PSTAT131 Final Project - Theo Lee (6867162) and Natasha Leodjaja (8935389)

Census Data

```
state.name <- c(state.name, "District of Columbia")
state.abb <- c(state.abb, "DC")
## read in census data
census <- read_csv("/Users/theolee/Desktop/acs2017_county_data.csv") %>%
  dplyr::select(-CountyId, -ChildPoverty, -Income, -IncomeErr, -IncomePerCap, -IncomePer
CapErr) %>%
  mutate(State = state.abb[match(`State`, state.name)]) %>%
  filter(State != "PR")
```

```
## Rows: 3220 Columns: 37
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (2): State, County
## dbl (35): CountyId, TotalPop, Men, Women, Hispanic, White, Black, Native, As...
```

```
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(census)
```

```
## # A tibble: 6 x 31
##   State County   TotalPop   Men  Women Hispanic White Black Native Asian Pacific
##   <chr> <chr>       <dbl> <dbl>  <dbl>    <dbl> <dbl> <dbl>  <dbl> <dbl>   <dbl>
## 1 AL    Autauga…    55036 26899  28137      2.7  75.4  18.9    0.3   0.9       0
## 2 AL    Baldwin…   203360 99527 103833      4.4  83.1   9.5    0.8   0.7       0
## 3 AL    Barbour…    26201 13976  12225      4.2  45.7  47.8    0.2   0.6       0
## 4 AL    Bibb Co…    22580 12251  10329      2.4  74.6  22      0.4   0         0
## 5 AL    Blount …    57667 28490  29177      9    87.4   1.5    0.3   0.1       0
## 6 AL    Bullock…    10478  5616   4862      0.3  21.6  75.6    1     0.7       0
## # … with 20 more variables: VotingAgeCitizen <dbl>, Poverty <dbl>,
## #   Professional <dbl>, Service <dbl>, Office <dbl>, Construction <dbl>,
## #   Production <dbl>, Drive <dbl>, Carpool <dbl>, Transit <dbl>, Walk <dbl>,
## #   OtherTransp <dbl>, WorkAtHome <dbl>, MeanCommute <dbl>, Employed <dbl>,
## #   PrivateWork <dbl>, PublicWork <dbl>, SelfEmployed <dbl>, FamilyWork <dbl>,
## #   Unemployment <dbl>
```

Education Data

```
## read in education data
education <- read_csv("/Users/theolee/Desktop/education.csv") %>%
  filter(!is.na(`2003 Rural-urban Continuum Code`)) %>%
  filter(State != "PR") %>%
  select(-`FIPS Code`,
         -`2003 Rural-urban Continuum Code`,
         -`2003 Urban Influence Code`,
         -`2013 Rural-urban Continuum Code`,
         -`2013 Urban Influence Code`) %>%
  dplyr::rename(County = `Area name`)
```

```
## Rows: 3283 Columns: 47
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (3): FIPS Code, State, Area name
## dbl (24): 2003 Rural-urban Continuum Code, 2003 Urban Influence Code, 2013 R...
```

```
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(education)
```

```
## # A tibble: 6 x 42
##   State County   `Less than a high sch… `High school diplom… `Some college (1-3…
##   <chr> <chr>                     <dbl>                <dbl>               <dbl>
## 1 AL    Autauga…                   6611                 3757                 933
## 2 AL    Baldwin…                  18726                 8426                2334
## 3 AL    Barbour…                   8120                 2242                 581
## 4 AL    Bibb Co…                   5272                 1402                 238
## 5 AL    Blount …                  10677                 3440                 626
## 6 AL    Bullock…                   4245                  958                 305
## # … with 37 more variables: Four years of college or higher, 1970 <dbl>,
## #   Percent of adults with less than a high school diploma, 1970 <dbl>,
## #   Percent of adults with a high school diploma only, 1970 <dbl>,
## #   Percent of adults completing some college (1-3 years), 1970 <dbl>,
## #   Percent of adults completing four years of college or higher, 1970 <dbl>,
## #   Less than a high school diploma, 1980 <dbl>,
## #   High school diploma only, 1980 <dbl>, Some college (1-3 years), 1980 <dbl>,
## #   Four years of college or higher, 1980 <dbl>,
## #   Percent of adults with less than a high school diploma, 1980 <dbl>,
## #   Percent of adults with a high school diploma only, 1980 <dbl>,
## #   Percent of adults completing some college (1-3 years), 1980 <dbl>,
## #   Percent of adults completing four years of college or higher, 1980 <dbl>,
## #   Less than a high school diploma, 1990 <dbl>,
## #   High school diploma only, 1990 <dbl>,
## #   Some college or associate's degree, 1990 <dbl>,
## #   Bachelor's degree or higher, 1990 <dbl>,
## #   Percent of adults with less than a high school diploma, 1990 <dbl>,
## #   Percent of adults with a high school diploma only, 1990 <dbl>,
## #   Percent of adults completing some college or associate's degree, 1990 <dbl>,
## #   Percent of adults with a bachelor's degree or higher, 1990 <dbl>,
## #   Less than a high school diploma, 2000 <dbl>,
## #   High school diploma only, 2000 <dbl>,
## #   Some college or associate's degree, 2000 <dbl>,
## #   Bachelor's degree or higher, 2000 <dbl>,
## #   Percent of adults with less than a high school diploma, 2000 <dbl>,
## #   Percent of adults with a high school diploma only, 2000 <dbl>,
## #   Percent of adults completing some college or associate's degree, 2000 <dbl>,
## #   Percent of adults with a bachelor's degree or higher, 2000 <dbl>,
## #   Less than a high school diploma, 2015-19 <dbl>,
## #   High school diploma only, 2015-19 <dbl>,
## #   Some college or associate's degree, 2015-19 <dbl>,
## #   Bachelor's degree or higher, 2015-19 <dbl>,
## #   Percent of adults with less than a high school diploma, 2015-19 <dbl>,
## #   Percent of adults with a high school diploma only, 2015-19 <dbl>,
## #   Percent of adults completing some college or associate's degree, 2015-19 <dbl>,
## #   Percent of adults with a bachelor's degree or higher, 2015-19 <dbl>
```

Preliminary Data Analysis

1. (1 pts) Report the dimension of census. (1 pts) Are there missing values in the data set? (1 pts) Compute the total number of distinct values in State in census to verify that the data contains all states and a federal district.

```
dim(census) # dimensions
```

```
## [1] 3142   31
```

```
sum(is.na(census)) # checking for NA values
```

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

```
length(table(census$State)) # calculating the number of distinct values in state
```

```
## [1] 51
```

The dimensions of census are 3142 rows by 31 columns. There are no missing values in the data set. The total number of distinct values in State in census is 51 because it includes Puerto Rico which is a US territory.

