INFX573 - Data Science I - Final Exam

Charudatta Deshpande

Deadline: Mon, Dec 11th, 5:30pm PST

These 100 points will give you up to 25 points of the final credit.

Instructions:

This is a take-home final examination. You have 53 hours and 30 minutes to solve these problems. You may use your computer, books/articles, notes, course materials, internet, etc., but all work must be your own. References must be appropriately cited. Links to solutions copied from websites, such as StackOverflow must be provided in code comments. Please explain your answers and show all work; a complete argument must be presented to obtain full credit. All plots must be appropriately labeled, and appropriate colors/labels/font sizes must be used.

Statement of Compliance:

You must include the "signed" Statement of Compliance in your submission. The Compliance Statement is found on the last page of this exam. Failure to do so will result in your exam not being accepted.

```
# Load some helpful libraries that we might need during exam.
library(tidyverse)
library(data.table)
library(mosaic)
library(rpart)
library(ggplot2)
library(foreach)
library(doParallel)
library(stringr)
library(dplyr)
library(choroplethr)
library(choroplethrMaps)
library(pscl)
library(maxLik)
```

Problem 1: 2016 Election Results Data (25pt)

Use the following datasets to analyze the US 2016 election results (available on canvas in files/data) by county.

a. US_County_Level_Presidential_Results_08-16.csv.bz2

b. county_data.csv.bz2

c. there is also an explanation file for the county data county_data_variables.pdf.

Note: the datasets can be merged by FIPS (Federal Information Processing Standards) codes. There are 3-digit county FIPS codes and 2-digit state FIPS codeds, some data use 5 digit FIPS instead: 2 digits for the state followed by 3 digits for the county.

```r
# Load the data.
results <- read.csv("US_County_Level_Presidential_Results_08-16.csv.bz2")
county <- read.csv("county_data.csv.bz2")
# Convert to data.table
results <- as.data.table(results)
county <- as.data.table(county)
```

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

Answer -

We will perform following steps to tidy and merge the data.

1. county dataset - Set 'COUNTY' column length to 3, pad with leading zeroes to align it with FIPS code.
2. county dataset - Set 'STATE' column length to 2, pad with leading zeroes to align it with FIPS code.
3. county dataset - Merge the two columns to create a single column named 'FIPS_CODE'.
4. results dataset - Set 'fips_code' column length to 5, pad with leading zeroes to align it with FIPS code from county dataset.
5. Merge (inner join) the two datasets based on this derived FIPS code. This will automatically exclude state level records from county dataset where COUNTY = '000'. They do not exist in results dataset and there is no corresponding data.
6. The new dataset is called county.results. This has same number of rows as results dataset (3112 rows). The rows in county dataset that did not have a corresponding row in results dataset have been omitted. The new dataset county.results will only have following columns that I think is relevant to election results analysis -

FIPS_CODE
REGION
DIVISION
STATE
COUNTY
county (for verification only)
STNAME
CTYNAME
CENSUS2010POP
POPESTIMATE2012
POPESTIMATE2016
RINTERNATIONALMIG2012
RINTERNATIONALMIG2016
RDOMESTICMIG2012
RDOMESTICMIG2016
RNETMIG2012
RNETMIG2016
total_2008
dem_2008
gop_2008
oth_2008
total_2012
dem_2012
gop_2012
oth_2012

total_2016
dem_2016
gop_2016
oth_2016

7. Modify the 'REGION' variable according to below key -

1 = Northeast
2 = Midwest
3 = South
4 = West

8. Modify the 'DIVISION' variable according to below key -

1 = New England
2 = Middle Atlantic
3 = East North Central
4 = West North Central
5 = South Atlantic
6 = East South Central
7 = West South Central
8 = Mountain
9 = Pacific

```
# Tidy data - perform step 1 above.
# Set 'COUNTY' column length to 3, pad with leading zeroes
county$COUNTY <- str_pad(county$COUNTY, 3, pad = "0")
# Tidy data - perform step 2 above.
#Set 'STATE' column length to 2, pad with leading zeroes
county$STATE <- str_pad(county$STATE, 2, pad = "0")
# Tidy data - perform step 3 above.
# county dataset - combine STATE and COUNTY columns to get FIPS_CODE
# Move new column to first place.
county$FIPS_CODE <- paste(county$STATE, county$COUNTY, sep="")
county<-county[,c(117, 1:116)]
# Tidy data - perform step 4 above.
# Set 'fips_code' column length to 5, pad with leading zeroes
results$fips_code <- str_pad(results$fips_code, 5, pad = "0")
# Tidy data - perform step 5 and 6 above.
# Merge (inner join) datasets based on FIPS code.
# Retain only specific columns
dt1 <- data.table(county, key = "FIPS_CODE")
dt2 <- data.table(results, key = "fips_code")
tmp1 <- dt1[dt2]
copy.columns <- c("FIPS_CODE","REGION","DIVISION","STATE" ,"COUNTY","county",
                  "STNAME" ,"CTYNAME" ,"CENSUS2010POP","POPESTIMATE2012",
                  "POPESTIMATE2016","RINTERNATIONALMIG2012",
                  "RINTERNATIONALMIG2016",
                  "RDOMESTICMIG2012" ,"RDOMESTICMIG2016","RNETMIG2012",
                  "RNETMIG2016","total_2008","dem_2008","gop_2008","oth_2008",
                  "total_2012","dem_2012","gop_2012",
                  "oth_2012","total_2016","dem_2016","gop_2016","oth_2016")
tmp2 <- subset(tmp1, select=copy.columns)
# Tidy data - perform step 7 above. Transform REGION.
tmp3 <- tmp2 %>%
                  mutate(REGION = derivedFactor
```

```
                          ("Northeast" = (REGION == '1'),
                           "Midwest"   = (REGION == '2'),
                           "South"     = (REGION == '3'),
                           "West"      = (REGION == '4')))
# Tidy data - perform step 8 above. Transform DIVISION.
county.results <- tmp3 %>%
                  mutate(DIVISION = derivedFactor
                          ("New England"        = (DIVISION == '1'),
                           "Middle Atlantic"    = (DIVISION == '2'),
                           "East North Central" = (DIVISION == '3'),
                           "West North Central" = (DIVISION == '4'),
                           "South Atlantic"     = (DIVISION == '5'),
                           "East South Central" = (DIVISION == '6'),
                           "West South Central" = (DIVISION == '7'),
                           "Mountain"           = (DIVISION == '8'),
                           "Pacific"            = (DIVISION == '9')))
```

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

Answer -

We have selected following variables for further analysis.

Their explanation is provided below -

FIPS_CODE - 5 digit FIPS code for a county.
REGION - Census Region code. Transformed in above question based on key provided.
DIVISION - Census Division code. Transformed in above question based on key provided.
STATE - State FIPS code (included for verification only).
COUNTY - County FIPS code (included for verification only).
county - County name. Same as CTYNAME (included for verification only).
STNAME - State name.
CTYNAME - County name.
CENSUS2010POP - 4/1/2010 resident total Census 2010 population.
POPESTIMATE2012 - 7/1/2012 resident total population estimate.
POPESTIMATE2016 - 7/1/2016 resident total population estimate.
RINTERNATIONALMIG2012 - Net international migration rate in period 7/1/2011 to 6/30/2012.
RINTERNATIONALMIG2016 - Net international migration rate in period 7/1/2015 to 6/30/2016.
RDOMESTICMIG2012 - Net domestic migration rate in period 7/1/2011 to 6/30/2012.
RDOMESTICMIG2016 - Net domestic migration rate in period 7/1/2015 to 6/30/2016.
RNETMIG2012 - Net migration rate in period 7/1/2011 to 6/30/2012.
RNETMIG2016 - Net migration rate in period 7/1/2015 to 6/30/2016.
total_2008 - Total votes in county in 2008 presidential election.
dem_2008 - Number of Democratic votes in county in 2008 presidential election.
gop_2008 - Number of Republican votes in county in 2008 presidential election.
oth_2008 - Number of Other Party votes in county in 2008 presidential election.
total_2012 - Total votes in county in 2012 presidential election.
dem_2012 - Number of Democratic votes in county in 2012 presidential election.
gop_2012 - Number of Republican votes in county in 2012 presidential election.
oth_2012 - Number of Other Party votes in county in 2012 presidential election.
total_2016 - Total votes in county in 2016 presidential election.
dem_2016 - Number of Democratic votes in county in 2016 presidential election.
gop_2016 - Number of Republican votes in county in 2016 presidential election.
oth_2016 - Number of Other Party votes in county in 2016 presidential election.

