# Final Exam

*Tapasvi Bansal*

*12/9/2017*

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
## Installing Packages and loading libraries
#install.packages("tidyverse")
#install.packages("pscl")
library(fiftystater)
library(pscl)
```

```
## Warning: package 'pscl' was built under R version 3.4.2
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```
library(ggplot2)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.4.2
```

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

```
## ✓ tibble  1.3.4     ✓ purrr   0.2.4
## ✓ tidyr   0.7.2     ✓ dplyr   0.7.4
## ✓ readr   1.1.1     ✓ stringr 1.2.0
## ✓ tibble  1.3.4     ✓ forcats 0.2.0
```

```
## Warning: package 'tidyr' was built under R version 3.4.2
```

```
## Warning: package 'purrr' was built under R version 3.4.2
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

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

```
library(dplyr)
library(doParallel)
```

```
## Warning: package 'doParallel' was built under R version 3.4.2
```

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

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

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

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

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

```
library(foreach)
```

## Problem 1: 2016 Election Results Data

## Problem 1, Question 1

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

```
######## Loading Data

USPE <- read.csv("US_County_Level_Presidential_Results_08-16.csv.bz2")
Countydata <- read_csv("county_data.csv.bz2")
```

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

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

```
#names(USPE)
#names(Countydata)
#View(USPE)
#View(Countydata)
```

```
####### Data Wrangling

Countydata$fips_code <- paste(Countydata$STATE, formatC(Countydata$COUNTY, width=3, flag="0"), sep="") #

Countydata <- Countydata %>% select(fips_code, everything())

Countydata$fips_code <- as.numeric(as.character(Countydata$fips_code))

Countydata <- Countydata %>% select(fips_code,SUMLEV, REGION, DIVISION,STNAME, CTYNAME,  grep("2012", na

USPE <- arrange(USPE, fips_code)

USPE <- USPE %>% select(fips_code, county , grep("2012", names(USPE)), grep("2016", names(USPE)))

#typeof(USPE$fips_code)
#typeof(Countydata$fips_code)



####### Merging datasets

ERdata <- inner_join(USPE, Countydata, by = "fips_code")

ERdata <- ERdata %>% select(fips_code,county,STNAME,REGION, DIVISION, CTYNAME, SUMLEV, everything())

ERdata <- subset(ERdata, select = c(1:3,8:47))
rownames(ERdata) <- NULL



####### Checking data excluded in the previous join

# anti_join(USPE, Countydata, by = "fips_code")
# View( anti_join(Countydata,USPE, by = "fips_code"))
# Countydata[which(Countydata$fips_code == '46113'),]
# USPE[which(USPE$fips_code == '46113'),]


####### Computing Percentage of Population that voted in individual counties, Voting Population Percentg

ERdata <- ERdata %>% mutate("vppercentage_2016" = total_2016/POPESTIMATE2016 * 100)
#typeof(ERdata$POPESTIMATE2016)

ERdata <- ERdata %>% mutate("demvpercentage_2016" = dem_2016/total_2016 * 100) %>% mutate("demvpercentag

#typeof(ERdata$demvpercentage_2016)



## Summarizing wins (major votes) for democrats and gop by state

ERdata <- ERdata %>% mutate(dem_win = ifelse(ERdata$dem_2016 > ERdata$gop_2016,1,0)) %>% mutate(gop_win
```

```
demW <- ERdata %>% select(STNAME, dem_win) %>% group_by(STNAME) %>% summarise(demwins = sum(dem_win))

gopW <- ERdata %>% select(STNAME, gop_win) %>% group_by(STNAME) %>% summarise(gopwins = sum(gop_win))

ERSdata <- inner_join(demW, gopW, by = "STNAME")

## Transforming columns for democrats, gop and others to 2 columns: party_name, votes

NERdata <- ERdata
NERdata <- NERdata %>% select(fips_code,county,democrats=dem_2016,
                              republicans=gop_2016,others=oth_2016,
                              total_votes=total_2016) %>%
  gather(party_name,votes,3:5) %>%
  arrange(fips_code)
```

==> As a part of tidying data, merged state and county codes to create the fips code in the county data. Also changed the type of fips code column to numeric for a further step, inner join.

==> Selected only the 2012 and 2016 year data from both the data set for analyis purpose. Removed the columns SUMLEV, REGION, DIVISION, CTYNAME from the final merged data frame as their purpose until now seems unnecessary for the further analysis.

## Problem 1, Question 2

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

==> The two datasets are US County Level Presidential Results and US County Census data. The US County Level Presidential Results data has the number of votes casted in total and individually in favour of both the Democratic Party and Republican Party (gop) county wise for the election years 2008, 2012 and 2016.

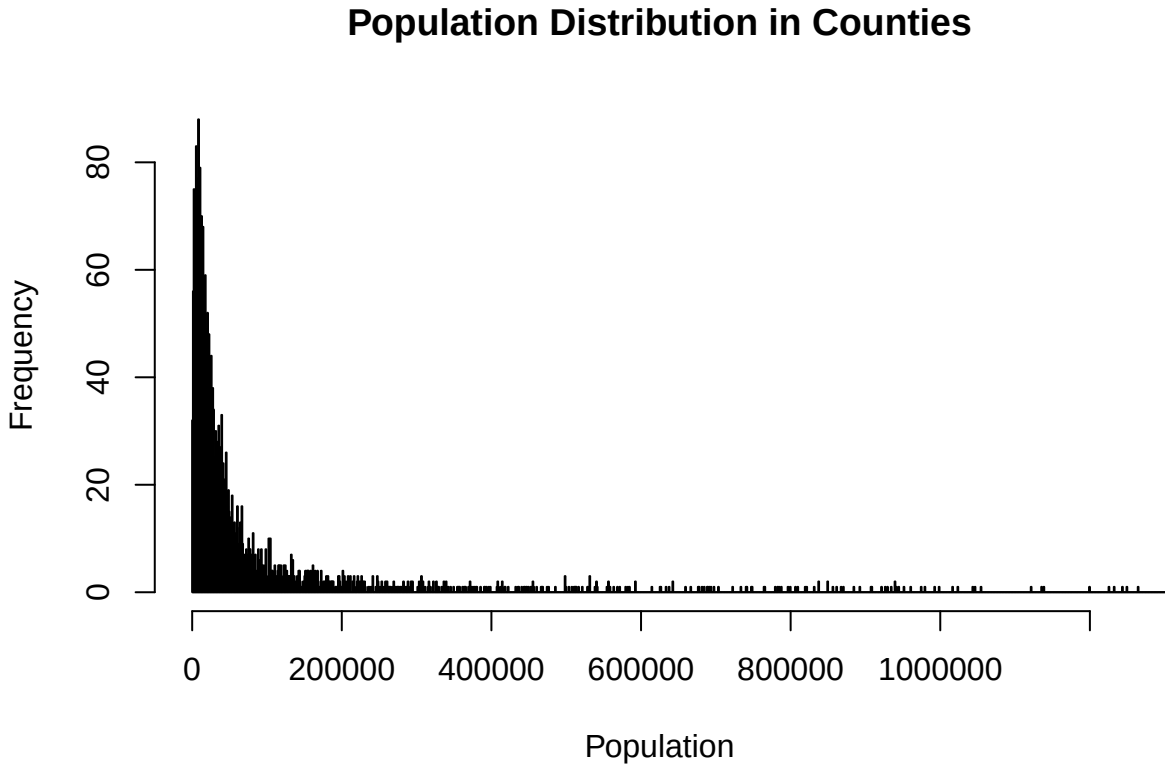
==> The US County Census data provides detailed numbers on the total population (estimate), births, birth rate, deaths, death rate, net migration, etc. county wise from 1st April 2010 to 30th June 2016.

==> Besides analyzing the percentage of population that casted votes out of total and for both the parties, additional analysis like effects of growth/decline of population , change in rate of immigration over the years on change in percentage of votes casted in individual counties.