2. (1 pts) Report the dimension of education. (1 pts) How many distinct counties contain missing values in the data set? (1 pts) Compute the total number of distinct values in County in education. (1 pts) Compare the values of total number of distinct county in education with that in census. (1 pts) Comment on your findings.

```
dim(education) # dimensions
```

```
## [1] 3143   42
```

```
sum(rowSums(is.na(education) | education == "")) # distinct counties containing NA value
s
```

```
## [1] 273
```

```
length(table(education$County)); length(table(census$County))# calculating the number of
distinct values in education
```

```
## [1] 1877
```

```
## [1] 1877
```

The dimensions of education are 3143 rows by 42 columns. There are 273 distinct counties containing missing values in the dataset. The total number of distinct counties in education and in census are the same.

Data Wrangling

3. (2 pts) Remove all NA values in education, if there is any.

```
education <- na.omit(education) # removing NA values from education
sum(is.na(education))
```

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

4. (2 pts) In education, in addition to State and County, we will start only on the following 4 features: Less than a high school diploma, 2015-19, High school diploma only, 2015-19, Some college or associate's degree, 2015-19, and Bachelor's degree or higher, 2015-19. Mutate the education dataset by selecting these 6 features only, and create a new feature which is the total population of that county.

```
# mutate education to contain 6 features
education <- education %>%
  select("State","County","Less than a high school diploma, 2015-19","High school diplom
a only, 2015-19","Some college or associate's degree, 2015-19","Bachelor's degree or hig
her, 2015-19") %>%
  mutate(CountyPopulation = rowSums(.[3:6]))
```

5. (3 pts) Construct aggregated data sets from education data: i.e., create a state-level summary into a dataset named education.state.

```
education.state <- education %>%
  group_by(State) %>%
  summarise_at(vars(-County), funs(sum))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
education.state
```

```
## # A tibble: 51 x 6
##    State `Less than a high… `High school dip… `Some college or… `Bachelor's deg…
##    <chr>                <dbl>             <dbl>             <dbl>            <dbl>
##  1 AK                   32338            126881            162816           137666
##  2 AL                  458922           1022839            993344           845772
##  3 AR                  270168            684659            593576           463236
##  4 AZ                  604935           1124129           1594817          1392598
##  5 CA                 4418675           5423462           7648680          8980726
##  6 CO                  314312            810659           1114680          1538936
##  7 CT                  232663            666828            608139           975465
##  8 DC                   44850             83185             76822           289259
##  9 DE                   66816            209449            178917           214138
## 10 FL                 1767583           4276237           4450224          4471701
## # … with 41 more rows, and 1 more variable: CountyPopulation <dbl>
```

6. (4 pts) Create a data set named state.level on the basis of education.state, where you create a new feature which is the name of the education degree level with the largest population in that state.

```
state.level <- education.state[-6]
state.level$majority <- colnames(state.level)[apply(state.level,1,which.max)]
```

```
## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion
```

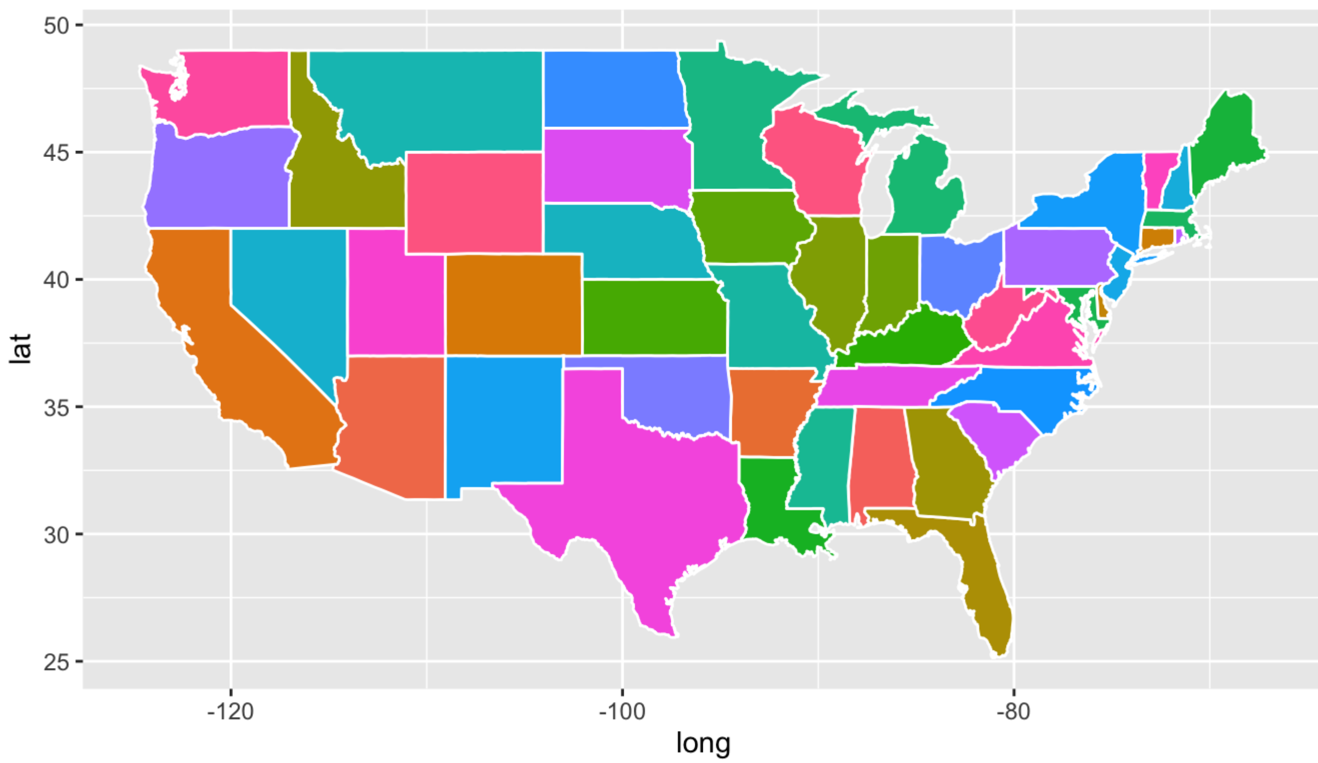
```
## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion

## Warning in FUN(newX[, i], ...): NAs introduced by coercion
```

```
state.level$majority
```

```
##  [1] "Some college or associate's degree, 2015-19"
##  [2] "High school diploma only, 2015-19"
##  [3] "High school diploma only, 2015-19"
##  [4] "Some college or associate's degree, 2015-19"
##  [5] "Bachelor's degree or higher, 2015-19"
##  [6] "Bachelor's degree or higher, 2015-19"
##  [7] "Bachelor's degree or higher, 2015-19"
##  [8] "Bachelor's degree or higher, 2015-19"
##  [9] "Bachelor's degree or higher, 2015-19"
## [10] "Bachelor's degree or higher, 2015-19"
## [11] "Bachelor's degree or higher, 2015-19"
## [12] "Bachelor's degree or higher, 2015-19"
## [13] "Some college or associate's degree, 2015-19"
## [14] "Some college or associate's degree, 2015-19"
## [15] "Bachelor's degree or higher, 2015-19"
## [16] "High school diploma only, 2015-19"
## [17] "Bachelor's degree or higher, 2015-19"
## [18] "High school diploma only, 2015-19"
## [19] "High school diploma only, 2015-19"
## [20] "Bachelor's degree or higher, 2015-19"
## [21] "Bachelor's degree or higher, 2015-19"
## [22] "Bachelor's degree or higher, 2015-19"
## [23] "Some college or associate's degree, 2015-19"
## [24] "Bachelor's degree or higher, 2015-19"
## [25] "High school diploma only, 2015-19"
## [26] "Some college or associate's degree, 2015-19"
## [27] "Some college or associate's degree, 2015-19"
## [28] "Bachelor's degree or higher, 2015-19"
## [29] "Some college or associate's degree, 2015-19"
## [30] "Some college or associate's degree, 2015-19"
## [31] "Bachelor's degree or higher, 2015-19"
## [32] "Bachelor's degree or higher, 2015-19"
## [33] "Some college or associate's degree, 2015-19"
## [34] "Some college or associate's degree, 2015-19"
## [35] "Bachelor's degree or higher, 2015-19"
## [36] "High school diploma only, 2015-19"
## [37] "High school diploma only, 2015-19"
## [38] "Some college or associate's degree, 2015-19"
## [39] "High school diploma only, 2015-19"
## [40] "Bachelor's degree or higher, 2015-19"
## [41] "Some college or associate's degree, 2015-19"
## [42] "Some college or associate's degree, 2015-19"
## [43] "High school diploma only, 2015-19"
## [44] "Bachelor's degree or higher, 2015-19"
## [45] "Some college or associate's degree, 2015-19"
## [46] "Bachelor's degree or higher, 2015-19"
## [47] "Bachelor's degree or higher, 2015-19"
## [48] "Bachelor's degree or higher, 2015-19"
## [49] "Some college or associate's degree, 2015-19"
## [50] "High school diploma only, 2015-19"
## [51] "Some college or associate's degree, 2015-19"
```