Out of these selected variables, I believe following variables may have a relationship with the outcome of presidential elections (for appropriate year) -

CENSUS2010POP
POPESTIMATE2012
POPESTIMATE2016
RINTERNATIONALMIG2012
RINTERNATIONALMIG2016
RDOMESTICMIG2012
RDOMESTICMIG2016
RNETMIG2012
RNETMIG2016

There is no data avaialable in the 'county dataset' for year 2008. We will be unable to analyze the 2008 election results, though the data exists in 'results' dataset.

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.

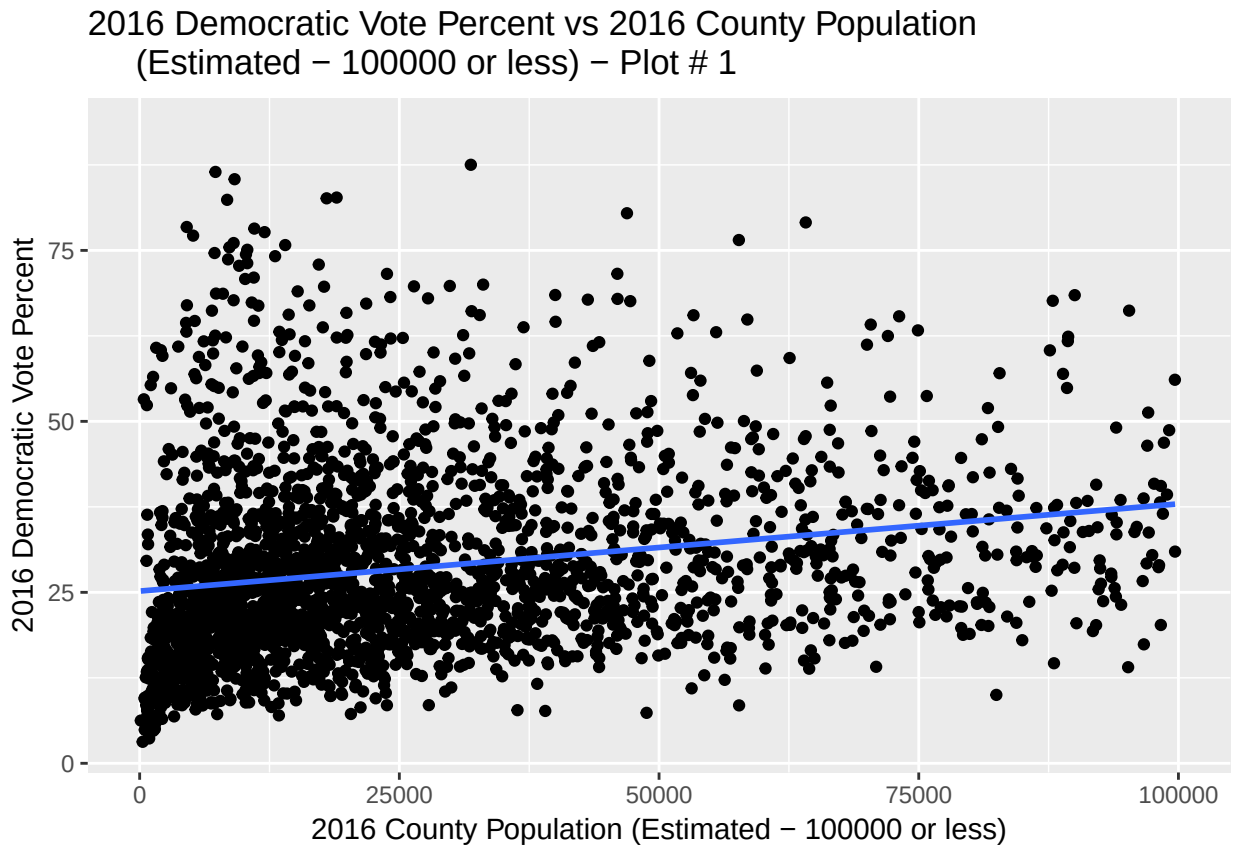
Answer -

We will first create columns for democratic votes percentage for 2008, 2012 and 2016. The we will plot 2016 democratic vote percentages against county population for 2016. We will use the variable POPESTIMATE2016 for population. This is only an estimate since last census was done in 2010. But this is the best available field in the dataset.

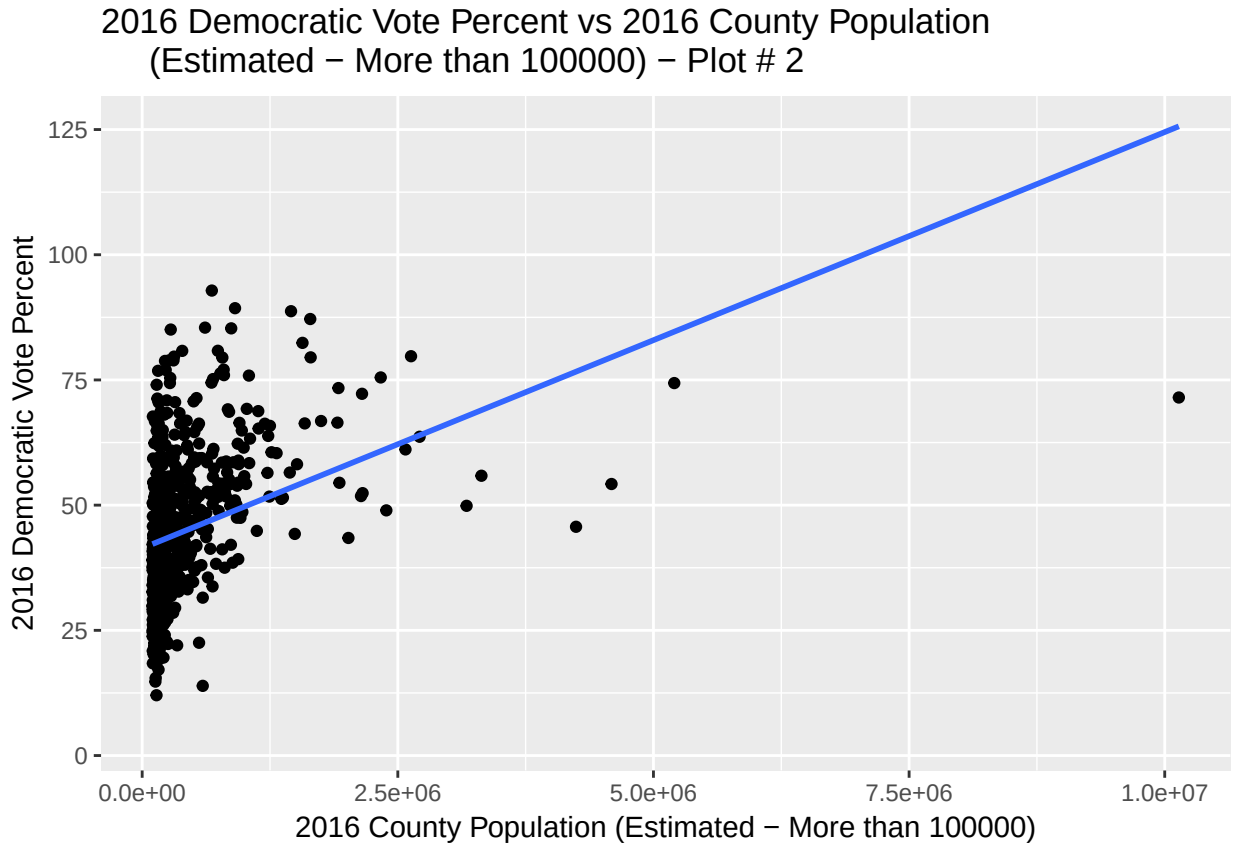
For the ease of plotting and to accommodate large population range, we will create 2 plots. For population upto 100,000 and for population above 100,000. We will also draw a linear regression line which would indicate the trend.