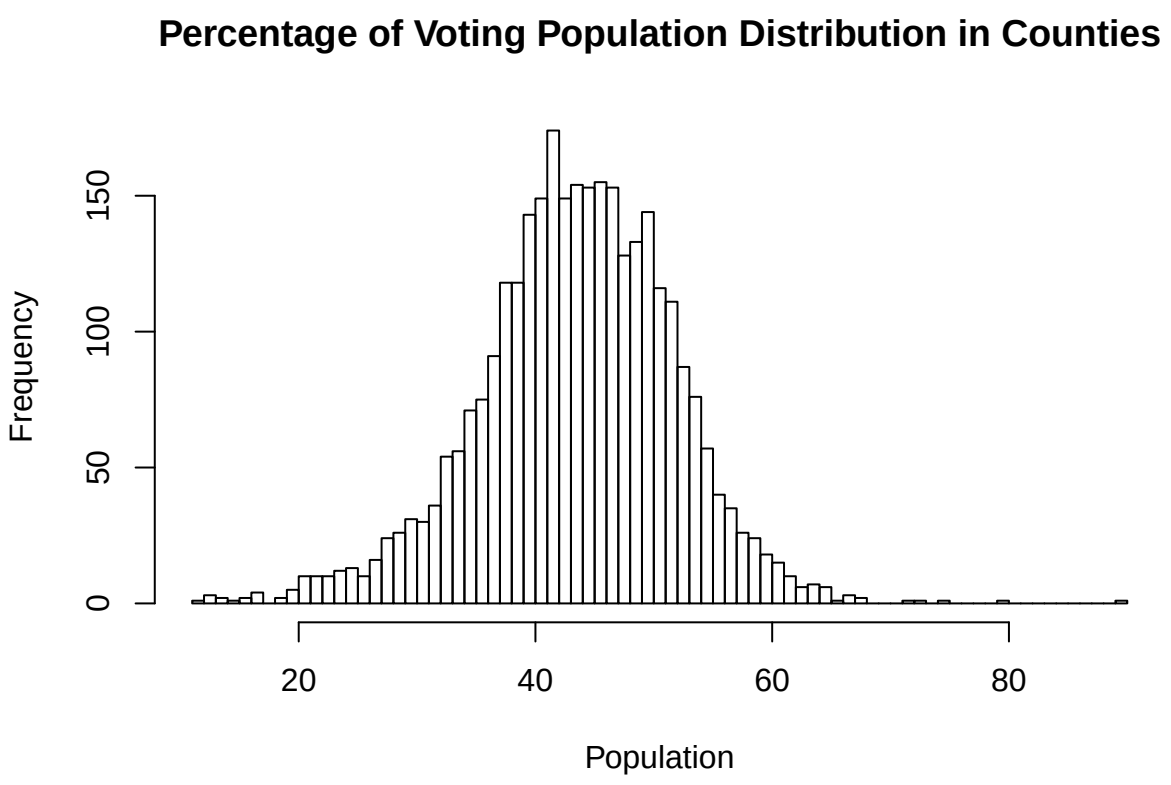
## Problem 1, Question 3

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

```
## Total Population Distribution
hist(x=ERdata$POPESTIMATE2016, breaks = 10000, xlim = c(0, 1250000), xlab = "Population", main = "Popula
```




```
## Percentage of Voting Population Distribution
hist(x=ERdata$vppercentage_2016, breaks = 100, xlab = "Population", main = "Percentage of Voting Popula
```




```
mean(ERdata$vppercentage_2016) - 3*sd(ERdata$vppercentage_2016)
```

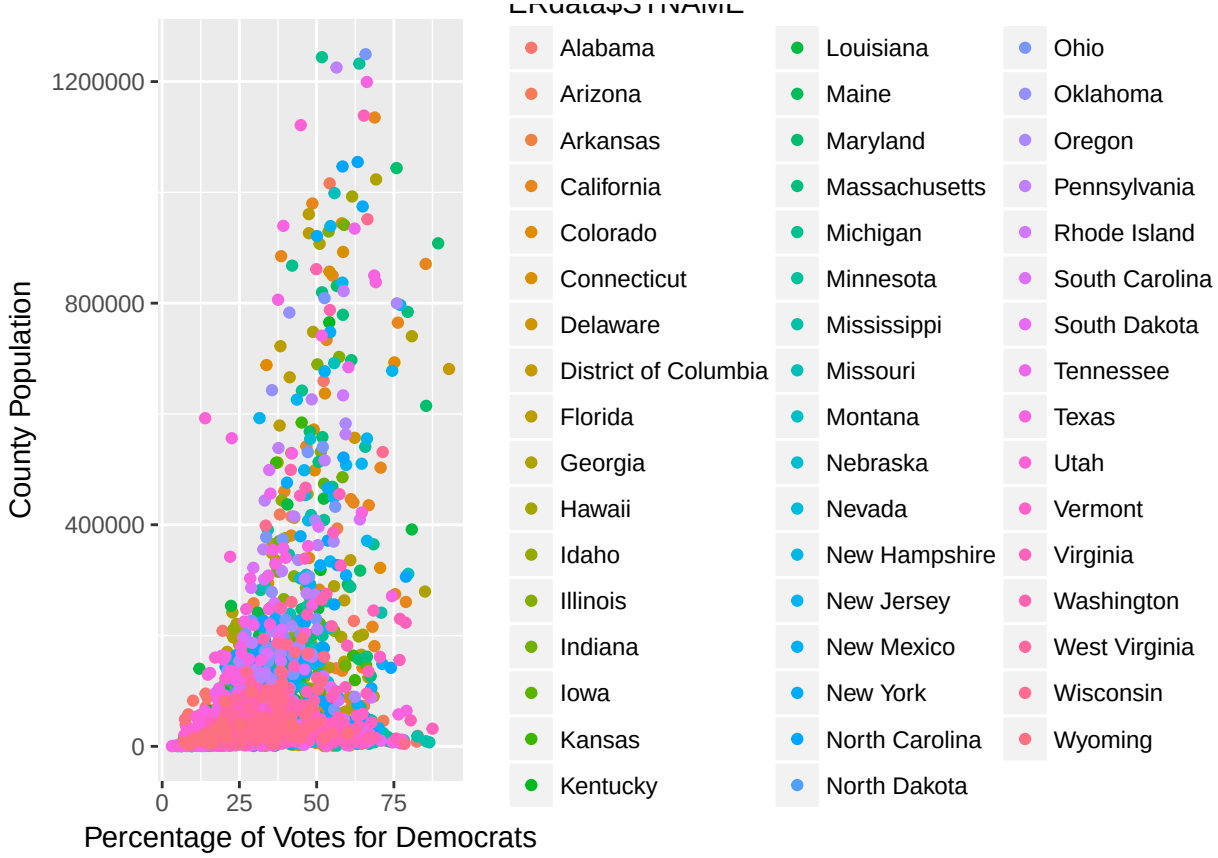
```
## [1] 18.69929
```

```
mean(ERdata$vppercentage_2016) + 3*sd(ERdata$vppercentage_2016)
```

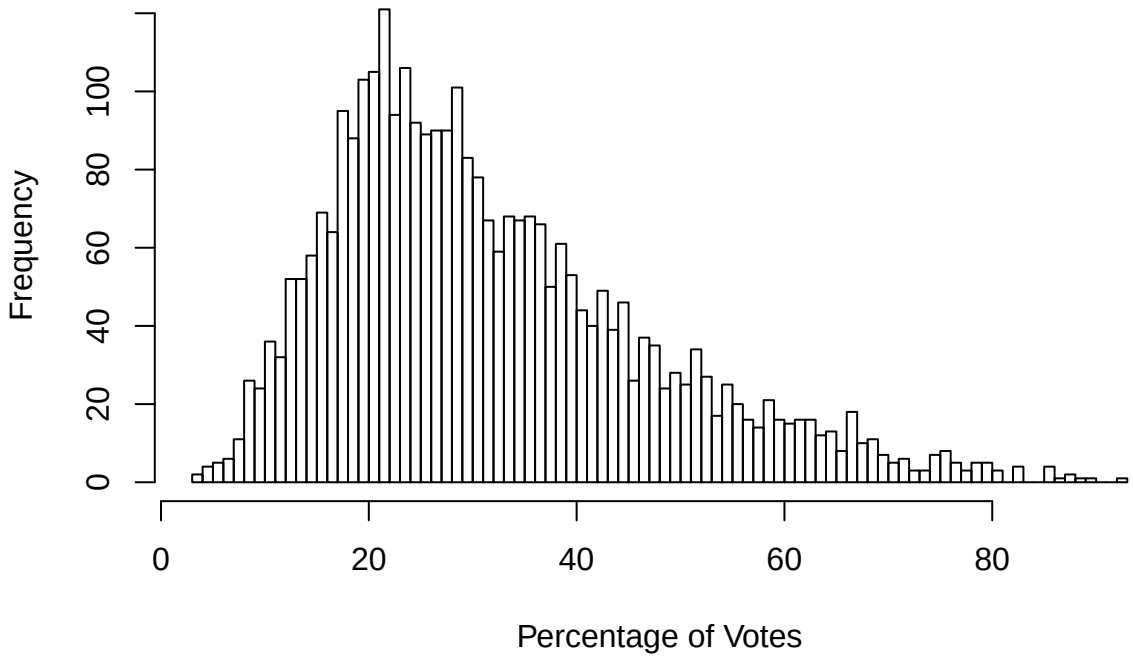
```
## [1] 68.2314
```

```
## Percentage of votes for democrats versus the county population
ggplot(ERdata) + geom_point(aes (x= demvpercentage_2016, y = POPESTIMATE2016, col= ERdata$STNAME)) + yl
```

```
## Warning: Removed 31 rows containing missing values (geom_point).
```