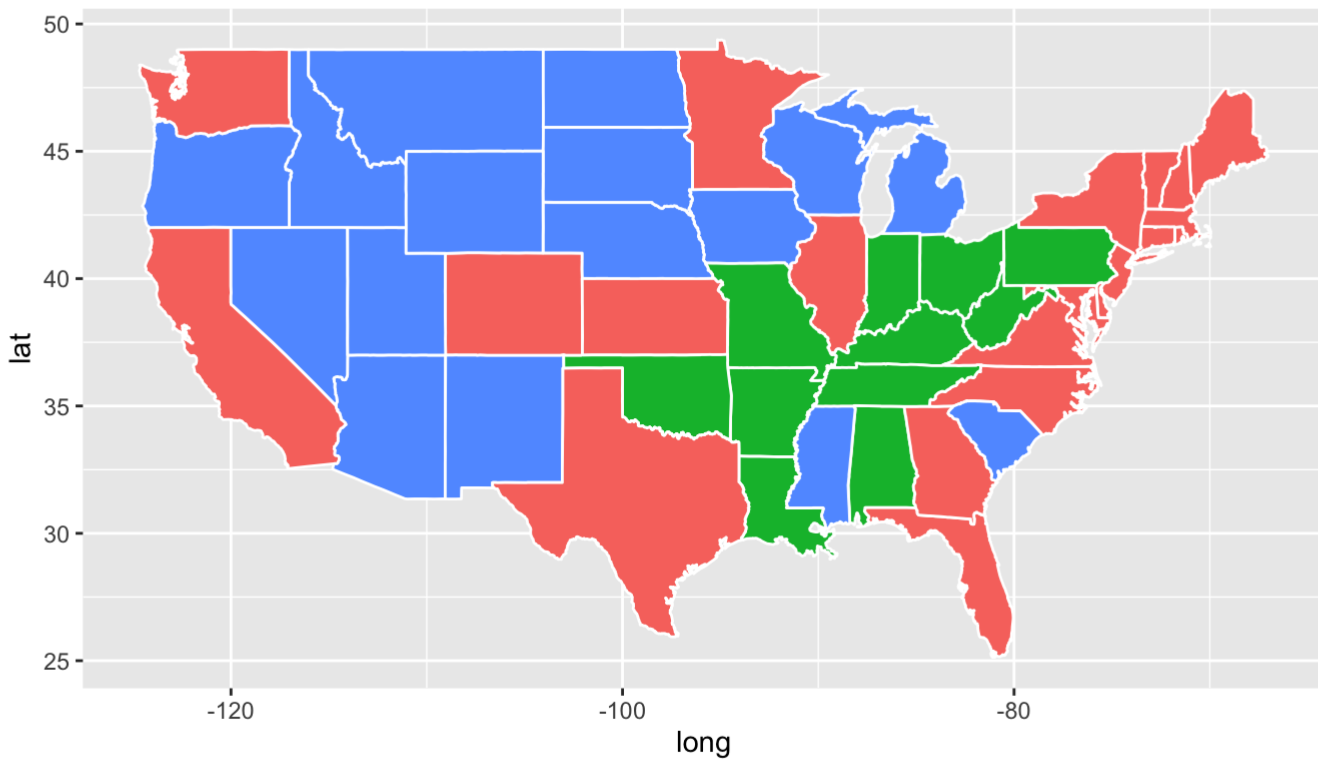
Visualization

```
# the variable states contain information to draw white polyogons
# fill-colors are determined by region
states <- map_data("state")

ggplot(data = states) +
  geom_polygon(aes(x = long, y = lat, fill = region, group = group),
               color = "white") +
  coord_fixed(1.3) +
  guides(fill=FALSE)  # color legend is unnecessary for this example and takes too long
```