We will also estimate a separate linear model and do statistical significance analysis on model.

```
# Create democratic votes percentage columns for 2008, 2012 and 2016.
county.results$dem_2008_percent <- county.results$dem_2008*100/county.results$total_2008
county.results$dem_2012_percent <- county.results$dem_2012*100/county.results$total_2012
county.results$dem_2016_percent <- county.results$dem_2016*100/county.results$total_2016
#Create a plot of 2016 vote percent vs county population 2016 estimate.
#For population upto 100,000
ggplot(county.results, aes(POPESTIMATE2016, dem_2016_percent)) +
labs(x="2016 County Population (Estimated - 100000 or less)",
     y="2016 Democratic Vote Percent",
     title = "2016 Democratic Vote Percent vs 2016 County Population
     (Estimated - 100000 or less) - Plot # 1") +
xlim(1, 100000) +
geom_point() +
geom_smooth(method=lm, se=FALSE)
```




```
#Create a plot of 2016 vote percent vs county population 2016 estimate.
#For population more than 100,000
ggplot(county.results, aes(POPESTIMATE2016, dem_2016_percent)) +
labs(x="2016 County Population (Estimated - More than 100000)",
     y="2016 Democratic Vote Percent",
     title = "2016 Democratic Vote Percent vs 2016 County Population
     (Estimated - More than 100000) - Plot # 2") +
xlim(100001, 10140000) +
geom_point() +
geom_smooth(method=lm, se=FALSE)
```




```
# Linear regression between percentage and population
m1 <- lm(dem_2016_percent ~ POPESTIMATE2016, data=county.results)
summary(m1)
```

```
##
## Call:
## lm(formula = dem_2016_percent ~ POPESTIMATE2016, data = county.results)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -122.346  -10.217   -2.689    7.800   56.985
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.002e+01  2.698e-01  111.27   <2e-16 ***
## POPESTIMATE2016 1.616e-05  7.762e-07   20.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.36 on 3109 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.1224, Adjusted R-squared:  0.1221
## F-statistic: 433.4 on 1 and 3109 DF,  p-value: < 2.2e-16
```

The linear trend line for both plots indicates that the democratic vote percent increases as population increases.

For linear regression analysis, the p-value is 2e-16, which indicates a statistically significant association. The Multiple R-squared value is 0.1224 which indicates that our model does a poor job at explaining the response variable. F-statistic value is 433.4 which is far greater than 1, which also indicates a statistically significant association.

Thus we reject the null hypothesis and conclude that relationship between demographic vote percentage and County population is statistically significant, and we note the need for a better model that would get a better Multiple R-squared value.

Conclusion -

From plots and linear model, we clearly see a statistically significant association between demographic vote percentage and County population. Larger the county population, better the vote percentage. Counties with larger population are usually associated with more industry presence, better education opportunities, larger local governments and bigger cities (imagine King County). This leads me to conclude that big cities in US mostly vote democratic, and the voters are probably more educated or at least have better education opportunities available.

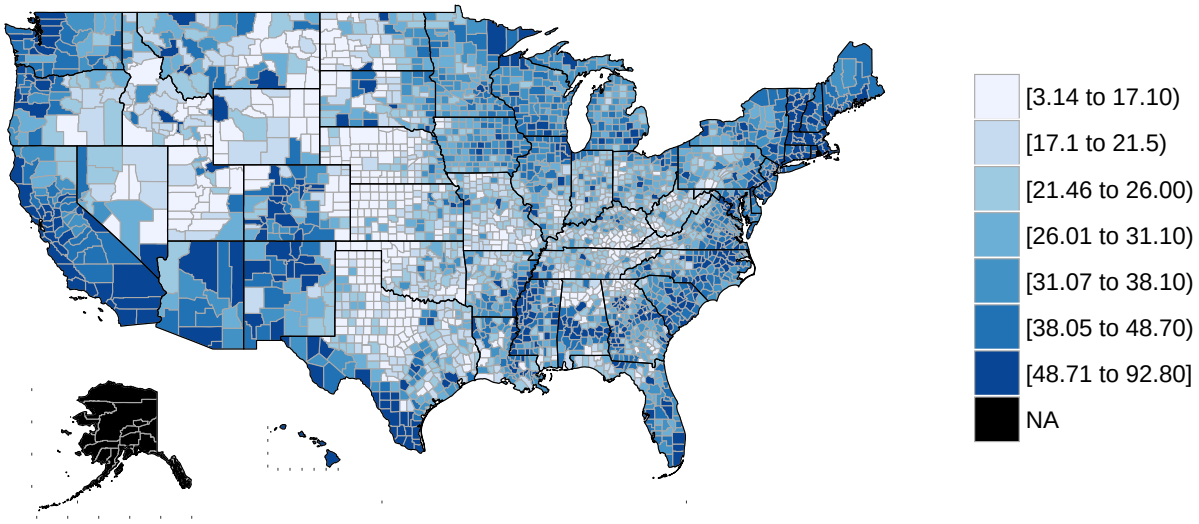
4. Create a map of percentage of votes for democrats. Do your best to reflect the continuous percentage of votes, and the different population sizes across counties and keep county boundaries as well legible as you can. Mark state boundaries on the map. Explain what did you do, and what worked well, 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”.

Answer -

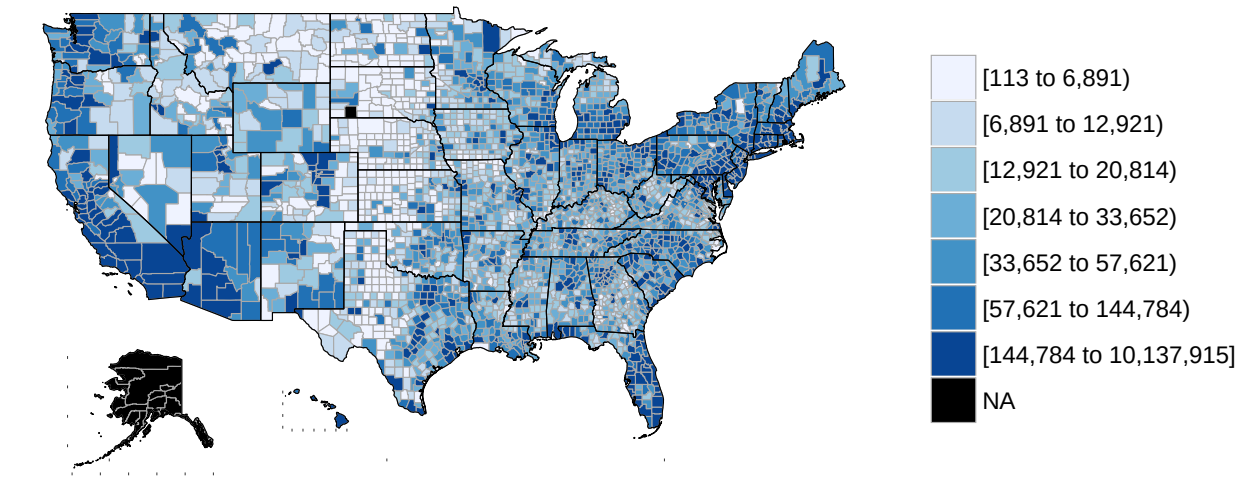
We will create two county level maps of US. One for demographic vote percentage, and the other for population. We will try to display these maps side by side so they will be easy to analyzie. We will use packages ‘choroplethr’ and ‘choroplethrMaps’ for this activity.

```
# Create data to be used by county_choropleth function.
region <- NULL
value <- NULL
region <- as.integer(county.results$FIPS_CODE)
value <- county.results$dem_2016_percent
plot1 <- data.frame(region, value)
region <- NULL
value <- NULL
region <- as.integer(county.results$FIPS_CODE)
value <- county.results$POPESTIMATE2016
plot2 <- data.frame(region, value)
par(mfrow=c(1,2))
# Create a map of democratic vote percent.
county_choropleth(plot1, title="Democratic Vote Percentage - 2016")
```

