 6.0179292 0.0002044036
## [949,] 4.4405292 0.0004148967
## [950,] 5.3281562 0.0002566725
## [951,] 5.1406216 0.0002403732
## [952,] 5.3514696 0.0002632853
## [953,] 4.3042805 0.0003292361
## [954,] 5.7742957 0.0001732017
## [955,] 6.1163323 0.0002002347
## [956,] 5.1646716 0.0002568302
## [957,] 5.1645395 0.0002340861
## [958,] 5.4840845 0.0002485307
## [959,] 3.9566122 0.0004237719
## [960,] 4.1181523 0.0003577565
## [961,] 4.5226435 0.0003345675
## [962,] 6.1263453 0.0001715609
## [963,] 6.2418168 0.0001171121
## [964,] 4.7453746 0.0002755966
## [965,] 7.2612276 0.0001450450
## [966,] 5.3426428 0.0002364716
## [967,] 4.5518136 0.0003226598
## [968,] 3.3445248 0.0004319017
## [969,] 6.6664767 0.0001370620
## [970,] 4.9495472 0.0002236571
## [971,] 5.4447907 0.0002315811
## [972,] 4.2146488 0.0005418587
## [973,] 5.0239174 0.0002116569
## [974,] 5.3192496 0.0002317958
## [975,] 4.9306836 0.0002162569
## [976,] 5.3025392 0.0002083508
## [977,] 4.5372928 0.0002545636
## [978,] 4.0147955 0.0003250215
## [979,] 5.0526920 0.0002794426
## [980,] 6.6516315 0.0001482543
## [981,] 4.9416765 0.0002994865
## [982,] 3.0363767 0.0005752417
## [983,] 5.9314302 0.0001993049
## [984,] 4.8199376 0.0002945811
## [985,] 4.9443619 0.0002991622
## [986,] 4.1978840 0.0003031705
## [987,] 4.2904879 0.0003689914
## [988,] 4.4976443 0.0003425321
## [989,] 5.0921211 0.0002106908
## [990,] 5.9550341 0.0001673444
## [991,] 3.5919136 0.0005032056
## [992,] 3.8902209 0.0004852594
## [993,] 3.1119282 0.0006161512
## [994,] 5.2357280 0.0002174530
## [995,] 2.5892520 0.0010054883
## [996,] 2.8397970 0.0010289230
## [997,] 4.5618841 0.0002757563
## [998,] 6.0597876 0.0001715115
## [999,] 4.5273694 0.0003302444
## [1000,] 3.8783327 0.0003987759
```



```
posterior_Para <- as.data.frame(output)
names(posterior_Para) <- c("mu", "lambda=1/sigma^2")

plot(posterior_Para[, "mu"], type='o', cex=.5, col=4) # Traceplot
abline(h=mean(y), col=2)
```



We can now visualize the posterior distribution and create an HPD interval for all the variables in the model.