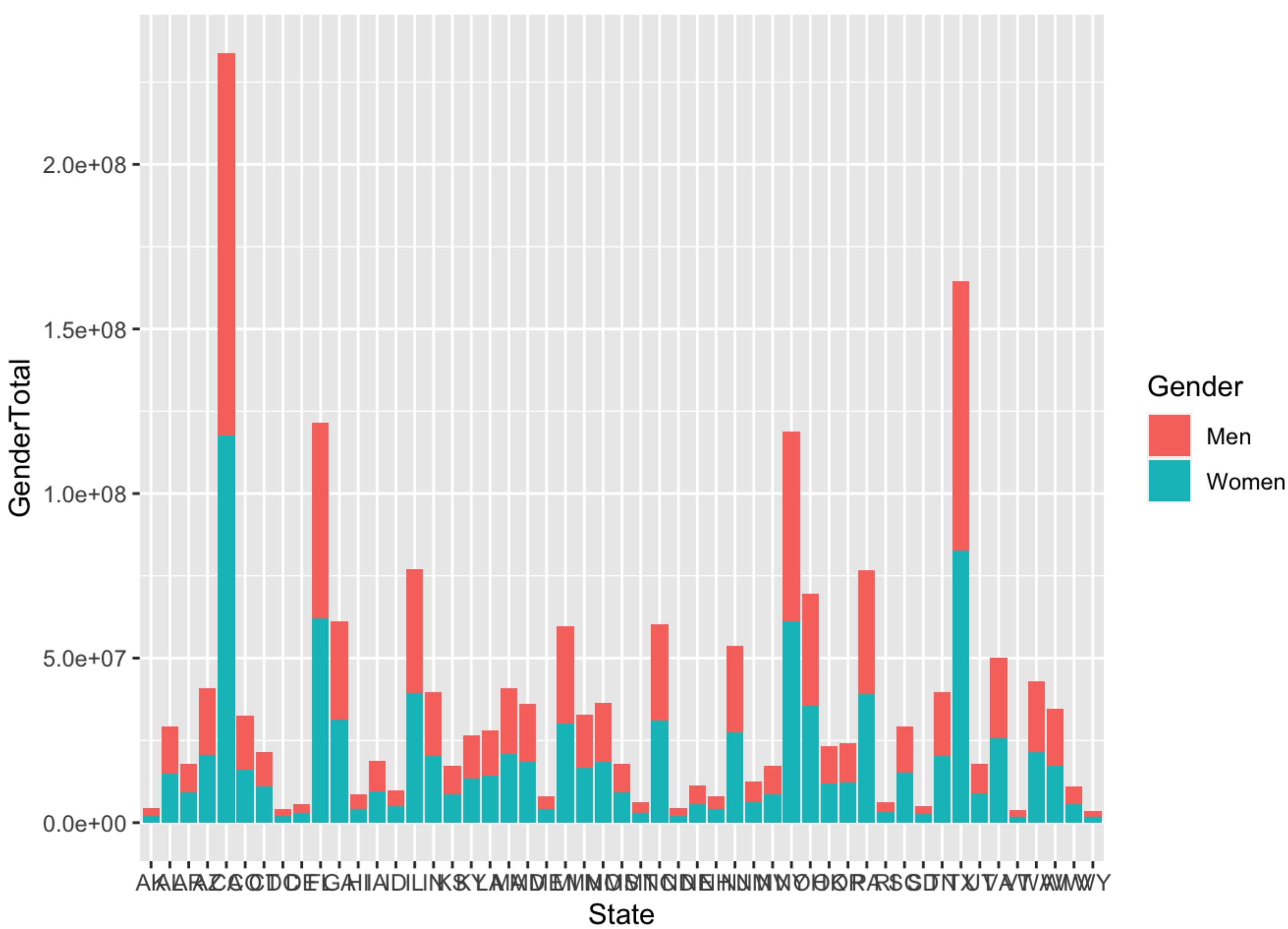
7. (6 pts) Now color the map (on the state level) by the education level with highest population for each state. Show the plot legend. First, combine states variable and state.level we created earlier using left_join(). Note that left_join() needs to match up values of states to join the tables. A call to left_join() takes all the values from the first table and looks for matches in the second table. If it finds a match, it adds the data from the second table; if not, it adds missing values. Here, we'll be combing the two data sets based on state name. However, the state names in states and state.level can be in different formats: check them! Before using left_join(), use certain transform to make sure the state names in the two data sets: states (for map drawing) and state.level (for coloring) are in the same formats. Then left_join().

```
# abbreviate state names
states['region'] <- state2abbr(states$region)

# rename region to State in order to left join
states <- states %>% rename_at('region',~'State')

# left join both datasets
states <- states %>% left_join(state.level, by='State')

# states for map drawing and state.level for coloring
ggplot(data = states) +
  geom_polygon(aes(x = long, y = lat, fill = majority, group = group),
               color = "white") +
  coord_fixed(1.3) +
  guides(fill=FALSE)
```