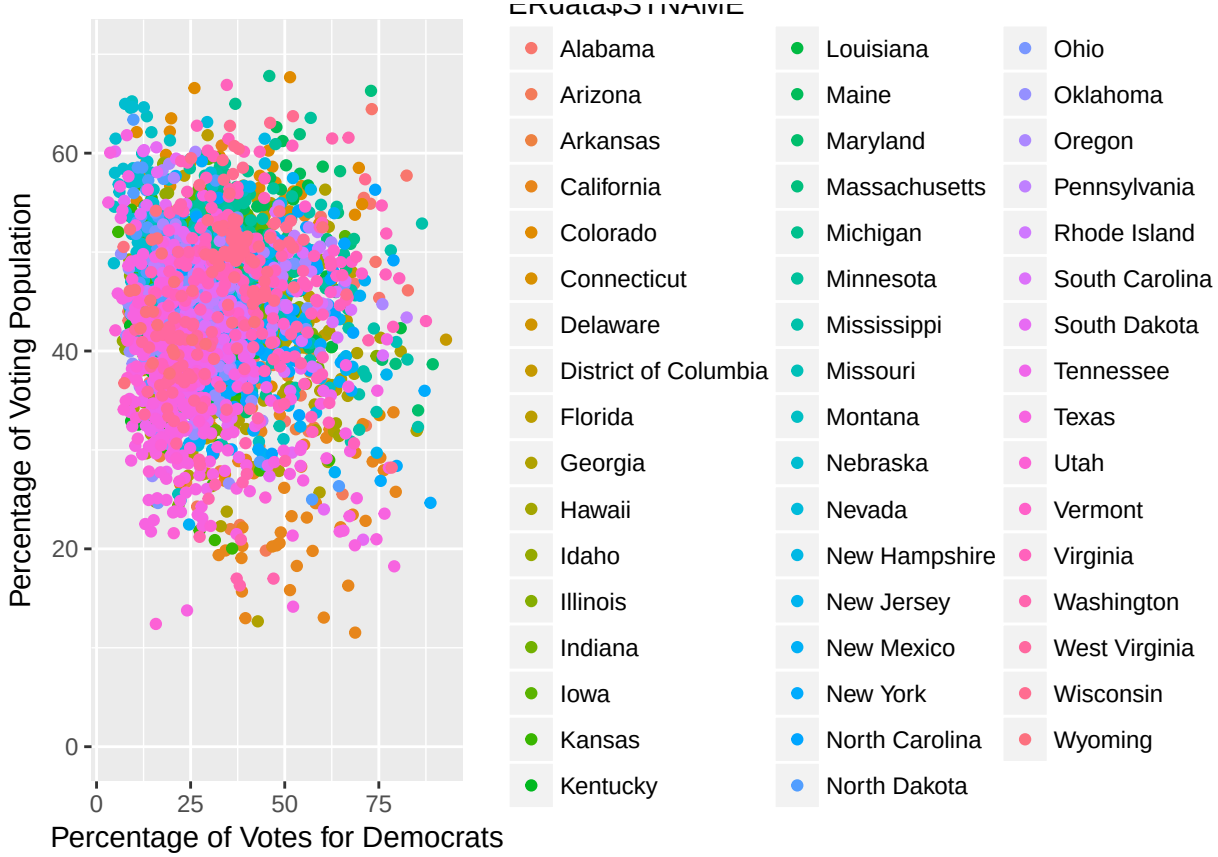



```
## Percentage of votes for democrats distribution
hist(x=ERdata$demvpercentage_2016, breaks = 100,  xlab = "Percentage of Votes", main = "Distribution of
```




```
## Percentage of votes for democrats versus the Percentage of Voting Population
ggplot(ERdata) + geom_point(aes (y= vppercentage_2016, x = demvpercentage_2016, col= ERdata$STNAME)) + y
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```




Conclusion:

==> The percentage of votes for democrats was appearing majorly between 15% to 50% in counties with population under 400000 in 2016. On further exploration, we also see that majority of counties has population under 400000.

==> The Percentage of Voting Population, i.e. people who voted out of the total population of the county is almost normally distributed.

==> It was insightful to know that Percentage of Voting Population between approx. 18% to 68% (major part, almost normally distributed), voted majorly between 15% to 50% for democrats. It can be said "People who practiced the right to vote more, voted less (to medium) for democrats."

## Problem 1, Question 4

4. Create a map of percentage of votes for democrats. Do your best to reflect the continuous percentage of votes, and the different population sizes across counties and keep county boundaries as well legible as you can. Mark state boundaries on the map. Explain what did you do, and what worked well, what did not work well.

Hint: there are many ways to map data in R. You may consider function ggplot::map_data that includes various maps, including US administrative boundaries. However, map_data counties do not include FIPS code. You may rely on merging data by state name and county name, given you a) convert your names to lower case, and b) remove the word " county" from the end of the names. This works for most of the counties, except for Lousiana where counties are called parish .

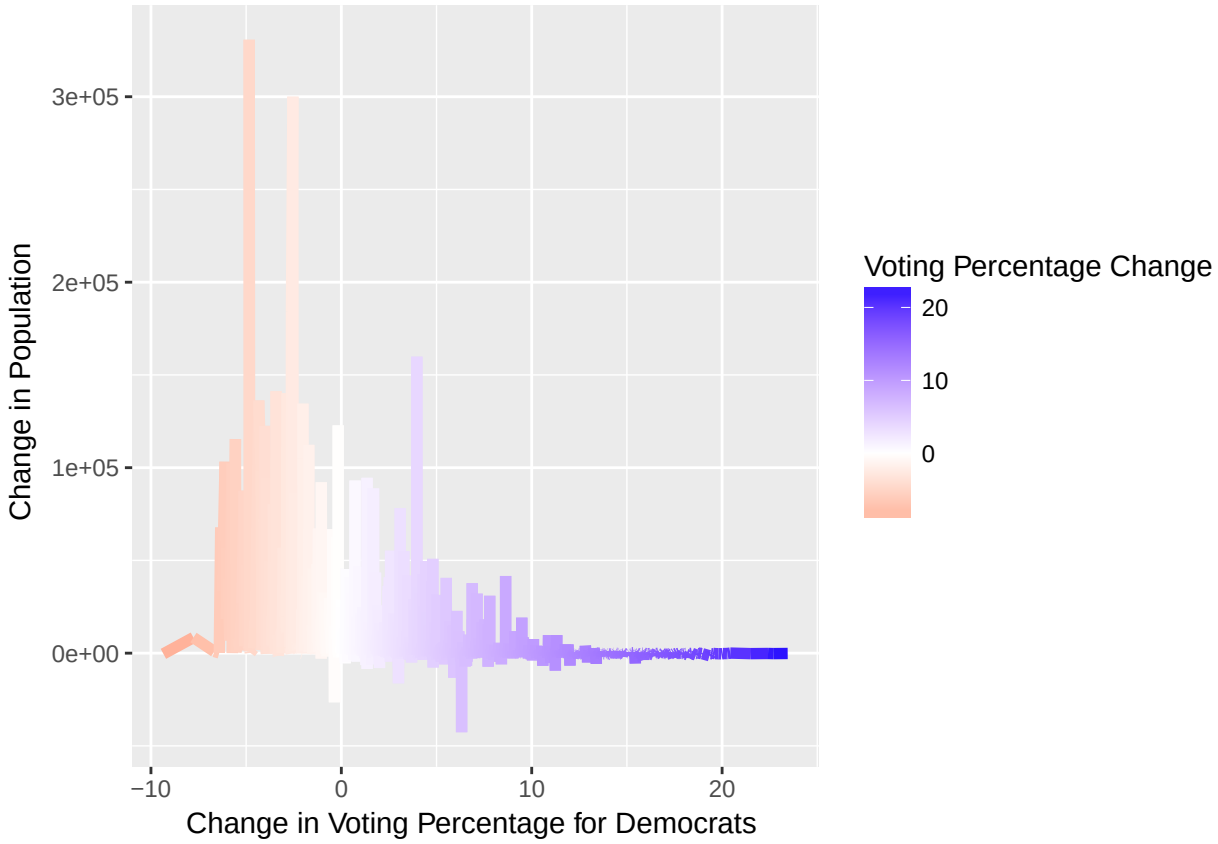
```
# library(maps)
# library(usmap)
# library(RColorBrewer)
```