```
print("The below shows the posterior credibility interval")
```

```
## [1] "The below shows the posterior credibility interval"
```

```
HPDinterval(B,prob=.95)
```

```
##               lower      upper
## (Intercept) -9.298935e-04 1.138640e-03
## equipOS-CC  -1.078090e-03 1.008721e-03
## equipOS-PC  -1.354587e-03 1.499532e-03
## equipPAPER  -3.568837e-03 3.62519e-03
## equipPUNCH  -1.593532e-03 1.422974e-03
## econpoor    -9.319083e-04 1.081213e-03
## econrich    -1.769492e-03 1.480296e-03
## gore        -1.637082e-06 1.291341e-07
## bush        -1.553927e-06 1.338634e-07
## ballots     -1.250755e-07 1.506955e-06
## undercount   9.763523e-01 1.017241e+00
## attr(,"Probability")
## [1] 0.95
```

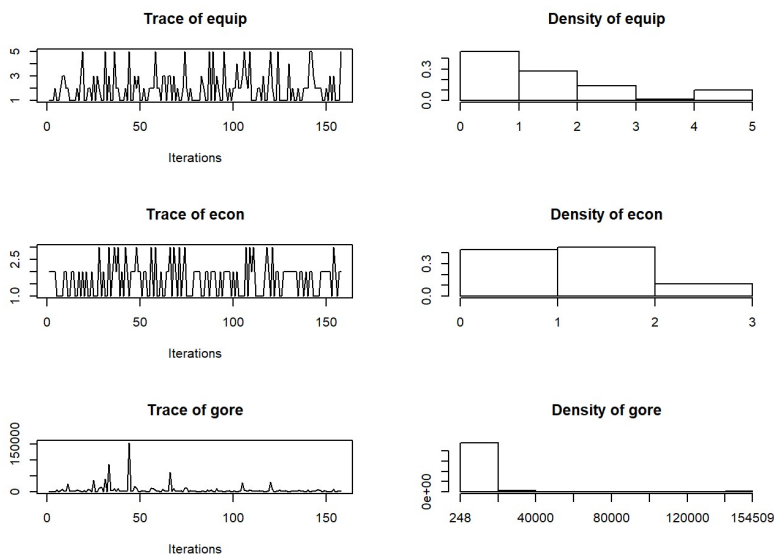
We can also see the distribution of our samples at the variable level:

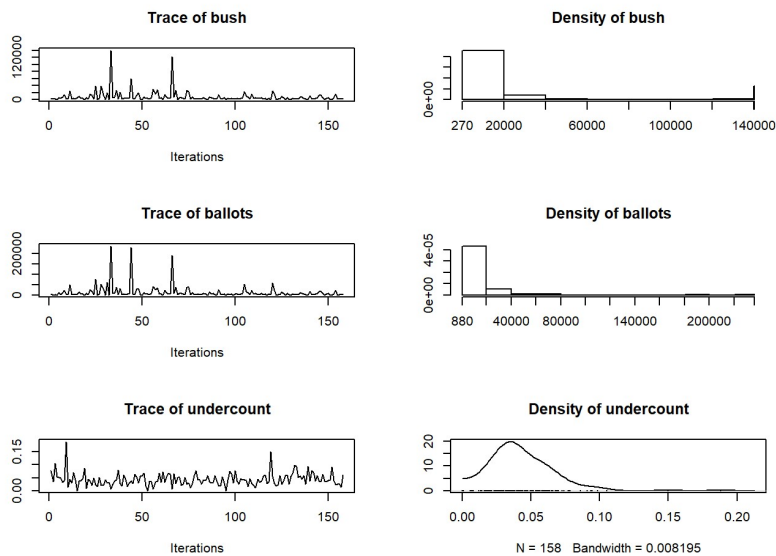
```
##### Gibbs Analysis #####

x = as.ts(gavot1)
SAMPLES<-as.mcmc(x) # converting your samples into an MCMC object
# Set lag of 20
autocorr(SAMPLES, lags = c(20), relative=TRUE)
```

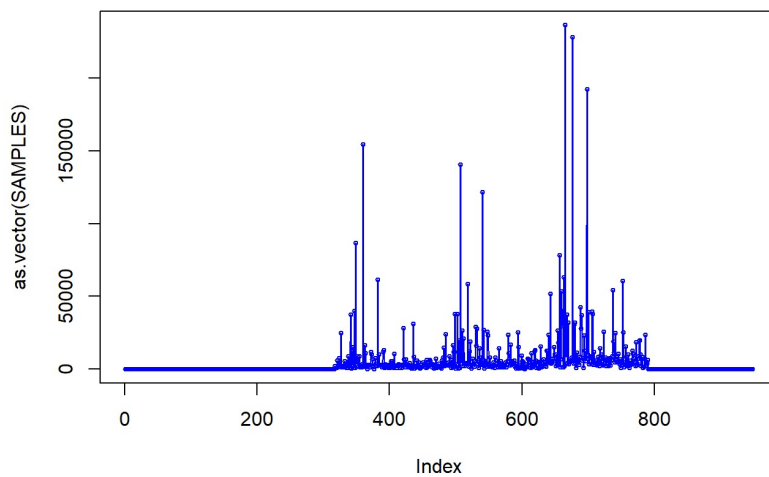
```
## , , equip
##
##      equip      econ      gore      bush      ballots undercount
## Lag 20 -0.1370968 0.0486471 -0.08550143 -0.054223 -0.07376014 -0.05477138
##
## , , econ
##
##      equip      econ      gore      bush      ballots undercount
## Lag 20 -0.01832662 -0.01086361 0.04779106 0.06728126 0.06143212 -0.09168336
##
## , , gore
##
##      equip      econ      gore      bush      ballots undercount
## Lag 20 -0.1003029 0.0346935 -0.02326343 -0.02531419 -0.02594341 -0.03720016
##
## , , bush
##
##      equip      econ      gore      bush      ballots undercount
## Lag 20 -0.09561629 0.09084396 -0.02711586 -0.01091682 -0.02068502 -0.1649025
##
## , , ballots
##
##      equip      econ      gore      bush      ballots undercount
## Lag 20 -0.1037921 0.06743382 -0.02630033 -0.01838111 -0.02405078 -0.107293
##
## , , undercount
##
##      equip      econ      gore      bush      ballots undercount
## Lag 20 -0.01565368 -0.05810611 0.1189557 0.1163424 0.1259678 0.0126133
```