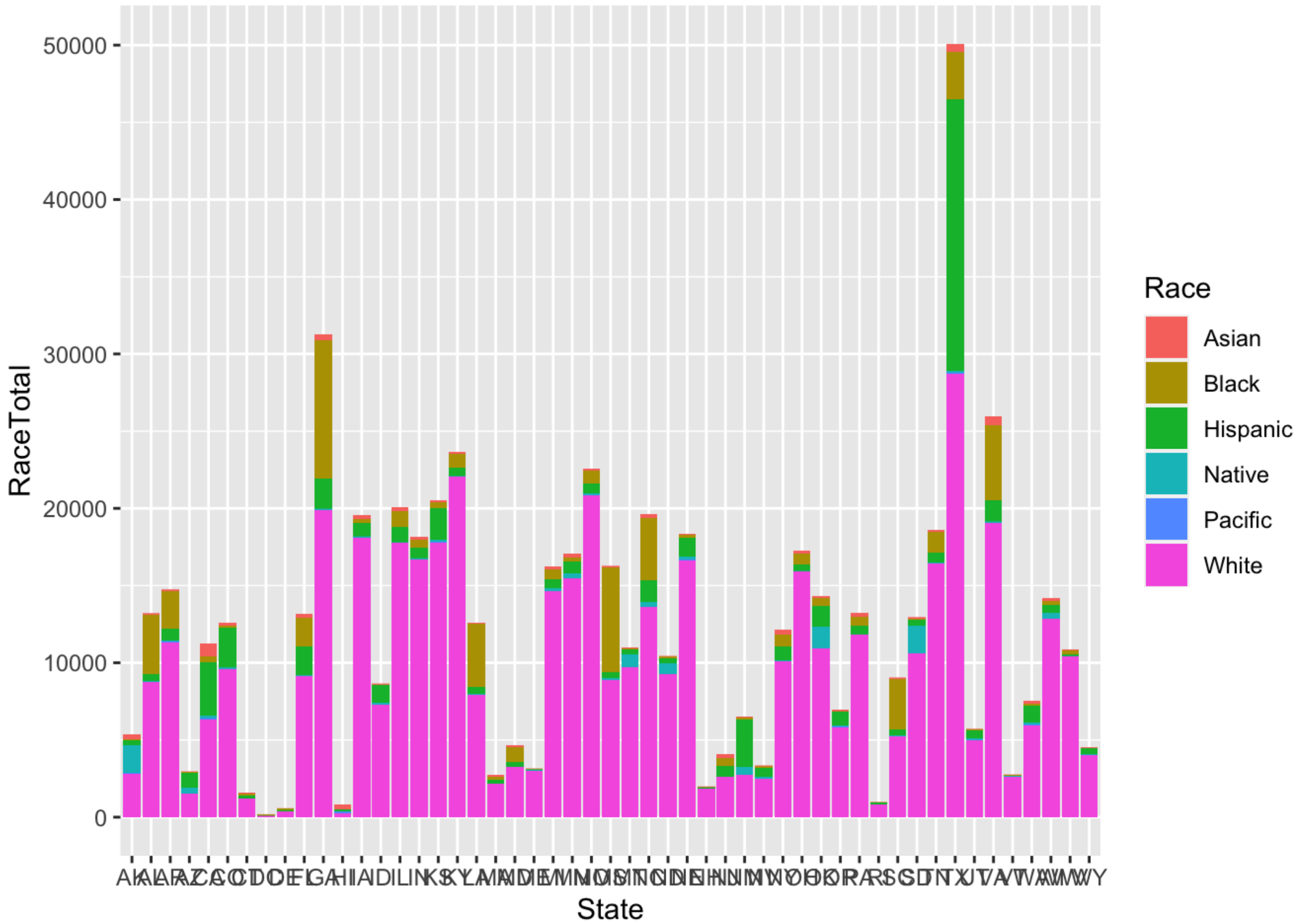


8. (6 pts) (Open-ended) Create a visualization of your choice using census data. Use this R graph gallery for ideas and inspiration.

```
# separate dataset into several datasets
# group them by State and get the total sum
q8 <- census %>%
  group_by(State) %>%
  summarise_at(vars(-County), funs(sum))

# group by gender
p1 <- q8 %>%
  pivot_longer('Men':'Women', names_to = "Gender", values_to = "GenderTotal")

# group by race
p2 <- p1 %>%
  pivot_longer('Hispanic':'Pacific', names_to = "Race", values_to = "RaceTotal")

# Stacked
par(mfrow=c(1,2))
ggplot(p2, aes(fill=Gender, y=GenderTotal, x=State)) +
    geom_bar(position="stack", stat="identity")
```




```
ggplot(p2, aes(fill=Race, y=RaceTotal, x=State)) +
    geom_bar(position="stack", stat="identity")
```



9. The census data contains county-level census information. In this problem, we clean and aggregate the information as follows. (4 pts) Start with census, filter out any rows with missing values, convert {Men, Employed, VotingAgeCitizen} attributes to percentages, compute Minority attribute by combining {Hispanic, Black, Native, Asian, Pacific}, remove these variables after creating Minority, remove {Walk, PublicWork, Construction, Unemployment}. (Note that many columns are perfectly collineared, in which case one column should be deleted.)

```
census2 <- na.omit(census)
census2 <- transform(census2,Men=census2$Men/census2$TotalPop)
census2 <- transform(census2,Employed=census2$Employed/census2$TotalPop)
census2 <- transform(census2,VotingAgeCitizen=census2$VotingAgeCitizen/census2$TotalPop)
census2$minority <- census2$Hispanic+census2$Black+census2$Native+census2$Asian+census2
$Pacific
census2 <- select(census2,-c(Hispanic, Black, Native, Asian, Pacific, Walk, PublicWork,
Construction, Unemployment, Women, White))
# taking out women and white features to reflect percentage men and minority
census.clean <- census2
```

10. (1 pts) Print the first 5 rows of census.clean

```
head(census.clean, 5)
```

```
##   State         County TotalPop       Men VotingAgeCitizen Poverty Professional
## 1    AL Autauga County    55036 0.4887528        0.7452576    13.7         35.3
## 2    AL Baldwin County   203360 0.4894129        0.7640441    11.8         35.7
## 3    AL Barbour County    26201 0.5334148        0.7735964    27.2         25.0
## 4    AL    Bibb County    22580 0.5425598        0.7821966    15.2         24.4
## 5    AL  Blount County    57667 0.4940434        0.7372154    15.6         28.5
##   Service Office Production Drive Carpool Transit OtherTransp WorkAtHome
## 1    18.0   23.2       15.4  86.0     9.6     0.1         1.3        2.5
## 2    18.2   25.6       10.8  84.7     7.6     0.1         1.1        5.6
## 3    16.8   22.6       24.1  83.4    11.1     0.3         1.7        1.3
## 4    17.6   19.7       22.4  86.4     9.5     0.7         1.7        1.5
## 5    12.9   23.3       19.5  86.8    10.2     0.1         0.4        2.1
##   MeanCommute  Employed PrivateWork SelfEmployed FamilyWork minority
## 1        25.8 0.4381132        74.1          5.6        0.1     22.8
## 2        27.0 0.4402390        80.7          6.3        0.1     15.4
## 3        23.4 0.3388420        74.1          6.5        0.3     52.8
## 4        30.0 0.3618689        76.0          6.3        0.3     24.8
## 5        35.0 0.3707493        83.9          4.0        0.1     10.9
```

Dimensionality reduction

11. Run PCA for the cleaned county level census data (with State and County excluded). (2 pts) Save the first two principle components PC1 and PC2 into a two-column data frame, call it pc.county. (2 pts) Discuss whether you chose to center and scale the features before running PCA and the reasons for your choice. (2 pts) What are the three features with the largest absolute values of the first principal component? (2 pts) Which features have opposite signs and what does that mean about the correlation between these features?

```
# remove state and county
census.clean2 <- subset(census.clean,select=-c(State, County))
pr.out = prcomp(census.clean2, center=TRUE, scale=TRUE) # center and scale features

# save PC1 and PC2 into a 2 col dataframe
pc.county <- data.frame(pr.out$x[,1],pr.out$x[,2])

# largest abs value of PC1
head(sort(abs(pr.out$rotation[,1]),decreasing=TRUE))
```

```
##   WorkAtHome SelfEmployed        Drive Professional   Production  PrivateWork
##    0.4267336    0.3605124    0.3578110    0.3446943    0.2916693    0.2701238
```