Democratic Vote Percentage − 2016




```
# Create a map of county population
county_choropleth(plot2, title="County Population Estimate - 2016")
```



Analysis of maps -

What worked well -

As can be seen, the two maps, vote percentage and population, look really nice and provide very useful insights. They are titled and the bar to the right indicates the values represented by different color. The maps also indicate that we do not have any election data ablout Alaska, and few other mainland counties indicated in Black.

State and County boundaries are very clearly marked, and the maps are visually appealing.

Conclusions from the part that worked well -

Our conclusion from previous question is reinforced, that as population increases, democratic vote percent increases. When two maps are placed side by side, they provide a clear correlation between the two variables of interest.

What did not work well -

The fundamental issue is that there are two maps to compare. A single map would have worked better. I tried my best to imagine how to represent two parameter on a single county map. I could not imagine a visualization that would provide information about two variables on same map. To do something like that, I would have had to sacrifice the simplicity of the map.

Then my next best idea was to plot the maps side by side so that would be easy for comparison. As you can see I used the command 'par(mfrow=c(1,2))' but that did not work. It seems that combining two maps is not possible with county_choropleth function. I would have been able to combine these with ggmap, but that would have required me to rework this question, and I ran out of time. In the end, I decided to keep these as two maps since they still provide very useful information and reinforce our conclusions.

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

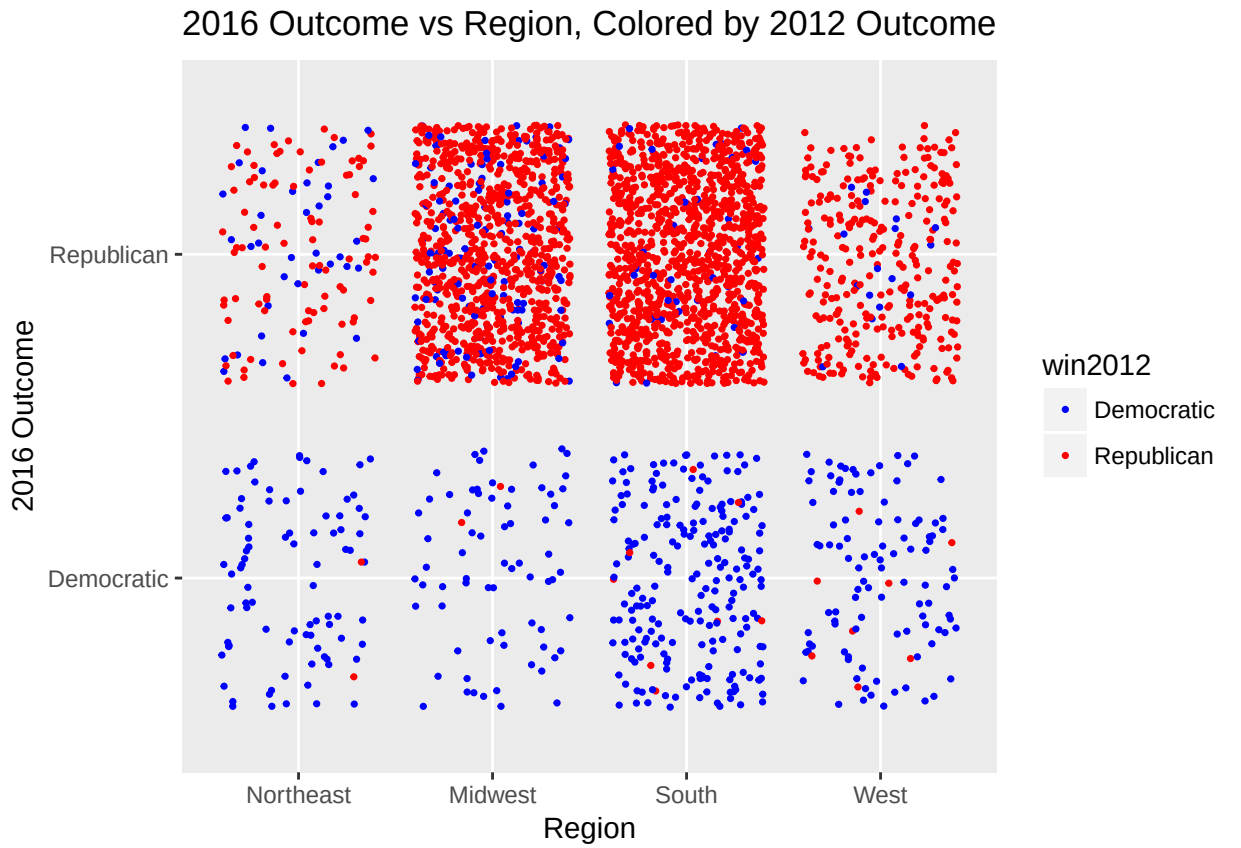
Answer -

We will analyze how Democrats and Replublicans performed in 2012 and 2016 in 4 major regions of the country -

Northeast
Midwest
South
West

First, we will create two columns which would indicate Democreatic or Republican wins in 2012 and 2016. Then we will plot a jitter map of 2016 outcome vs regions and color them by 2012 results. We will then analyze the results.

```
# Create columns to identify who won the county in 2012 and 2016.
county.results$win2012 <- ifelse(county.results$dem_2012 > county.results$gop_2012,
                                 "Democratic", "Republican")
county.results$win2016 <- ifelse(county.results$dem_2016 > county.results$gop_2016,
                                 "Democratic", "Republican")
# Create a jitter plot to analyze relationship between win and region.
# Color the points by 20112 outcome. Use Blue and Red per party lines.
# Remove 'NA' records for REGION.
county.results %>%
drop_na(REGION) %>%
ggplot(aes(REGION, win2016, col=win2012)) +
geom_jitter(size = 0.6) +
scale_colour_manual(values = c("Democratic" = "blue", "Republican" = "red")) +
labs(x="Region",
     y="2016 Outcome",
     title = "2016 Outcome vs Region, Colored by 2012 Outcome")
```



Analysis of plot -

Red dots in the 2016 Democratic win area indicate the counties that were won by Republicans in 2012. Similarly, Blue dots in 2016 Republican win area indicate the counties that were won by Democrats in 2012. From this plot, we can draw a number of conclusions -

1. In both 2012 and 2016, Republicans won a lot more counties than Democrats. But in 2012, Democrats won Presidency by both, electoral vote and popular vote. In 2016, despite losing presidency by electoral vote, Democrats still won popular vote by bigger margin. This indicates that despite winning very few counties by comparison, democrats got more votes both times. This reinforces our previous conclusion that democrats tend to win counties with larger population, and republicans win a lot more counties with less population.
2. Democrats lost more counties to republicans in 2016, than republicans to democrats. But democrats still won popular vote by bigger margin than in 2012. This indicates that democrats lost samller counties, and their vote share increased in bigger counties.
3. Democrats seem to do really poor in Midwest, and in 2016 they did even worse than 2012.
4. Northeast saw the biggest defection from democrats in 2016. They lost many counties to republicans, but won won only 2 counties from republicans.
5. 2016 Presidential election result was influenced by very few 'swing counties'. These counties are shown my opposite colored dots in 2016 win areas. An overwhelming majority of counties voted for same party in 2016 as they did in 2016. The 'swing counties', very less in number on comparison, elected a republican president in 2016.