```
# library(dplyr)
# library(magrittr)
# map_data("county")
#
# county_data <- map_data("county")
# countydatamap <- county_data %>% select(lat, long, group, order, state = "region", county = "subregio
# ERdatamap <- ERdata %>% select ("fips_code", "county", state = "STNAME", "demvpercentage_2016","POPES
#
# colors = c("#d0d1e6", "#a6bddb", "#74a9cf", "#2b8cbe", "#045a8d", "#034e7b")
# ERdatamap$colorBuckets <- as.numeric(cut(ERdatamap$demvpercentage_2016, c(0.0, 0.2, 0.4, 0.6, 0.8, 1.
# leg.txt <- c("< 20%", "20%-39%", "40%-59%", "60%-79%", "80%-99%", "100%")
#
# ERVmapdata <- left_join(county_data, ERdata, by = c("state", "county"))
```

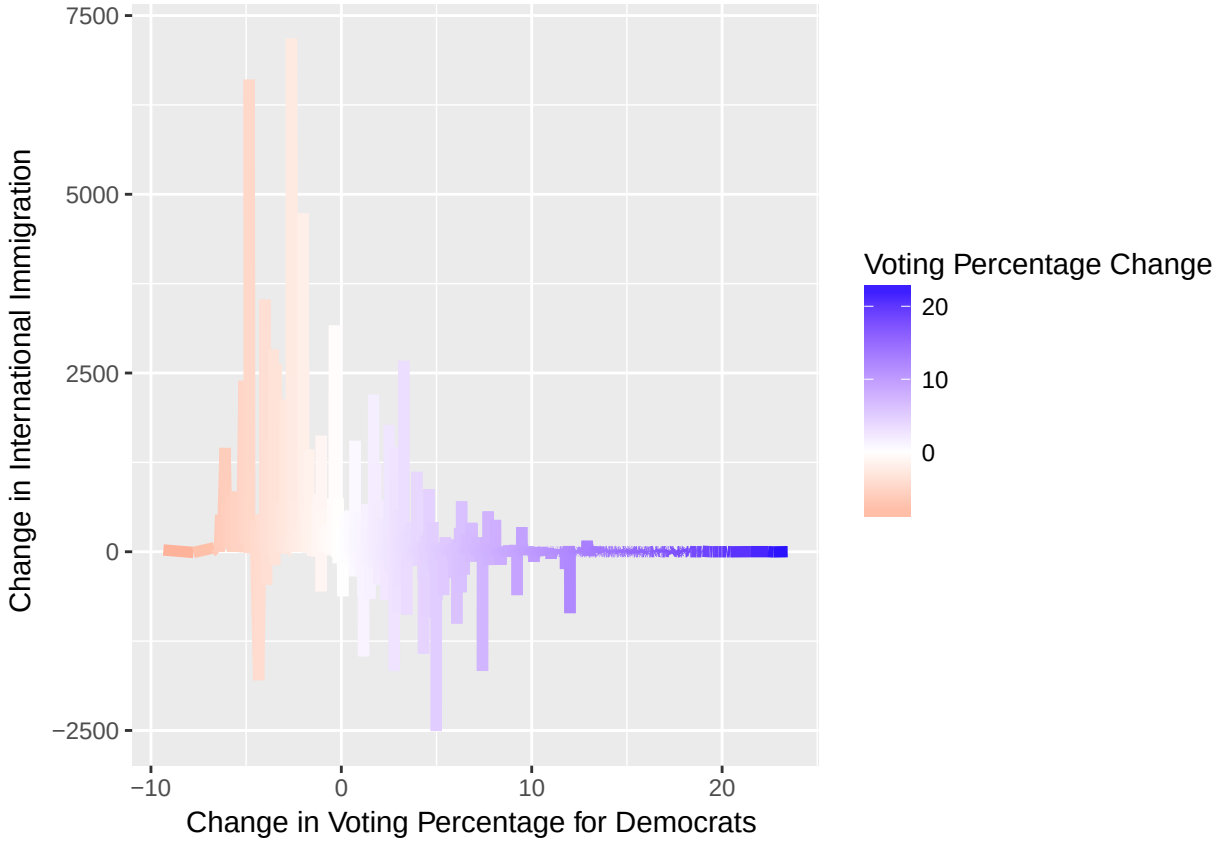
## Problem 1, Question 5

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

```
ggplot(ERdata %>% mutate(change_demvpercentage = demvpercentage_2012 - demvpercentage_2016) %>% mutate(
```



```
ggplot(ERdata %>% mutate(change_demvpercentage = demvpercentage_2012 - demvpercentage_2016) %>% mutate(
```




## Problem 2: 2016 Election Model (25pt)

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

## Problem 2, Question 1

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

```
names(ERdata)
```

```
##  [1] "fips_code"             "county"
##  [3] "STNAME"                "total_2012"
##  [5] "dem_2012"              "gop_2012"
##  [7] "oth_2012"              "total_2016"
##  [9] "dem_2016"              "gop_2016"
## [11] "oth_2016"              "POPESTIMATE2012"
## [13] "NPOPCHG_2012"          "BIRTHS2012"
## [15] "DEATHS2012"            "NATURALINC2012"
## [17] "INTERNATIONALMIG2012"  "DOMESTICMIG2012"
## [19] "NETMIG2012"            "RESIDUAL2012"
```

```
## [21] "GQESTIMATES2012"       "RBIRTH2012"
## [23] "RDEATH2012"            "RNATURALINC2012"
## [25] "RINTERNATIONALMIG2012" "RDOMESTICMIG2012"
## [27] "RNETMIG2012"           "POPESTIMATE2016"
## [29] "NPOPCHG_2016"          "BIRTHS2016"
## [31] "DEATHS2016"            "NATURALINC2016"
## [33] "INTERNATIONALMIG2016"  "DOMESTICMIG2016"
## [35] "NETMIG2016"            "RESIDUAL2016"
## [37] "GQESTIMATES2016"       "RBIRTH2016"
## [39] "RDEATH2016"            "RNATURALINC2016"
## [41] "RINTERNATIONALMIG2016" "RDOMESTICMIG2016"
## [43] "RNETMIG2016"           "vppercentage_2016"
## [45] "demvpercentage_2016"   "demvpercentage_2012"
## [47] "dem_win"               "gop_win"
```

==> Variables like Population Estimate (or change in population), Net migration ( or change in migration) can outrightly be considered as relevant to do analysis around voting patterns for 2016 elections (or votes for democrats).

==> The reason behind finding out such a corelation is because of the democrats forming the government for the term(s) before. Since they formed the government before, any change in population/migration during their government (2012-2016) can explain their influence on the US population and whether the people were inclined to vote or not to vote in favour for democrats (or gop).

## Problem 2, Question 2

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

```
#View(ERdata)

## Observing estimates for a binary logistic regression model and corresponding marginal effects

blrmodel <- mfx::logitmfx(dem_win ~ POPESTIMATE2016 + NETMIG2016 , data= ERdata)
```

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

```
blrmodel
```

```
## $mfxest
##                         dF/dx    Std. Err.         z        P>|z|
## POPESTIMATE2016  7.817787e-07 6.210476e-08 12.588064 2.456243e-36
## NETMIG2016      -2.932530e-05 4.500291e-06 -6.516311 7.205766e-11
##
## $fit
##
## Call:  glm(formula = formula, family = binomial(link = "logit"), data = data,
##     start = start, control = control, x = T)
##
## Coefficients:
##     (Intercept)  POPESTIMATE2016       NETMIG2016
##      -2.307e+00        6.167e-06       -2.313e-04
##
## Degrees of Freedom: 3110 Total (i.e. Null);  3108 Residual
## Null Deviance:       2696
## Residual Deviance: 2236  AIC: 2242
```

```
##
## $dcvar
## character(0)
##
## $call
## mfx::logitmfx(formula = dem_win ~ POPESTIMATE2016 + NETMIG2016,
##     data = ERdata)
##
## attr(,"class")
## [1] "logitmfx"
```

```
blrmodel1 <- mfx::logitmfx(dem_win ~ POPESTIMATE2016 + changepop , data= ERdata %>% mutate(changepop = P
```

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

```
blrmodel1
```

```
## $mfxest
##                         dF/dx    Std. Err.         z        P>|z|
## POPESTIMATE2016  8.272600e-07 7.023670e-08 11.778173 5.057772e-32
## changepop       -5.696062e-06 1.054201e-06 -5.403205 6.546059e-08
##
## $fit
##
## Call:  glm(formula = formula, family = binomial(link = "logit"), data = data,
##     start = start, control = control, x = T)
##
## Coefficients:
##     (Intercept)  POPESTIMATE2016        changepop
##      -2.317e+00        6.626e-06       -4.562e-05
##
## Degrees of Freedom: 3110 Total (i.e. Null);  3108 Residual
## Null Deviance:       2696
## Residual Deviance: 2257  AIC: 2263
##
## $dcvar
## character(0)
##
## $call
## mfx::logitmfx(formula = dem_win ~ POPESTIMATE2016 + changepop,
##     data = ERdata %>% mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
##         mutate(changemig = NETMIG2012 - NETMIG2016))
##
## attr(,"class")
## [1] "logitmfx"
```

```
## Estimating a logistic regression model

Vdmodel <- glm( dem_win ~ POPESTIMATE2016 + NETMIG2016 ,
                data= ERdata %>% mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>% mutate (chang
```

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