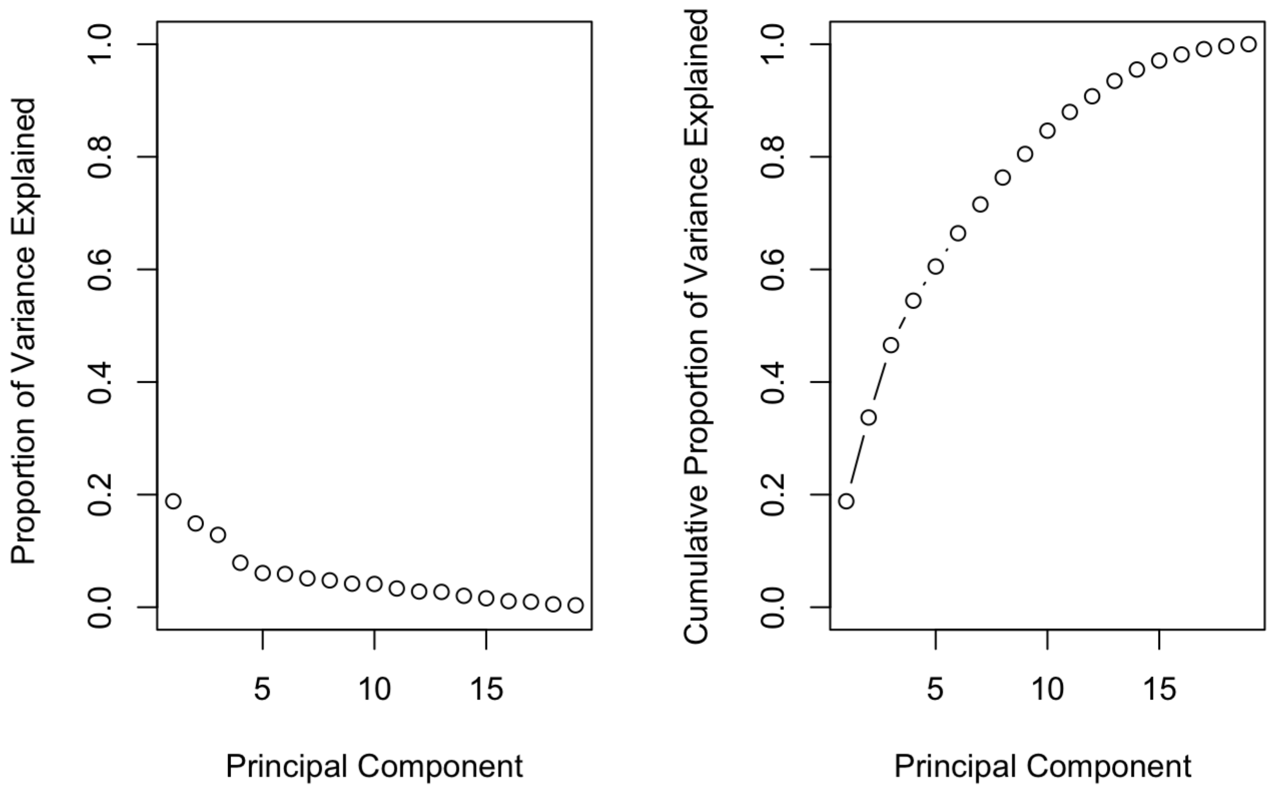
```
# features have opposite signs
pr.out$rotation[,1]
```

```
##        TotalPop             Men VotingAgeCitizen          Poverty
##      0.02647537      0.06734237       0.02508638      -0.24039363
##    Professional         Service           Office       Production
##      0.34469432     -0.09122182      -0.14792201      -0.29166926
##           Drive         Carpool          Transit      OtherTransp
##     -0.35781105     -0.06792515       0.10831749       0.11448636
##      WorkAtHome     MeanCommute         Employed      PrivateWork
##      0.42673365     -0.17805008       0.26003242      -0.27012383
##    SelfEmployed      FamilyWork         minority
##      0.36051238      0.21732612      -0.11484242
```

WorkAtHome, SelfEmployed and Drive are the three features with the largest absolute values of the first principal component. We chose to center and scale our variables to minimize differences between the way they are recorded (where some are percentages, others are hard numbers). Features that have opposite signs in the first PC include Poverty, Service, Office, Production, Drive, Carpool, MeanCommute, PrivateWork, and minority. The opposite sign implies they are negatively correlated with the first PC.

12. (2 pts) Determine the number of minimum number of PCs needed to capture 90% of the variance for the analysis. (2 pts) Plot proportion of variance explained (PVE) and cumulative PVE.

```
pr.out = prcomp(census.clean[-c(1:2)], center=TRUE, scale=TRUE)
pr.var=pr.out$sdev^2 # proportion of variance explained by each PC
pve=(pr.var)/(sum(pr.var))

par(mfrow=c(1,2))
plot(pve, xlab="Principal Component",
ylab="Proportion of Variance Explained ", ylim=c(0,1),type='b')
plot(cumsum(pve), xlab="Principal Component ",
ylab=" Cumulative Proportion of Variance Explained ", ylim=c(0,1), type='b')
```



We need roughly 12 PCs to capture 90% of the variance.