Problem 2: 2016 Election Model (25pt)

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

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

Answer -

We will use the same data created in last question. We have already included variables that might be relevant for the election results in the county.results dataset.

I believe following variables may be related to election results. This is specific to 2016. Similar analysis can be performed for 2012, but for this question we will keep it limited to year 2016.

REGION - We have seen in Q1.5, region has a distinguishable relationship with election outcome. We will use that in our analysis.

POPESTIMATE2016 - As seen in Q1, 2016 population has a statistically significant relationship with 2016 election outcome.

win2012 - This is a derived field. This would indicate which party won the county in 2012. As seen from previous plots, it is a strong indicator of 2016 results.

RINTERNATIONALMIG2016 - This is rate of county population leaving the country in 2016. Since total county population has statistically significant relationship with outcome, this could be related with election outcome too.

RDOMESTICMIG2016 - This is rate of county population relocating elsewhere in the country in 2016. Since total county population has statistically significant relationship with outcome, this could be related with election outcome too.

RNETMIG2016 - Combined effect of RINTERNATIONALMIG2016 and RDOMESTICMIG2016.

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).

Answer -

We will first create binary value columns for win2012 and win2016.

Then we will establish a logistic regression model and predict it to calculate the probability that a county voted democratic. The results column glm.result will be added to the dataset. This new column glm.result will indicate the probability, based on the logistic regression model, that a county voted democratic.

```
# Omit records without any data (NA). There is only 1.
county.results <- county.results[!is.na(county.results$STATE),]
# Create a binary value column for win2012.
# 1 = Democrat win, 0 = Republican win
county.results$win2012.bin <-ifelse(county.results$win2012 == "Democratic", 1, 0)
county.results$win2012.bin <- factor(county.results$win2012.bin)
# Create a binary value column for win2016.
# 1 = Democrat win, 0 = Republican win
county.results$win2016.bin <-ifelse(county.results$win2016 == "Democratic", 1, 0)
county.results$win2016.bin <- factor(county.results$win2016.bin)
# Estimate a logistic regression with some predictor variables.
glm1 <- glm(win2016.bin ~ REGION + POPESTIMATE2016 + win2012.bin,
            family=binomial(link="logit"), data=county.results)
summary(glm1)
```

```
##
## Call:
## glm(formula = win2016.bin ~ REGION + POPESTIMATE2016 + win2012.bin,
##     family = binomial(link = "logit"), data = county.results)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.4049  -0.1296  -0.1211  -0.0409   3.4829
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -6.078e+00  3.701e-01 -16.425  < 2e-16 ***
## REGIONMidwest   -1.043e+00  2.635e-01  -3.959 7.52e-05 ***
## REGIONSouth      1.140e+00  2.662e-01   4.282 1.85e-05 ***
## REGIONWest       1.276e+00  3.249e-01   3.927 8.60e-05 ***
## POPESTIMATE2016  3.418e-06  4.675e-07   7.311 2.64e-13 ***
## win2012.bin1     6.076e+00  2.866e-01  21.203  < 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: 2696.29  on 3110  degrees of freedom
## Residual deviance:  849.15  on 3105  degrees of freedom
## AIC: 861.15
##
## Number of Fisher Scoring iterations: 8
```

```
# Estimate probability that a county voted democratic by predicting the model
county.results$glm.result <- predict(glm1, type="response")
summary(county.results$glm.result)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0008082 0.0010770 0.0077870 0.1562000 0.0145100 1.0000000
```

```
# Add another column which would indicate binary probability- 0 or 1.
# A value greater than or equal to 0.5 will round to 1.
# A value of less than 0.5 will round to 0.
county.results$glm.result.bin <- ifelse(predict(glm1, type="response")>= 0.5, 1, 0)
print(table(county.results$win2016.bin, county.results$glm.result.bin))
```

```
##
##        0    1
##   0 2521  104
##   1   63  423
```

```
# Calculate success ratio.
success.rate.glm1 <- table(county.results$win2016.bin,
                           county.results$glm.result.bin) %>% diag() %>% sum()/nrow(county.results)
success.rate.glm1
```

```
## [1] 0.9463195
```

As can be seen, when converted to binary outcomes, this model gives us a strong success ratio of 0.9463195.

3. Experiment with a few different specifications and report the best one you got. Explain what did you do.

Hint: we did not talk about choosing between different logistic regression models. You may use a pseudo-R2 value in a similar fashion as you use R2 for linear models. For instance, pscl::pR2 will provide a number of different pseudo-R2 values for estimated glm models, you may pick McFadden's version.

Answer -

We will try a few combinations of predictor variables. We will use pR2 function and use McFadden's pseudo R-squared value to estimate the accuracy of the model.

```
# Use pR2 on above glm1.
pR2(glm1)
```

```
##           llh       llhNull            G2      McFadden          r2ML
##  -424.5734323 -1348.1457842  1847.1447038     0.6850686     0.4477455
##          r2CU
##     0.7724252
```

```
# Estimate second model with some predictor variables.
glm2 <- glm(win2016.bin ~ REGION + POPESTIMATE2016,
            family=binomial(link="logit"), data=county.results)
pR2(glm2)
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -1097.5811410 -1348.1457842   501.1292863     0.1858587     0.1487786
##          r2CU
##     0.2566644
```

```
# Estimate third model with some predictor variables.
glm3 <- glm(win2016.bin ~ REGION + POPESTIMATE2016 + RINTERNATIONALMIG2016 +
              RDOMESTICMIG2016, family=binomial(link="logit"), data=county.results)
pR2(glm3)
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -1012.3305425 -1348.1457842   671.6304834     0.2490942     0.1941752
##          r2CU
##     0.3349801
```

```
# Estimate fourth model with some predictor variables.
glm4 <- glm(win2016.bin ~ REGION + POPESTIMATE2016 + RINTERNATIONALMIG2016 +
              RDOMESTICMIG2016 + RNETMIG2016, family=binomial(link="logit"),
            data=county.results)
pR2(glm4)
```

```
##           llh       llhNull            G2      McFadden          r2ML
## -1012.3168177 -1348.1457842   671.6579330     0.2491043     0.1941823
##          r2CU
##     0.3349923
```

Note - we will interpret the results in Q 2.6. For this question, we will only select the best model based on McFadden's pseudo R-squared value.

As can be seen above glm1 (predictors - REGION, POPESTIMATE2016 and win2012.bin) yields the best available 0.6850686 value for McFadden's pseudo R-squared value. We will consider that as our best model for next 3 sub-questions.

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

Answer -

Statistical significance indicates that a result, usually finding a correlation between variables, cannot be attributed to random chance. When a result is statistically significant, we would reject the null hypothesis. Null hypothesis suggests that the variables under consideration have no associated correlation. When a result is statistically significant, we can reject null hypothesis and conclude that there is a measurable association between variables under consideration.

Confidence level is how likely the value will fall within your confidence interval. The confidence coefficient is the proportion of samples of a given size that may be expected to contain the true value of the response variable. That is, for a 95 % confidence interval, if many samples are collected and the confidence interval was measured, in the long run about 95 % of these intervals would contain the true value of the response variable that we are trying to estimate a relationship with.

For a normal distribution, the 95% confidence interval is evenly distributed around 0, and is usually considered from 2.5% to 97.5% values.