```
summary(Vdmodel)
```

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

```
## glm(formula = dem_win ~ POPESTIMATE2016 + NETMIG2016, family = binomial(link = "logit"),
##     data = ERdata %>% mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
##         mutate(changemig = NETMIG2012 - NETMIG2016))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.7983  -0.4906  -0.4572  -0.4411   2.3357
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -2.307e+00  6.637e-02 -34.766  < 2e-16 ***
## POPESTIMATE2016  6.167e-06  4.196e-07  14.699  < 2e-16 ***
## NETMIG2016      -2.313e-04  3.392e-05  -6.819 9.15e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2236.1  on 3108  degrees of freedom
## AIC: 2242.1
##
## Number of Fisher Scoring iterations: 6
```

```
lrp <- predict(Vdmodel, type="response") > 0.5
crt <- table(ERdata$dem_win,lrp)
diasum <- crt %>% diag() %>% sum()
PdtPercentage <- diasum/sum(crt)
PdtPercentage
```

```
## [1] 0.8736741
```

==> Marginal effects show the change in probability when the predictor or independent variable increases by one unit. The logistic regression models (blrmodel and blrmodel1) reflects the marginal effects. Here, in the case of Election Result data, we see the close "marginal" effects with variables like POPESTIMATE2016 & NETMIG2016 and POPESTIMATE2016 & changepop on the probability of voting democratic.

==>The logistic regression model predicted the data with 87.36% accuracy.

## Problem 2, Question 3

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

```
Vdmodel1 <- glm( dem_win ~ POPESTIMATE2016 + NETMIG2016 + changepop ,
                 data= ERdata %>%
                 mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
                 mutate(changemig = NETMIG2012 -NETMIG2016),family=binomial(link="logit"))
```

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

```
summary(Vdmodel1)
```

```
##
```

```
## Call:
## glm(formula = dem_win ~ POPESTIMATE2016 + NETMIG2016 + changepop,
##     family = binomial(link = "logit"), data = ERdata %>% mutate(changepop = POPESTIMATE2016 -
##         POPESTIMATE2012) %>% mutate(changemig = NETMIG2012 -
##         NETMIG2016))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.9330  -0.4897  -0.4579  -0.4426   2.2958
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -2.294e+00  6.705e-02 -34.219  < 2e-16 ***
## POPESTIMATE2016  5.729e-06  5.402e-07  10.606  < 2e-16 ***
## NETMIG2016      -3.013e-04  6.618e-05  -4.553 5.29e-06 ***
## changepop        2.078e-05  1.663e-05   1.250    0.211
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2234.5  on 3107  degrees of freedom
## AIC: 2242.5
##
## Number of Fisher Scoring iterations: 6
```

```
lrp1 <- predict(Vdmodel1, type="response") > 0.5
crt1 <- table(ERdata$dem_win,lrp1)
diasum1 <- crt1 %>% diag() %>% sum()
PdtPercentage1 <- diasum1/sum(crt1)
PdtPercentage1
```

```
## [1] 0.8756027
```

```
pR2(Vdmodel1)
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -1117.2385118 -1348.1457842   461.8145447     0.1712777     0.1379532
##          r2CU
##     0.2379890
```

==>The logistic regression model predicted the data with 87.56% accuracy.

==>McFadden value for the model is 0.17127

```
Vdmodel2 <- glm( dem_win ~ nonvoterpop + changepop + changemig,
                 data= ERdata %>%
                 mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
                 mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
                 mutate(changemig = NETMIG2012 -NETMIG2016),family=binomial(link="logit"))
```

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

```
summary(Vdmodel2)
```

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

```
## glm(formula = dem_win ~ nonvoterpop + changepop + changemig,
##     family = binomial(link = "logit"), data = ERdata %>% mutate(changepop = POPESTIMATE2016 -
##         POPESTIMATE2012) %>% mutate(nonvoterpop = POPESTIMATE2016 -
##         total_2016) %>% mutate(changemig = NETMIG2012 - NETMIG2016))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.9873  -0.4956  -0.4640  -0.4498   2.2575
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.258e+00  6.588e-02 -34.280  < 2e-16 ***
## nonvoterpop  1.000e-05  8.442e-07  11.847  < 2e-16 ***
## changepop   -2.878e-05  8.836e-06  -3.257  0.00113 **
## changemig    1.645e-04  4.560e-05   3.607  0.00031 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2268.6  on 3107  degrees of freedom
## AIC: 2276.6
##
## Number of Fisher Scoring iterations: 6
```

```
lrp2 <- predict(Vdmodel2, type="response") > 0.5
crt2 <- table(ERdata$dem_win,lrp2)
diasum2 <- crt2 %>% diag() %>% sum()
PdtPercentage2 <- diasum2/sum(crt2)
PdtPercentage2
```

```
## [1] 0.8733526
```

```
pR2(Vdmodel2)
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -1134.3130980 -1348.1457842   427.6653724     0.1586124     0.1284384
##          r2CU
##     0.2215747
```

==>The logistic regression model predicted the data with 87.33% accuracy.

==>McFadden value for the model is 0.15861

```
Vdmodel3 <- glm( dem_win ~ nonvoterpop + POPESTIMATE2016 + changepop,
                 data= ERdata %>%
                 mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
                 mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
                 mutate(changemig = NETMIG2012 -NETMIG2016),family=binomial(link="logit"))
```

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

```
summary(Vdmodel3)
```

```
##
## Call:
## glm(formula = dem_win ~ nonvoterpop + POPESTIMATE2016 + changepop,
```

```
##     family = binomial(link = "logit"), data = ERdata %>% mutate(changepop = POPESTIMATE2016 -
##         POPESTIMATE2012) %>% mutate(nonvoterpop = POPESTIMATE2016 -
##         total_2016) %>% mutate(changemig = NETMIG2012 - NETMIG2016))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.2559  -0.4932  -0.4567  -0.4389   2.4074
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -2.329e+00  6.725e-02 -34.627  < 2e-16 ***
## nonvoterpop     -6.413e-06  2.982e-06  -2.150   0.0315 *
## POPESTIMATE2016  1.045e-05  1.877e-06   5.567 2.59e-08 ***
## changepop       -4.481e-05  8.317e-06  -5.389 7.10e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2253.0  on 3107  degrees of freedom
## AIC: 2261
##
## Number of Fisher Scoring iterations: 6
```

```
lrp3 <- predict(Vdmodel3, type="response") > 0.5
crt3 <- table(ERdata$dem_win,lrp3)
diasum3 <- crt3 %>% diag() %>% sum()
PdtPercentage3 <- diasum3/sum(crt3)
PdtPercentage3
```

```
## [1] 0.8717454
```

```
pR2(Vdmodel3)
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -1126.5186730 -1348.1457842   443.2542224     0.1643940     0.1327948
##          r2CU
##     0.2290901
```

==>The logistic regression model predicted the data with 87.174% accuracy.

==>McFadden value for the model is 0.16439

```
Vdmodel4 <- glm( dem_win ~ nonvoterpop + POPESTIMATE2016 + NPOPCHG_2016 +
                   changepop + RINTERNATIONALMIG2016,data=
                   ERdata %>%
                   mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
                   mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
                   mutate(changemig = NETMIG2012 -NETMIG2016),family=binomial(link="logit"))
```

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

```
summary(Vdmodel4)
```

```
##
## Call:
## glm(formula = dem_win ~ nonvoterpop + POPESTIMATE2016 + NPOPCHG_2016 +
```

```
##     changepop + RINTERNATIONALMIG2016, family = binomial(link = "logit"),
##     data = ERdata %>% mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
##         mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
##         mutate(changemig = NETMIG2012 - NETMIG2016))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8817  -0.4740  -0.4044  -0.3631   2.3847
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.737e+00  8.230e-02 -33.254  < 2e-16 ***
## nonvoterpop           -5.386e-06  2.702e-06  -1.993   0.0462 *
## POPESTIMATE2016        8.511e-06  1.741e-06   4.887 1.02e-06 ***
## NPOPCHG_2016          -2.169e-04  9.068e-05  -2.392   0.0167 *
## changepop              1.098e-05  2.583e-05   0.425   0.6708
## RINTERNATIONALMIG2016  3.993e-01  3.554e-02  11.233  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2094.5  on 3105  degrees of freedom
## AIC: 2106.5
##
## Number of Fisher Scoring iterations: 6
```

```
lrp4 <- predict(Vdmodel4, type="response") > 0.5
crt4 <- table(ERdata$dem_win,lrp4)
diasum4 <- crt4 %>% diag() %>% sum()
PdtPercentage4 <- diasum4/sum(crt4)
PdtPercentage4
```

```
## [1] 0.8804243
```

```
pR2(Vdmodel4)
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -1047.2306997 -1348.1457842   601.8301689     0.2232066     0.1758909
##          r2CU
##     0.3034370
```

==>The logistic regression model predicted the data with 88.042% accuracy.

==>McFadden value for the model is 0.22320