Clustering

13. (2 pts) With census.clean (with State and County excluded), perform hierarchical clustering with complete linkage. (2 pts) Cut the tree to partition the observations into 10 clusters. (2 pts) Re-run the hierarchical clustering algorithm using the first 2 principal components from pc.county as inputs instead of the original features. (2 pts) Compare the results and comment on your observations. For both approaches investigate the cluster that contains Santa Barbara County. (2 pts) Which approach seemed to put Santa Barbara County in a more appropriate clusters? Comment on what you observe and discuss possible explanations for these observations.

```
# perform hierarchical clustering with complete linkage
cen.dist = dist(census.clean)
```

```
## Warning in dist(census.clean): NAs introduced by coercion
```

```
set.seed(1)
cen.hclust = hclust(cen.dist, method='complete') # complete linkage

# cut the tree to partition into 10 clusters
cen.clus = cutree(cen.hclust, 10)

# rerun hierarchical clustering using first 2 PCA from pc.county as inputs
pc.dist = dist(pc.county)
set.seed(1)
pc.hclust = hclust(pc.dist)
pc.clus = cutree(pc.hclust, 10)

# compare results and comment on your observations
table(cen.clus)
```

```
## cen.clus
##    1    2    3    4    5    6    7    8    9   10
## 3034   69    2    9   12    1    2    5    7    1
```

```
table(pc.clus)
```

```
## pc.clus
##    1    2    3    4    5    6    7    8    9   10
## 1734  272   42   79  772   19  109    1  100   14
```

```
# investigate cluster that contains SB county (index 228)
cen.clus[228] # 1 cluster
```

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

```
pc.clus[228] # 5 clusters
```

```
## 228
##   5
```

It seems that cen.clus has a higher first cluster observation as compared to pc.clus. This is because we're computing the clusters for all data instead of using PC1 and PC2. When investigating clusters that contains Santa Barbara county, cen.clus produced 1 cluster while pc.clus produced 5 clusters. pc.clus approach seemed to put Santa Barabra County in a more appropriate cluster because it clusters PC1 and PC2 which is more informative than clustering all data.

Modeling

```
# we join the two datasets
all <- census.clean %>%
  left_join(education, by = c("State"="State", "County"="County")) %>%
  na.omit
```

14. (4 pts) Transform the variable Poverty into a binary categorical variable with two levels: 1 if Poverty is greater than 20, and 0 if Poverty is smaller than or equal to 20. Remove features that you think are uninformative in classfication tasks.

```
# partition dataset into 80% training and 20% testing
set.seed(123)
n <- nrow(all)
idx.tr <- sample.int(n, 0.8*n)
all.tr <- all[idx.tr, ]
all.te <- all[-idx.tr, ]

# 10 cross validation folds
set.seed(123)
nfold <- 10
folds <- sample(cut(1:nrow(all.tr), breaks=nfold, labels=FALSE))

# error rate function
calc_error_rate = function(predicted.value, true.value){
  return(mean(true.value!=predicted.value))
}

# records is used to record the classification performance of
# each method in the subsequent problems
records = matrix(NA, nrow=3, ncol=2)
colnames(records) = c("train.error","test.error")
rownames(records) = c("tree","logistic","lasso")
```

```
# transforming poverty into a binary categorical variable
all.tr = all.tr %>%
  mutate(Poverty=as.factor(ifelse(Poverty>20,"1","0")))
all.te = all.te %>%
  mutate(Poverty=as.factor(ifelse(Poverty>20,"1","0")))

# removing redundant features
all.tr <- select(all.tr,-c(VotingAgeCitizen,Transit,OtherTransp,MeanCommute))
all.te <- select(all.te,-c(VotingAgeCitizen,Transit,OtherTransp,MeanCommute))
```

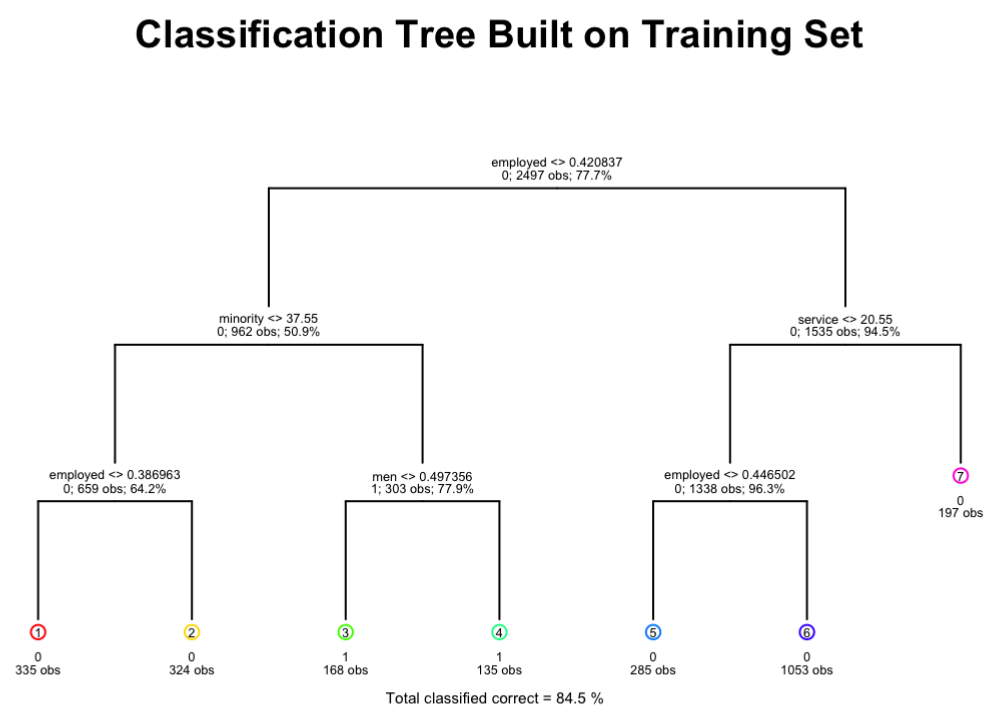
Classification

15. Decision tree: (2 pts) train a decision tree by cv.tree(). (2 pts) Prune tree to minimize misclassification error. Be sure to use the folds from above for cross-validation. (2 pts) Visualize the trees before and after pruning. (1 pts) Save training and test errors to records object. (2 pts) Interpret and discuss the results of the decision tree analysis. (2 pts) Use this plot to tell a story about Poverty.

```
# removing spaces in features for use with cv.out()
all2.tr <- clean_names(all.tr)
all2.te <- clean_names(all.te)

# define the true labels of the test cases
poverty.test <- all2.te$poverty

tree.all2 <- tree(poverty~.,data=all2.tr)
```

```
## Warning in tree(poverty ~ ., data = all2.tr): NAs introduced by coercion
```

```
# Plot the tree
draw.tree(tree.all2, nodeinfo=TRUE, cex = 0.4)
title("Classification Tree Built on Training Set")
```




```
# Set random seed
set.seed(3)
# K-Fold cross validation
cv = cv.tree(tree.all2, FUN=prune.misclass, K=folds) # Print out cv
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
## Warning in tree(model = m[rand != i, , drop = FALSE]): NAs introduced by
## coercion
```