Note - Part of this answer was derived from information available on internet. Refer to references for details. Most of this is in my own words, but I took help from the provided references to build my answer.

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

Answer -

```
# Provide summary of glm1, the best model based on
summary(glm1)
```

```
##
## Call:
## glm(formula = win2016.bin ~ REGION + POPESTIMATE2016 + win2012.bin,
##     family = binomial(link = "logit"), data = county.results)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.4049  -0.1296  -0.1211  -0.0409   3.4829
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -6.078e+00  3.701e-01 -16.425  < 2e-16 ***
## REGIONMidwest   -1.043e+00  2.635e-01  -3.959 7.52e-05 ***
## REGIONSouth      1.140e+00  2.662e-01   4.282 1.85e-05 ***
## REGIONWest       1.276e+00  3.249e-01   3.927 8.60e-05 ***
## POPESTIMATE2016  3.418e-06  4.675e-07   7.311 2.64e-13 ***
## win2012.bin1     6.076e+00  2.866e-01  21.203  < 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: 2696.29  on 3110  degrees of freedom
## Residual deviance:  849.15  on 3105  degrees of freedom
## AIC: 861.15
##
## Number of Fisher Scoring iterations: 8
```

Based on the summary and p-values, are three predictor variables, REGION, POPESTIMATE2016 and win2012.bin have a statistically significant relationship with 2016 election outcomes from a county.

6. Interpret the results. Provide correct interpretable explanations about what the most important effect are and what do the particular numeric results mean.

Hint: you may use either odds ratios or marginal effects.

Answer -

We will interpret the results based on odds ratio. First we will calculate the odds ratio for glm 1.

```
## odds ratios and 95% CI
exp(cbind(odds.ratio = coef(glm1), confint(glm1)))
```

```
##                    odds.ratio        2.5 %       97.5 %
## (Intercept)     2.292055e-03 1.079761e-03 4.628277e-03
## REGIONMidwest   3.523193e-01 2.089967e-01 5.880802e-01
## REGIONSouth     3.126380e+00 1.855887e+00 5.278701e+00
## REGIONWest      3.581999e+00 1.909167e+00 6.841442e+00
## POPESTIMATE2016 1.000003e+00 1.000003e+00 1.000004e+00
## win2012.bin1    4.353506e+02 2.560335e+02 7.916678e+02
```

Based on the summary and p-values, are three predictor variables, REGION, POPESTIMATE2016 and win2012.bin have a statistically significant relationship with 2016 election outcomes from a county.

Odds ratio analysis -

Looking at the odds ratios, REGIONMidwest has an odds ratio of 0.352 which means compared to Northeast, Midwest has only 35.2% odds of democratic win. For South, the odds are much better at 3.126 - 1 = 2.126 or 212.6% odds of democratic win. For West, the odds are still higher at 3.581 - 1 = 2.581 or 258.1% odds of democratic win compared to Northeast.

POPESTIMATE2016 has a positive correlation with election outcome, as suggested by our earlier models. The coefficient is very close to 1, which indicates that a unit increase in population increases the odds ratio of a democratic win by a factor of 1.

win2012.bin - This is a binary variable indicating if democratic candidate won in 2012 election or not. The odds ratio is highest at 435.35 - 1 = 434.35. Thus, if a democratic candidate won in 2012, the odds of the democratic win in 2016 are 434.35 compared to a republican win. This is a very strong relationship.

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

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

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

Answer -

We will choose **Poission distribution** for this analysis.

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

```
# Create a new column in the county.results dataset that will have
# random poission numbers. We will choose a rate parameter of 10.
# We will then have a vector of 3111 observations.
rate.parameter <- 10
county.results$random.poisson.number.column <- rpois(nrow(county.results), rate.parameter)
```

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

```
# glm1 was our best regression from Q2. We will use that and add
# random.poisson.number as additional predictor.
glm1.poisson <- glm(win2016.bin ~ REGION + POPESTIMATE2016 + win2012.bin +
```

```
                    random.poisson.number.column,
                    family=binomial(link="logit"), data=county.results)
```

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

```
# store the coeffcient for the random variable
random.poisson.number.coeff <- coef(glm1.poisson)["random.poisson.number.column"]
random.poisson.number.coeff
```

```
## random.poisson.number.column
##                -0.002034255
```

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

```
# Perform the process 3000 times. Also store the time taken by
# sequential processing. This will be used later.
R <- 3000
sequentual.process.time <-
  system.time(random.number.coeff.vector <- foreach(i = 1:R, .combine=c) %do% {
  random.poisson.number <- rpois(nrow(county.results), rate.parameter)
  glm1.poisson <- glm(win2016.bin ~ REGION + POPESTIMATE2016 + win2012.bin +
                    random.poisson.number,
                    family=binomial(link="logit"), data=county.results)
  coef(glm1.poisson)["random.poisson.number"]
})
```

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

Answer -

```
# Calculate mean and standard daviation for random.number.coeff.vector
mean <- mean(random.number.coeff.vector)
mean
```

```
## [1] 0.0009039455
```

```
sd <- sd(random.number.coeff.vector)
sd
```

```
## [1] 0.02829686
```

Mean and standard daviation are printed above. Since set.seed is not specified, the values are not reproducible.

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

Answer -

```
# Calculate 95% confidence interval of the vector random.number.coeff.vector
quantile(random.number.coeff.vector, c(0.025, 0.975))
```

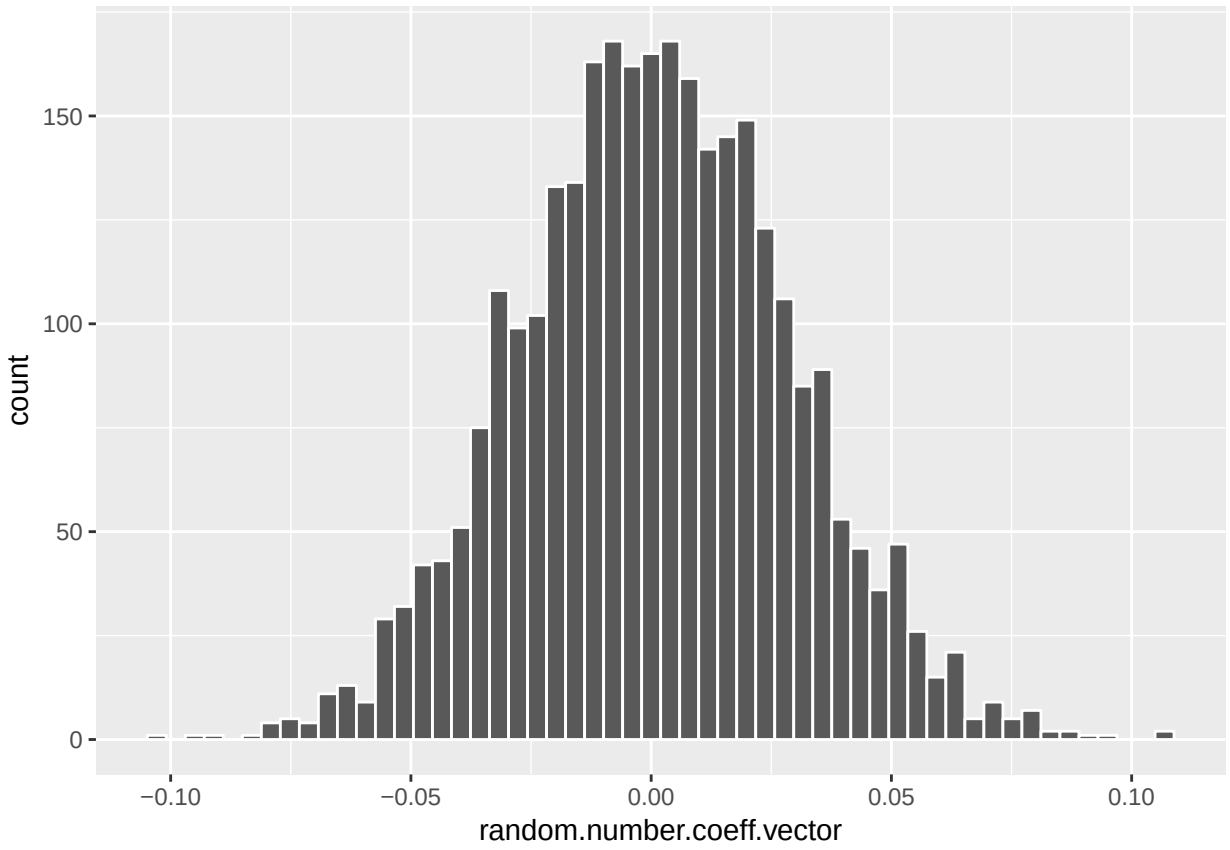
```
##        2.5%       97.5%
## -0.05381541  0.05684578
```

95% confidence interval values are printed above. Since set.seed is not specified, the values are not reproducible.