```
Vdmodel5 <- glm( dem_win ~ nonvoterpop + POPESTIMATE2016 + NETMIG2016 +
                   RINTERNATIONALMIG2016  + changepop, data=
                   ERdata %>%
                   mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
                   mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
                   mutate(changemig = NETMIG2012 -NETMIG2016),family=binomial(link="logit"))
```

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

```
summary(Vdmodel5)
```

```
##
## Call:
## glm(formula = dem_win ~ nonvoterpop + POPESTIMATE2016 + NETMIG2016 +
##     RINTERNATIONALMIG2016 + changepop, family = binomial(link = "logit"),
##     data = ERdata %>% mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
##         mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
##         mutate(changemig = NETMIG2012 - NETMIG2016))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8397  -0.4707  -0.4032  -0.3648   2.3734
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.719e+00  8.232e-02 -33.024  < 2e-16 ***
## nonvoterpop           -1.027e-05  2.827e-06  -3.631 0.000283 ***
## POPESTIMATE2016        1.073e-05  1.765e-06   6.076 1.24e-09 ***
## NETMIG2016            -2.846e-04  6.709e-05  -4.242 2.22e-05 ***
## RINTERNATIONALMIG2016  3.851e-01  3.553e-02  10.839  < 2e-16 ***
## changepop              1.672e-05  1.728e-05   0.968 0.333134
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2081.1  on 3105  degrees of freedom
## AIC: 2093.1
##
## Number of Fisher Scoring iterations: 6
```

```
lrp5 <- predict(Vdmodel5, type="response") > 0.5
crt5 <- table(ERdata$dem_win,lrp5)
diasum5 <- crt5 %>% diag() %>% sum()
PdtPercentage5 <- diasum5/sum(crt5)
PdtPercentage5
```

```
## [1] 0.8826744
```

```
pR2(Vdmodel5)
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -1040.5557866 -1348.1457842   615.1799952     0.2281578     0.1794197
##          r2CU
##     0.3095247
```

==>The logistic regression model predicted the data with 88.267% accuracy.

==>McFadden value for the model is 0.22815

```
Vdmodel6 <- glm( dem_win ~ nonvoterpop + changemig + changepop + POPESTIMATE2016 +
                   NETMIG2016 + NPOPCHG_2016 + RINTERNATIONALMIG2016 ,
                   data= ERdata %>%
                   mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
                   mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
```

```
                  mutate(changemig = NETMIG2012 -NETMIG2016),family=binomial(link="logit"))
```

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

```
summary(Vdmodel6)
```

```
##
## Call:
## glm(formula = dem_win ~ nonvoterpop + changemig + changepop +
##     POPESTIMATE2016 + NETMIG2016 + NPOPCHG_2016 + RINTERNATIONALMIG2016,
##     family = binomial(link = "logit"), data = ERdata %>% mutate(changepop = POPESTIMATE2016 -
##         POPESTIMATE2012) %>% mutate(nonvoterpop = POPESTIMATE2016 -
##         total_2016) %>% mutate(changemig = NETMIG2012 - NETMIG2016))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7765  -0.4664  -0.4022  -0.3629   2.3740
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.727e+00  8.269e-02 -32.983  < 2e-16 ***
## nonvoterpop           -1.369e-05  3.115e-06  -4.395 1.11e-05 ***
## changemig             -9.931e-05  7.028e-05  -1.413    0.158
## changepop             -4.750e-07  3.385e-05  -0.014    0.989
## POPESTIMATE2016        1.226e-05  1.860e-06   6.591 4.36e-11 ***
## NETMIG2016            -5.463e-04  1.239e-04  -4.407 1.05e-05 ***
## NPOPCHG_2016           2.582e-04  1.649e-04   1.566    0.117
## RINTERNATIONALMIG2016  3.882e-01  3.567e-02  10.885  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2074.4  on 3103  degrees of freedom
## AIC: 2090.4
##
## Number of Fisher Scoring iterations: 6
```

```
lrp6 <- predict(Vdmodel5, type="response") > 0.5
crt6 <- table(ERdata$dem_win,lrp6)
diasum6 <- crt6 %>% diag() %>% sum()
PdtPercentage6 <- diasum6/sum(crt6)
PdtPercentage6
```

```
## [1] 0.8826744
```

```
pR2(Vdmodel6)
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -1037.2248633 -1348.1457842   621.8418418     0.2306286     0.1811750
##          r2CU
##     0.3125528
```

==>The logistic regression model predicted the data with 88.267% accuracy.

==>McFadden value for the model is 0.2306285

==> The last two logistic regression model (Vdmdodel5 and Vdmodel6) predicted the data with the highest accuracy, 88.267%. However, the last model (Vdmodel6) McFadden value's is the highest, 0.23062. ==> The best model is "Vdmodel6". With:

1) "nonvoterpop" - Non-Voter population
2) "changemig" - Numeric Change in net migration from 2012 to 2016
3) "changepop" - Numeric Change in resident total population estimate from 2012 to 2016
4) "POPESTIMATE2016" - 7/1/2016 resident total population estimate
5) "NETMIG2016" - Net migration in period 7/1/2015 to 6/30/2016
6) "NPOPCHG_2016" - Numeric change in resident total population 7/1/2015 to 7/1/2016
7) "RINTERNATIONALMIG2016 - Net international migration rate in period 7/1/2015 to 6/30/2016

….as the set of continuous predictor variables predicting the binary outcome, probability of voting democratic (the probability that a county voted for democrats in 2016 elections / the probability that democrats received more votes than GOP)

## Problem 2, Question 4

4. Explain the meaning of statistical significance. What does it mean that an estimated coe cient is statistically signi cant (at 5% con dence level)?

==> Statistical Significance is the likelihood of a relationship between variables that is more than just a random chance. If we observe an estimate of the coefficient more than two standard deviations away from zero, then we have reason to think that the Null Hypothesis is very unlikely. If we reject the null hypothesis with 95% confidence, then we typically say that our independent variable has a statistically significant effect on our dependent variable.

## Problem 2, Question 5

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

==> Preferred Model: The last (or 6th) Model, Vdmodel6.

==> Results that are statistically significant are:

=>"nonvoterpop", "POPESTIMATE2016", "NETMIG2016", "RINTERNATIONALMIG2016""

## Problem 2, Question 6

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

```
sd(ERdata$dem_win)
```

```
## [1] 0.363122
```

In logistic regression, the odds ratio represents the constant effect of a predictor variable (X), on the likelihood that one outcome (out of binary outcome) will occur(Y). In regression models, we often want a measure of the unique effect of each X on Y to observe that constant effect.

The z-value in logistic regression is the coefficient(predictor variable) estimate divided by corresponding standard error. The significance of this statistic is based on the Z distribution, given by the P Value column, so the coefficient(predictor variable) with small p-values are the more "significant.

==> HENCE, predictor variable with higher absolute z-value are strong predictors.

(((( FROM Problem 2, Question 2 Explanation:

==> Marginal effects show the change in probability when the predictor or independent variable increases by one unit. The logistic regression models (blrmodel and blrmodel1) reflects the marginal effects. Here, in the case of Election Result data, we see the close "marginal" effects with variables like POPESTIMATE2016 & NETMIG2016 and POPESTIMATE2016 & changepop on the probability of voting democratic.

))))

## Problem 3: Simulate the E ect of Additional Random Coe cients (25pt)

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

## Problem 3, Question 1

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

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

```
rnumbers <- rpois(nrow(ERdata),lambda=1000)
#typeof(rnumbers)
```

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

```
LRVdmodel <- glm( dem_win ~ nonvoterpop + changemig + changepop + POPESTIMATE2016 +
                    NETMIG2016 + NPOPCHG_2016 + RINTERNATIONALMIG2016 + rnumbers,
                    data= ERdata %>%
                    mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
                    mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
                    mutate(changemig = NETMIG2012 -NETMIG2016),family=binomial(link="logit"))
```

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

```
summary(LRVdmodel)
```

```
##
## Call:
## glm(formula = dem_win ~ nonvoterpop + changemig + changepop +
##     POPESTIMATE2016 + NETMIG2016 + NPOPCHG_2016 + RINTERNATIONALMIG2016 +
##     rnumbers, family = binomial(link = "logit"), data = ERdata %>%
##     mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
##     mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>% mutate(changemig = NETMIG2012 -
##     NETMIG2016))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7780  -0.4690  -0.4017  -0.3628   2.4057
##
## Coefficients:
```

```
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.056e+00  1.842e+00  -0.574    0.566
## nonvoterpop           -1.364e-05  3.106e-06  -4.390 1.13e-05 ***
## changemig             -1.029e-04  7.045e-05  -1.460    0.144
## changepop             -1.113e-07  3.390e-05  -0.003    0.997
## POPESTIMATE2016        1.225e-05  1.855e-06   6.604 3.99e-11 ***
## NETMIG2016            -5.446e-04  1.240e-04  -4.391 1.13e-05 ***
## NPOPCHG_2016           2.536e-04  1.650e-04   1.537    0.124
## RINTERNATIONALMIG2016  3.898e-01  3.569e-02  10.922  < 2e-16 ***
## rnumbers              -1.676e-03  1.847e-03  -0.908    0.364
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2696.3  on 3110  degrees of freedom
## Residual deviance: 2073.6  on 3102  degrees of freedom
## AIC: 2091.6
##
## Number of Fisher Scoring iterations: 6
```