```
plot(SAMPLES) # this produces both density and trace plot
```

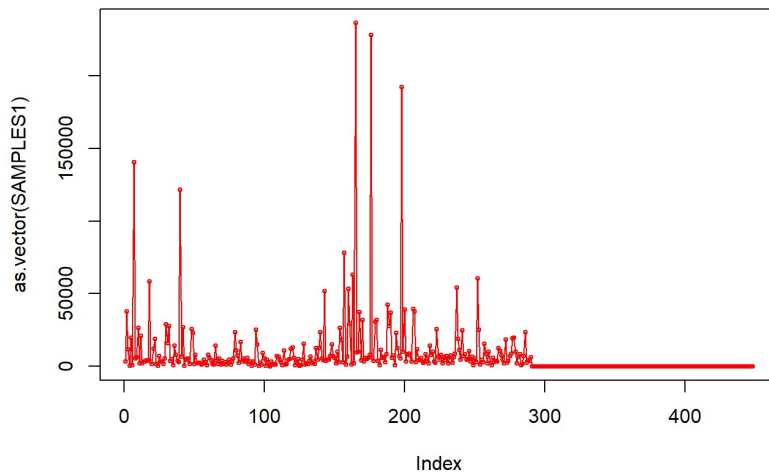




```
plot(as.vector(SAMPLES), type='o', cex=.5, col=4)
```



```
burnIn=1:500
SAMPLES1=SAMPLES[-burnIn] # removing burn-in
plot(as.vector(SAMPLES1), type='o', cex=.5, col=2)
```



```
summary(SAMPLES1)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
##      0.00     0.05   2566.00   7925.52   6825.25  236525.00
```

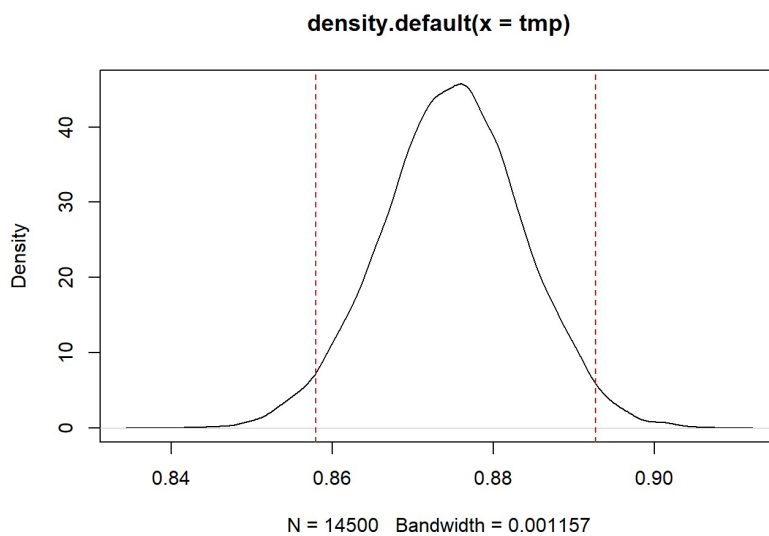
```
effectiveSize(SAMPLES1)
```

```
##      var1
## 31.75344
```

Now we can draw from our prior and perform hypothesis testing. This is not effective as with our prior we have nothing to test against. We can, however, see the density of our posterior \* sample of the prior.