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

Answer -

```
# Plot the distribution of the estimates in a histogram
qplot(random.number.coeff.vector, geom="histogram", col=I("White"),
      bins=sqrt(length(random.number.coeff.vector)))
```



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

Answer -

I spent more than 70 minutes making sense of 'randomly distributed', and what it means in this context. However I didn't understand what 'randomly distributed' meant, and I could only make sense of it if it was 'normally distributed'. I raised the question on Slack at 1.44 PM on 12/10.

For lack of time, I am assuming that the question really means 'normally distributed'. Then I can theoritically calculate 95% confidence intervals absed on below code.

Same goal can be achieved using a t.test function.

```
# With the above mean and standard daviation, calculate
# standard error.
error <- qnorm(0.975)*sd/sqrt(R)
left <- mean - error
right <- mean + error
# left and right margins of 95% CI are below.
left
```

```
## [1] -0.0001086259
```

```
right
```

```
## [1] 0.001916517
```

```
# Do same thing using t.test function.
t.test(random.number.coeff.vector)
```

```
##
##  One Sample t-test
##
## data:  random.number.coeff.vector
## t = 1.7497, df = 2999, p-value = 0.08027
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0001090347  0.0019169257
## sample estimates:
##    mean of x
## 0.0009039455
```

Theoritical 95% confidence interval values are printed above.

Note - Part of this answer was derived from information available on internet. Refer to references for details. Most of this is my own work, but I took help from the provided references to build my answer, specifically the code above that calculates error, left margin and right margin.

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

Answer -

```
# Use 4 cores
cl <-makeCluster(4)
# register your data for dedicated memory
clusterExport(cl, "county.results")
registerDoParallel(cl)
# Run the same loop R times (3000) and calculate time. Use
# 'dopar' for parallel processing.
parallel.process.time <-
  system.time(random.number.coeff.vector <- foreach(i = 1:R, .combine=c) %dopar% {
  random.poisson.number <- rpois(nrow(county.results), rate.parameter)
  glm1.poisson <- glm(win2016.bin ~ REGION + POPESTIMATE2016 + win2012.bin +
                      random.poisson.number,
                      family=binomial(link="logit"), data=county.results)
  coef(glm1.poisson)["random.poisson.number"]
})
sequentual.process.time
```

```
##    user  system elapsed
##   57.59    0.07   57.92
```

```
parallel.process.time
```

```
##    user  system elapsed
##    1.39    0.27   28.45
```

```
# savings in seconds in elapsed times can be found below
sequentual.process.time - parallel.process.time
```

```
##    user  system elapsed
```

```
##   56.20   -0.20   29.47
```

Based on above values, at the time of running this simulation, the runtime was reduced to more than hald (more than 50% reduction). But this is at the time of running it on console. When this PDF is knit, it may provide a slightly different value. But it should be very close to what we have here.

Problem 4: Coin Tossing Game (25p)

You are offered 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,{.z. . ,H} n, T), you will receive $2^n. The tosses are independent.

1. Assume the coin fair.
(a) Compute your expected payoff in this game.

Answer -

Let us assume 'n' is the number of total tosses. So n starts with 1 and can reach upto ∞ in theory. Whenever a Tail appears, we will have 'n-1' number of heads, which could be 0 if tail appears on very first toss. The player will receive USD 2^{n-1}.

Probability of a tail on nth flip is -

$P(tail; nflips) = (1/2)^n$.

The player will receive USD 2^{n-1}.

We get an expectation of payoff like below -

E(Payoff) = $\sum_{n=1}^{\infty}(1/2)^n * 2^{(n-1)}$

E(Payoff) = $\sum_{n=1}^{\infty}(1/2)$

E(Payoff) = ∞

The theoritical payoff is infinite amount of dollars if infinite amount of trials are conducted. Practically, we are bound by available money to invest and time. To receive USD 32 (2^5), 6 tosses have to be made that are (H, H, H, H, H, T) and the probability of that happening is (1/2)^6 = 0.015625 which is slightly more than 1 percent.

To summarize, in theory a player can win infinite money but practially, crossing $100 would probably happen once in a lifetime of a player.

Note - Part of this answer was derived from information available on internet. Refer to references for details. Most of this is my own work, but I took help from the provided references to build my answer.

(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.

Answer -

The theoritical payoff is infinite amount of dollars, if a player has infinite amount of money to invest and infinite amount of time. Because receiving a large amount of payoff will take very large number of trials. Practially, neither is possible, and for small values of n, winning more than a few dollars is highly unlikely.

From above summary, in theory a player can infinite money but practially, crossing USD 100 would probably happen once in a lifetime of a player. Even earning USD 32 has a probability of 0.015625.

If I were playing, I would like to keep the probability at about 25%, which is two tosses. My expected payoff is USD 2. So I would only pay $1 to play this game. And even with that, there is a 75% probability that I will not get the USD 2 too.

Note - Above is my theoritical answer. But in practise since everyone is guranteed at least USD 1, the organizer may not accept a USD 1 investment, and may ask for more. So more likely, I would have to settle somewhere at an entry fee of USD 2 or slightly more. But in that case, I am more likely to not make any profit.

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 find the Maximum Likelihood estimator of p the probability of receiving a head.

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

Answer -

Let p be the probability to receive a head. That means probability to receive a tail is (1 - p).

For solution, refer to attached image file.

Note - all image files are also submitted in Canvas. Also, I noticed that images are not in the order of RMD file, and knit process rearranges them while knitting the PDF. I was unable to put the images exactly where I wanted them to be. Apologies for any inconvenience.

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

Answer -

Considering 10 events above -

Two events - P(T) here (n=0)
Three events - P(H, T) here (n = 1)
One event - P(H, H, T) here (n = 2)
Two events - P(H, H, H, T) here (n = 3)
One event - P(H, H, H, H, T) here (n = 4)
One event - P(H, H, H, H, H, H, T) here (n = 6)

Total number of heads = 21
Total number of tails = 10
Total number of flips = 31

All these can be multiplied to calculate the probability to receive all 10 outcomes.

Refer to attached image.

Note - all image files are also submitted in Canvas. Also, I noticed that images are not in the order of RMD file, and knit process rearranges them while knitting the PDF. I was unable to put the images exactly where I wanted them to be. Apologies for any inconvenience.

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

Answer -

Refer to attached image. Answer 4.2.b and 4.2.c are on same image.

Note 1 - all image files are also submitted in Canvas. Also, I noticed that images are not in the order of RMD file, and knit process rearranges them while knitting the PDF. I was unable to put the images exactly where I wanted them to be. Apologies for any inconvenience.

Note 2 - All of this is my own work. I took help from the provided references to verify my answer.