```
# rlrp <- predict(LRVdmodel, type="response") > 0.5
# rcrt <- table(ERdata$dem_win,rlrp)
# rdiasum <- rcrt %>% diag() %>% sum()
# rPdtPercentage <- rdiasum/sum(rcrt)
# rPdtPercentage
# pR2(LRVdmodel)
```

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

```
rnocoeff <- coef(LRVdmodel)['rnumbers']
print(rnocoeff)
```

```
##     rnumbers
## -0.001676223
```

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

```
##### Implementing through Parallel Process because the sequential processing was failing..
##### .. due to lack of computing power (majority of times)

### The following code works fine!!

# R <- 1000
# coefficientlist <- c()
# for(i in 1:R)
# {
#   rnumbers1 <- rpois(nrow(ERdata),lambda=100)
#   LRVdmodel1 <-  glm( dem_win ~ nonvoterpop + changemig + changepop + POPESTIMATE2016 +
#                     NETMIG2016 + NPOPCHG_2016 + RINTERNATIONALMIG2016 + rnumbers1,
#                     data= ERdata %>%
#                     mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
#                     mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
```

```
#                     mutate(changemig = NETMIG2012 -NETMIG2016),family=binomial(link="logit"))
#
#   rnocoeff1 <- coef(LRVdmodel1)['rnumbers1']
#   coefficientlist <- append(coefficientlist,rnocoeff1)
# }
# #print (coefficientlist)
# length (coefficientlist)

######## Implementing through Parallel Process because the function was failing normally
######## This following code is used for the solution:

registerDoParallel(cores=20)
R <- 1000
newcoefficientlist <-foreach(i <- 1:R, .combine=append)%dopar%
{
  rnumbers1 <- rpois(nrow(ERdata),lambda=100)
  LRVdmodel1 <-  glm( dem_win ~ nonvoterpop + changemig + changepop + POPESTIMATE2016 +
                    NETMIG2016 + NPOPCHG_2016 + RINTERNATIONALMIG2016 + rnumbers1,
                    data= ERdata %>%
                    mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
                    mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
                    mutate(changemig = NETMIG2012 -NETMIG2016),family=binomial(link="logit"))

  rnocoeff1 <- coef(LRVdmodel1)['rnumbers1']

}

length (newcoefficientlist)
```

```
## [1] 1000
```

## Problem 3, Question 2

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

```
meancf <- mean(newcoefficientlist)
sdcf <- sd(newcoefficientlist)
meancf
```

```
## [1] 8.32023e-05
```

```
sdcf
```

```
## [1] 0.005851846
```

==> Sample mean of the estimated coefficients is 0.0001162082

==> Sample standard deviation of the estimated coefficients is 0.005790411

..

## Problem 3, Question 3

3. Find the 95% con dence interval of the coe cient based on your simulations.

```
quantile(newcoefficientlist,0.025)
```

```
##        2.5%
## -0.01092074
```

```
quantile(newcoefficientlist,0.975)
```

```
##      97.5%
## 0.01188913
```

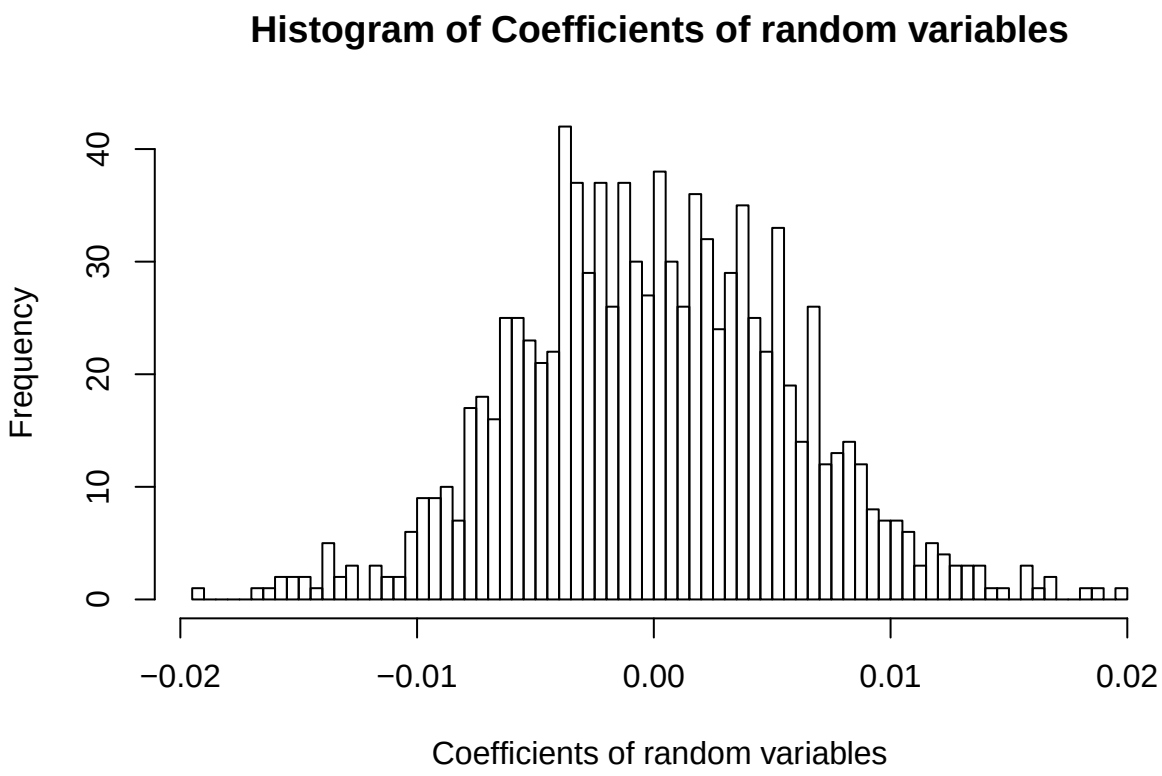
==> The 95% confidence interval of the coefficents : (-0.01129169 , 0.01187347 )

..

### Problem 3, Question 4

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

```
hist(newcoefficientlist, breaks = 100, xlab = "Coefficients of random variables", main = paste("Histogra
```




### Problem 3, Question 5

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

```
pstvalue <- meancf + 1.96 * sdcf
ngtvalue <- meancf - 1.96 * sdcf
print(pstvalue)
```

```
## [1] 0.01155282
```

```
print(ngtvalue)
```

```
## [1] -0.01138642
```

==> The 95% theoretical confidence interval of the coefficents are: ( 0.01146541 ,-0.011233).

..

## Problem 3, Question 6

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

```
######### With Sequential processing
######### The following code works fine!!

#
# start_time <- Sys.time()
# R <- 1000
# ncoefficientlist <- c()
# for(i in 1:R)
# {
#   rnumbers1 <- rpois(nrow(ERdata),lambda=100)
#   LRVdmodel1 <-  glm( dem_win ~ nonvoterpop + changemig + changepop + POPESTIMATE2016 +
#                     NETMIG2016 + NPOPCHG_2016 + RINTERNATIONALMIG2016 + rnumbers1,
#                     data= ERdata %>%
#                     mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
#                     mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
#                     mutate(changemig = NETMIG2012 -NETMIG2016),family=binomial(link="logit"))
#
#   rnocoeff1 <- coef(LRVdmodel1)['rnumbers1']
#   ncoefficientlist <- append(coefficientlist,rnocoeff1)
# }
#
#
# end_time <- Sys.time()
# print(end_time-start_time)    # Time difference of 1.261015 mins



######## With Parallel Processing

registerDoParallel(cores=20)
Prllstart_time <- Sys.time()
R <- 1000
pcoefficientlist <-foreach(i <- 1:R, .combine=append)%dopar%
{
  rnumbers1 <- rpois(nrow(ERdata),lambda=100)
  LRVdmodel1 <-  glm( dem_win ~ nonvoterpop + changemig + changepop + POPESTIMATE2016 +
                    NETMIG2016 + NPOPCHG_2016 + RINTERNATIONALMIG2016 + rnumbers1,
                    data= ERdata %>%
                    mutate(changepop = POPESTIMATE2016 - POPESTIMATE2012) %>%
                    mutate(nonvoterpop = POPESTIMATE2016 - total_2016) %>%
                    mutate(changemig = NETMIG2012 -NETMIG2016),family=binomial(link="logit"))
```

```
  rnocoeff1 <- coef(LRVdmodel1)['rnumbers1']