```
##### Bayesian Linear Hypothesis #####

# Drawing from our flat prior [0:1]
contrast=runif(n=11, min=1e-12, max=.9999999999)
tmp=B%*contrast
plot(density(tmp));abline(v=0)
abline(v=HPDinterval(as.mcmc(tmp),p=.95),col=2,lty=2)
```



```
mean(tmp>0)
```

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

```
mean(tmp<0)
```

## [1] 0

We can see from this regression output that it modeled the data similarly to the output of the `lm()` function used above. We were able to expand our sample using the Gibbs Sample, which decreased cross-variable correlation and co-linearity significantly. These aspects of the Bayesian Analysis are strengths when compared to the linear model.

## Final Models and Statistical Analysis

Overall, we have two final models:

### OLS Frequentist Regression

Final Model (OLS Frequentist Regression):

$$\text{UNDERCOUNT} = 5.807496 * 10^{-3}x_1 + 2.273752 * 10^{-2}x_2 - 9.044148 * 10^{-3}x_3 + 6.082960 * 10^{-3}x_4 + 2.089696 * 10^{-2}x_5 - 1.894076 * 10^{-2}x_6 \\ - 3.554025 * 10^{-5}x_7 - 3.386005 * 10^{-5}x_8 + 3.282120 * 10^{-5}x_9 + 2.898298 * 10^{-2} + \epsilon_i$$

$$\text{UNDERCOUNT} = \beta_1 + \text{EQUIP}x_1 + \text{ECON}x_2 + \text{GORE}x_3 + \text{BUSH}x_4 + \text{BALLOTS}x_5 + \epsilon_i$$

The above functions show both the specific numerical coefficients within our regression function as well as the generic variables they represent.

### Bayesian Analysis

Within a Bayesian Analysis (applied using a Normal Model) our result is our prior, posteriors and likelihood function:

Final Model (Bayesian Approach):

Likelihood function:

Likelihood Function of Undercount (Y):

$$p(\text{UNDERCOUNT (Y)}|\text{EQUIP,ECON,GORE,BUSH,BALLOTS}) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(\text{UNDERCOUNT}_i - \beta_1 - \text{EQUIP}_2x_i - \text{ECON}_3x_i - \right. \\ \left. - \text{BUSH}_5x_i - \text{BALLOTS}_6x_i)^2\right\}$$

Prior:

Flat Prior:

$$p(\beta_1, \dots, \beta_p) \propto 1$$

Posterior Distributions:

Posterior Distribution of Regression Coefficients:

$$p(\text{EQUIP,ECON,GORE,BUSH,BALLOTS}|\text{UNDERCOUNT (Y)}) \propto 1 \times \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(\text{UNDERCOUNT}_i - \beta_1 - \text{EQUIP}_2x_i - \text{ECON}_3x_i - \right. \\ \left. - \text{BUSH}_5x_i - \text{BALLOTS}_6x_i)^2\right\}$$

In the generic formed a posterior is defined in the form of:

$$p(\theta|y) \propto p(\theta) * p(y|\theta)$$

Within the Bayesian Analysis we also derived our posterior numerically (for undercount), created a larger more effective sample and computed HPD credibility intervals for all the variables in the model. We also used burn-in and thinning to control for stationarity and auto-correlation respectively.

### Model Diagnostics Summary

As mentioned above, our Stepwise model contains statistically significant variables, however, fails to hold some assumptions which throw its' ability to accurately model the (likely non-linear) data into question. The Bayesian Model held up all the assumptions and we successfully sampled, tuned our sample and produced posterior draws for all of our variables.

### Conclusion and Detailed Results

In the context of our problem, the result is clear. The level of income had a statistically significant effect on whether or not a location was more likely to be undercounted. This was surprisingly uncorrelated with any one type of a machine (at a significant level) which showed that economic status was a better predictor than voting machine. This is likely because in many instances the lowest performing voting machines are already in lower economic areas (resulting in co-linearity of these variables).

Overall, although this relationship was likely not very linear both models were able to successfully examine the relationship between the predictors and the predicted undercount.

### Discussions of Strengths and Weaknesses of the Models (Bayesian vs. Frequentist)

The strengths of the frequentist model are its' interpretability. We are able to directly examine the effects of coefficients and add and remove variables to examine different relationships.

The weaknesses are that it is limited to modeling linear relationships, of which are data likely is not.

The Bayesian Model creates a more effective data set, implementing stationarity and reducing auto-correlation. Its' main weakness is in interpretation. Though we have intervals and means (and standard errors) for an expanded data set we do not have that easily interpretable function that a reader can immediately understand and apply to make inference on new situations. This is especially a weakness in the social sciences within which researchers often would like to extrapolate models to explain more than one unique situation.

## Future Ideas

Detailed in paper!