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

Charudatta Deshpande

Final Exam

Date: 12/10/2017

Q. 4. 2. a

Probability to receive 'n' heads & a tail.

$$P(n \text{ heads \& } 1 \text{ tail}) = {}^{n+1}C_{n}\, p^{n} * (1-p)$$

Q. 4. 2. b

Since these events are independent, we can multiply them to get total probability.

$$P(\text{All Events}) = \left[{}^{1}C_{0}\, p^{0} * (1-p) \right] \times \left[{}^{1}C_{0}\, p^{0} * (1-p) \right]$$

$$\times \left[{}^{2}C_{1}\, p^{1} (1-p) \right] \times \cdots\cdots$$

$$\propto \left[{}^{7}C_{6}\, p^{6} (1-p) \right]$$

Figure 1: Charu Q. 4.2.a - Analytical Solution

The total likelihood is denoted by $L(p)$.

Q.4.2.b. (cont)

$$L(p) = \prod_{i=1}^{10} P(n \text{ heads \& } 1 \text{ tail})$$

Q.4.2.c ~~(cont)~~

Log likelihood is →

$$l(p) = \sum_{i=1}^{10} \log \left(C_{n_i}^{n_i+1} p^{n_i} (1-p) \right)$$

$$= \sum_{i=1}^{10} \left[\log C_{n_i}^{n_i+1} + n_i \log p + \log(1-p) \right]$$

$$= \sum_{i=1}^{10} \left[\log C_{n_i}^{n_i+1} \right] + \log p \sum_{i=1}^{10} n_i$$

$$+ 10 \log (1-p)$$

Figure 2: Charu Q. 4.2.b - Analytical Solution

Answer -

Refer to attached image.

Note 1 - all image files are also submitted in Canvas. Also, I noticed that images are not in the order of RMD file, and knit process rearranges them while knitting the PDF. I was unable to put the images exactly where I wanted them to be. Apologies for any inconvenience.

In general, maximum likelihood parameter for p is (total number of heads received/total number of flips).

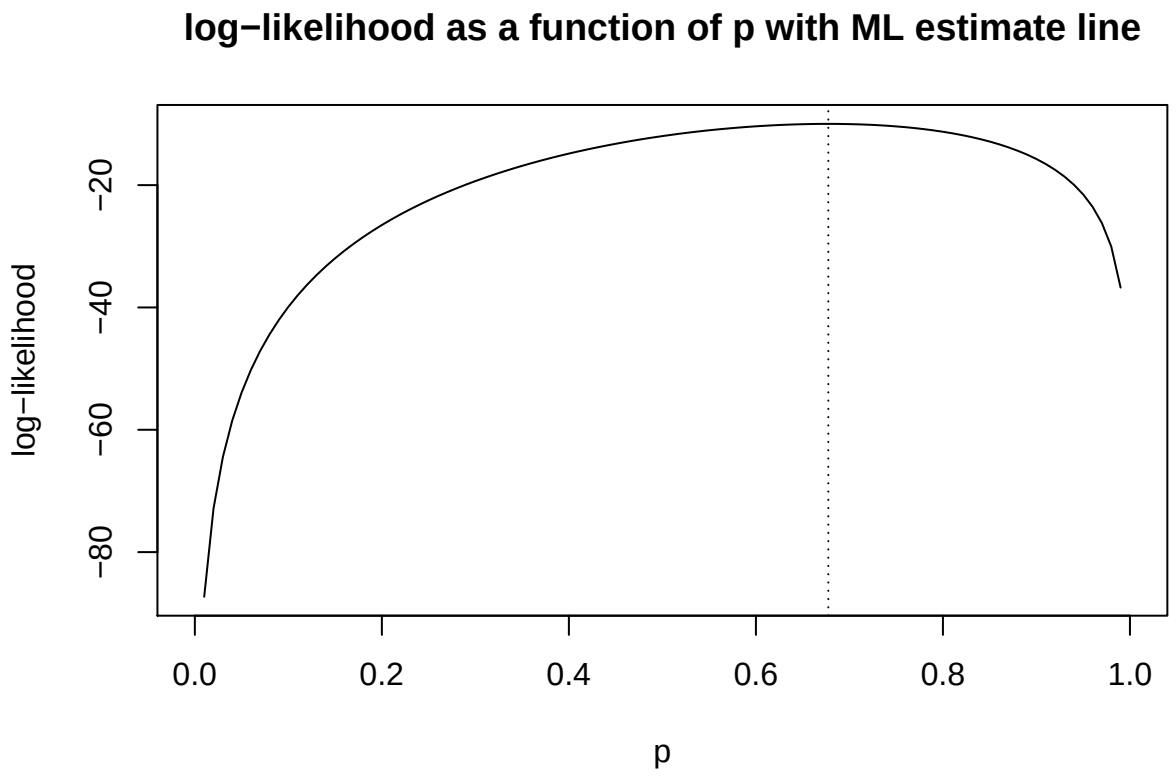
Note 2 - All of this is my own work. I took help from the provided references to verify my answer.

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

Answer -

Now we will plot log-likelihood as a function of p and draw ML estimate line.

```
# Load values of number of heads in a vector
n <- c(0,0,1,1,1,2,3,3,4,6)
# Code the function as derived - refer to image.
loglik <- function(p) sum(log(choose(n+1,n))) + log(p)*sum(n) +
                      10*log(1-p)
curve(loglik, 0, 1, xlab="p", ylab="log-likelihood",
      main="log-likelihood as a function of p with ML estimate line")
abline(v=21/31, lty=3)
```



The computed optimum value for p is = 21/31 = 0.6774194.

For verification, We can use maxLik function and verify it is truly the maximum value for the function.

Q. 4.2.d

Set derivative to 0 to find max. likelihood.

$$\frac{\partial l(p)}{\partial p} = 0 + \frac{\sum_{i=1}^{10} n_i}{\hat{p}} - \frac{10}{1-\hat{p}}$$

$$\frac{10}{1-\hat{p}} = \frac{K}{\hat{p}} \quad (K = \text{total heads received in experiment})$$

From data, $K = 3+2+6+4+6$

$K = 21$

$$10\hat{p} = 21(1-\hat{p})$$

$$10\hat{p} = 21 - 21\hat{p}$$

$$31\hat{p} = 21$$

$$\hat{p} = \frac{21}{31} = 0.6774$$

This gives us the max. likelihood of observing this data.

In general,

$$\hat{p} = \frac{\text{Total number of heads}}{\text{Total number of flips}}$$

Figure 3: Charu Q. 4.2.d - Analytical Solution

```r
# Run maxLik function on our defined log likelihood function.
m <- maxLik(loglik, start=0.5)
m
```

```
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 2 iterations
## Return code 1: gradient close to zero
## Log-Likelihood: -9.986791 (1 free parameter(s))
## Estimate(s): 0.6774194
```

The above function returns exact same value, 0.6774194. The accuracy of our calculations has been verified.

Signed Statement of Compliance:

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

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

Charudatta Jayant Deshpande

(signature)

December 10th, 2017

(date)

References -

Q 2.4 - Answering statisticsl significance -
https://measuringu.com/statistically-significant/

Q 2.4 - Answering 95% CI -
http://www.itl.nist.gov/div898/handbook/eda/section3/eda352.htm

Q3.5 - Answering 95% CI -
http://www.cyclismo.org/tutorial/R/confidence.html

Q. 4.1.a - Calculate expected payoff
http://mathforum.org/library/drmath/view/56670.html

Q 4.2c and 4.2d - for verification only.
http://classes.engr.oregonstate.edu/eecs/spring2012/cs534/notes/maximum-likelihood.pdf

By this time I had already derived my own conclusions and wanted to verify it. I verified it using maxLik function and above link.

General reference note for Q.3 -

Most of the answers for Q.3 are derived from INFX 573 class on 12/05/2017. The questions asked here are very similar to those covered in that class.