}

Prllend_time <- Sys.time()
print(Prllend_time - Prllstart_time)  # Time difference of 46.3649 secs
```

```
## Time difference of 44.14329 secs
```

```
length(pcoefficientlist)
```

```
## [1] 1000
```

==> Sequential Processing Time: 1.261015 mins

==> Parallel Processing Tiem: 46.3649 secs

==> Parallel processing performed better than Sequential Processing, by processing approximately 40 Seconds faster.

..

## Problem 4: Coin Tossing Game (25p)

You are ordered to participate in a coing-tossing game. The rules are following: the coin is tossed until tail comes up. If tail comes up in the first flip (T), you receive $1. If a single head pops up before the tail (H,T), you get $2. If two heads pop up (H,H,T), you receive $4. If three heads appear before the first tail (H,H,H,T), you get $8. And so forth, so if the realized sequence is n heads and thereafter a tail (H,H,...,H,T), you will receive $2n. The tosses are independent.
n

## Problem 4, Question 1

1. Assume the coin fair.

(a) Compute your expected payoff in this game.

Value of dollar earned until tail is observed on coin toss/flip:

==> Value if first flip (T) = 1

==> Value if second flip (H,T) = 2

==> Value if thrid flip (H,H,T) = 4

==> Value if fourth flip (H,H,H,T) = 8

.. so on —- (1)

==> The associated probablity (p) with the events:

==> p for first flip = 1/2

==> p for second flip = 1/4 (as p is independent events)

==> p for third flip = 1/8

==> p for fourth flip = 1/16

.. so on —- (2)

==> From (1) and (2), the expected value will be:

Expected Value = [1 * (1/2)^1] + [2^1 * (1/2)^2] + [2^2 * (1/2)^3] + 2^3 * (1/2)^4] + ….

Expected Value = [2^0 * (1/2)^1] + [2^1 * (1/2)^2] + [2^2 * (1/2)^3] + 2^3 * (1/2)^4] + ….

Expected Value = [2^-1] + [2^-1] + [2^-1] + [2^-1] + ….

Expected Value = infinite (oxo)

==> However, in practicality if we observe, the prize doubles up, but the value of the prize doesn't double up, rather increases by a factor less than 2.

(b) How much would you actually be willing to pay for the participation? Note: we are asking your judgement/opinion here, not a computed value.

==> None. Like any game of chance (gamble), this game can entice the player to continue forever until the player losses all.

==> Also, if the above question means paying for the participation just once, I'll be happy to pay $1 (assuming that if tail comes first, I'll get my money back), just to have some thrill.

## Problem 4, Question 2

2. Assume such a game is played 10 times and the outcomes are: twice (T); three times (H,T); once (H,H,T); twice (H,H,H,T), once (H,H,H,H,T); and once (H,H,H,H,H,H,T). This is your data. Your task is to nd the Maximum Likelihood estimator of p the probability of receiving a head.

(a) Write down the probability to receive n heads and a tail.

Probablility (p) = Probablity of heads (Ph) * Probablity of a tail (Pt)

==> p = Ph^n * Pt — (Ph^n as Ph is independent events)

.. it can also be written as :

in case of fair coin Pt = (1- Ph)

==> p = Ph^n * (1- Ph) – ( = L(Ph|n), Likelihood function )

==> p = (1/2)^n * 1/2 = (1/2)^n+1 – (if Ph =1/2)

(b) Write down the probability to receive all 10 outcomes listed above.

1) Outcome is: twice (T) = Proabality of Tails * Proabality of Tails

==> Probability = 1/2 * 1/2 = 1/4

2) Outcome is: three times (H,T) = (Proabality of Heads * Proabality of Tails)^3

==> Probability = (1/2*1/2) ^3 = 1/64

3) Outcome is: once (H,H,T);

==> Probability = (1/2)^2 * 1/2 = 1/8

4) Outcome is: twice (H,H,H,T);

==> Probability = ((1/2)^3 * 1/2)^2 = 1/256

5) Outcome is: once (H,H,H,H,T);

==> Probability = ((1/2)^4 * 1/2) = 1/32

6) Outcome is: once (H,H,H,H,H,H,T)

==> Probability = ((1/2)^6 * 1/2) = 1/128

..

(c) Write the log-likelihood function of this data as a function of the parameter.

==> Likelihood function, L(Ph|n) = Ph^n * (1- Ph)

==> log (L(Ph|n)) = log(Ph^n * (1- Ph))

==> l(Ph|n) = (n) log Ph + log (1-Ph)

..

(d) Analytically solve this log-likelihood for the optimal probabilty p .

==> Analytic solution: set derivative to 0

==> 0 = n/Ph - 1/(1-Ph)

==> n/Ph = 1/(1-Ph)

==> n = Ph/ (1-Ph)

==> Ph = n/(n+1)

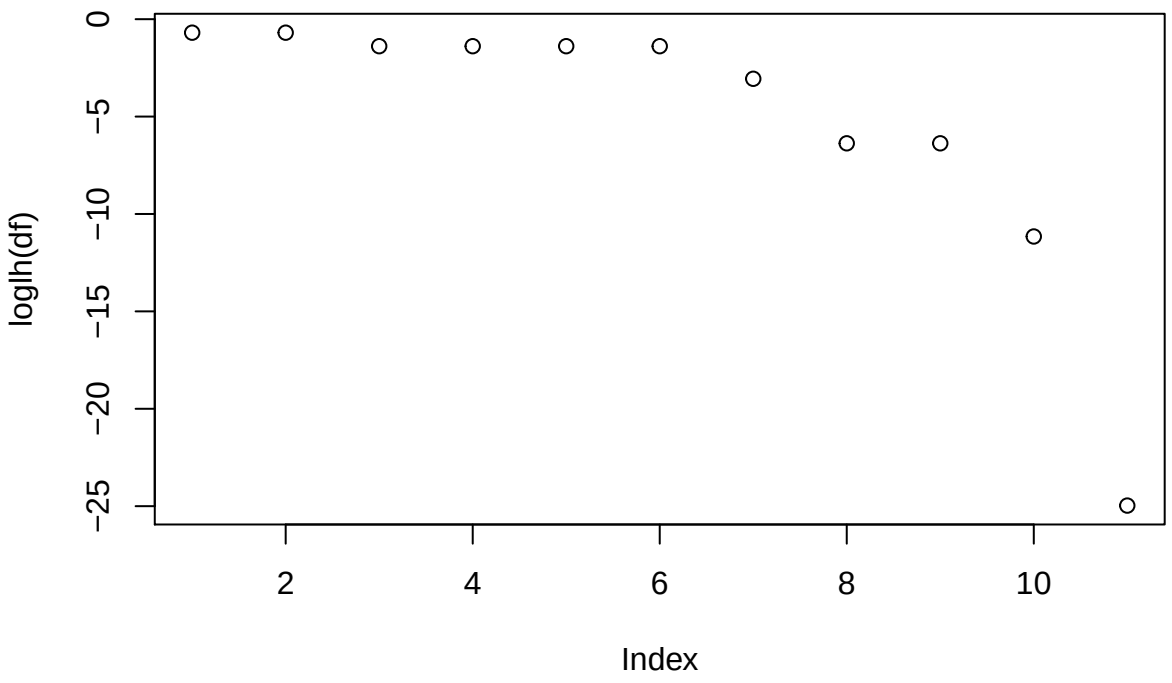
..

(e) Plot the log-likelihood as a function of p. Mark the ML estimator p on the gure.

```
nheads <- c (0, 0, 1, 1, 1, 1, 2, 3, 3, 4, 6)
Ph <- c(1/2, 1/2, 1/2, 1/2, 1/2, 1/2 , 1/4, 1/8,1/8, 1/16, 1/64)
df <- cbind(nheads, Ph)

loglh <-
  function(df)
  {
    for (i in df) {
    loglike = nheads*log(Ph) + log(1-Ph)
    return(loglike)
    print(loglike)
    }
  }

plot(loglh(df))
```




## REFERENCES:


```
# Nabble.com (June,2008). Retrieved From:
# http://r.789695.n4.nabble.com/Number-of-digits-in-paste-funciton-td860874.html

# Stackoverflow.(Oct 2016). Retrieved From:
# https://stackoverflow.com/questions/18587334/subset-data-to-contain-only-columns-whose-names-match-a-
```


## Statement of Compliance

Please copy and sign the following statement. You may do it on paper (and include the image le), or add the following text with your name and date in the rmarkdown document. I a rm that I have had no conversation regarding this exam with any persons other than the instruc- tor or the teaching assistant. Further, I certify that the attached work represents my own thinking. Any information, concepts, or words that originate from other sources are cited in accordance with University of Washington guidelines as published in the Academic Code (available on the course website). I am aware of the serious consequences that result from improper discussions with others or from the improper citation of work that is not my own.

Name: TAPASVI BANSAL

Date: 12/11/17