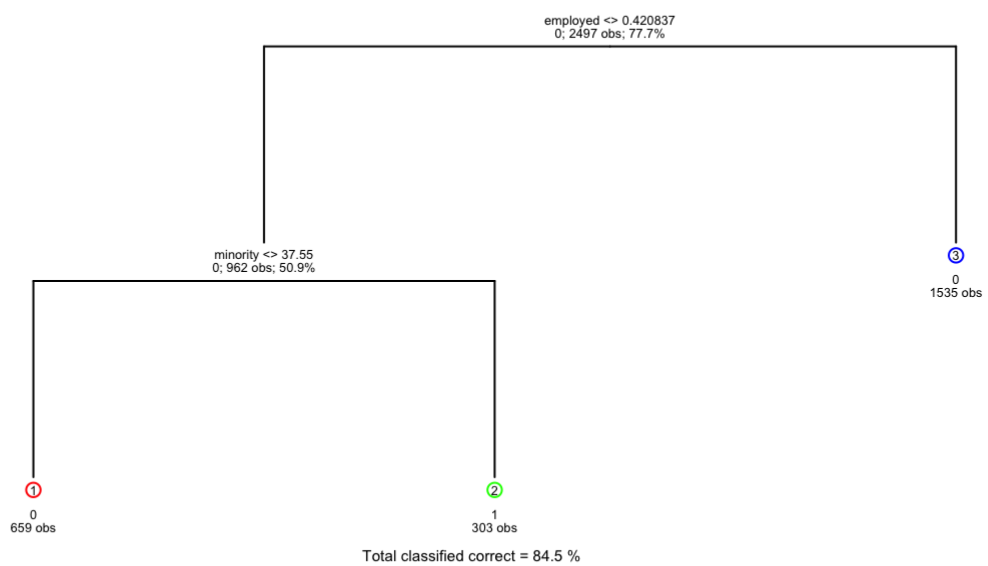
```
## Warning in pred1.tree(tree, tree.matrix(nd)): NAs introduced by coercion
```

```
best.cv = min(cv$size[cv$dev == min(cv$dev)])
best.cv
```

```
## [1] 3
```

```
pruned.tree <- prune.misclass(tree.all2,best.cv)
draw.tree(pruned.tree, nodeinfo=TRUE, cex = 0.4)
title("Pruned tree of size 3")
```

Pruned tree of size 3




```
set.seed(123)
# unpruned tree
tr.unpruned = predict(tree.all2, all2.tr, type = "class")
```

```
## Warning in pred1.tree(object, tree.matrix(newdata)): NAs introduced by coercion
```

```
ts.unpruned = predict(tree.all2, all2.te, type = "class")
```

```
## Warning in pred1.tree(object, tree.matrix(newdata)): NAs introduced by coercion
```

```
# calculate training error and test error
tr.unpruned.err <- calc_error_rate(tr.unpruned,all2.tr$poverty)
ts.unpruned.err <- calc_error_rate(ts.unpruned,all2.te$poverty)
tr.unpruned.err;ts.unpruned.err
```

```
## [1] 0.1553865
```

```
## [1] 0.168
```

```
# pruned tree
tr.pruned = predict(pruned.tree, all2.tr, type = "class")
```

```
## Warning in pred1.tree(object, tree.matrix(newdata)): NAs introduced by coercion
```

```
ts.pruned = predict(pruned.tree, all2.te, type = "class")
```

```
## Warning in pred1.tree(object, tree.matrix(newdata)): NAs introduced by coercion
```

```
# calculate training error and test error
tr.pruned.err <- calc_error_rate(tr.pruned,all2.tr$poverty)
ts.pruned.err <- calc_error_rate(ts.pruned,all2.te$poverty)

# put the values into records table
records[1,1] <- tr.pruned.err
records[1,2] <- ts.pruned.err
records
```

```
##          train.error test.error
## tree       0.1553865      0.168
## logistic          NA         NA
## lasso             NA         NA
```

The tree with 3 terminal nodes results in the lowest error. The test error rate for the training dataset is 0.16 and the test error rate for the testing dataset is 0.17 after pruning (producing a lower test error where we trim the tree to a pre-determined size). We can see from the tree that people who are a minority and employed have the same poverty rate of people who are a minority and unemployed. Whereas people who are self employed are less likely to fall under poverty.

16. (2 pts) Run a logistic regression to predict Poverty in each county. (1 pts) Save training and test errors to records variable. (1 pts) What are the significant variables? (1 pts) Are they consistent with what you saw in decision tree analysis? (2 pts) Interpret the meaning of a couple of the significant coefficients in terms of a unit change in the variables.

```
set.seed(123)
n <- nrow(all)
idx.tr <- sample.int(n, 0.8*n)
all.tr <- all[idx.tr, ]
all.te <- all[-idx.tr, ]
# define the true labels of the test cases
poverty.test <- all.te$Poverty

# transforming poverty into a binary categorical variable
all.tr = all.tr %>%
  mutate(Poverty=as.factor(ifelse(Poverty>20,"1","0")))
all.te = all.te %>%
  mutate(Poverty=as.factor(ifelse(Poverty>20,"1","0")))

all.tr <- select(all.tr,-c(State,County))
all.te <- select(all.te,-c(State,County))

# logistic regression on training data to predict poverty
glm.fit = glm(Poverty~. , data=all.tr, family=binomial)
```

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

```
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Poverty ~ ., family = binomial, data = all.tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3122  -0.4188  -0.1575  -0.0029   3.4222
##
## Coefficients: (1 not defined because of singularities)
##                                                Estimate Std. Error z value
## (Intercept)                                   2.751e+01  4.409e+00   6.239
## TotalPop                                      1.579e-04  1.992e-05   7.927
## Men                                          -3.468e+01  2.977e+00 -11.649
## VotingAgeCitizen                              4.438e+00  1.975e+00   2.247
## Professional                                  5.262e-02  2.539e-02   2.073
## Service                                       9.230e-02  2.892e-02   3.192
## Office                                        9.219e-03  3.070e-02   0.300
## Production                                    8.075e-02  2.316e-02   3.487
## Drive                                        -5.084e-02  2.984e-02  -1.704
## Carpool                                      -1.510e-03  3.736e-02  -0.040
## Transit                                       9.689e-02  6.458e-02   1.500
## OtherTransp                                  -1.171e-01  6.760e-02  -1.732
## WorkAtHome                                   -1.293e-01  4.826e-02  -2.679
## MeanCommute                                  -2.794e-02  1.638e-02  -1.706
## Employed                                     -2.975e+01  1.977e+00 -15.048
## PrivateWork                                  -2.587e-02  1.672e-02  -1.548
## SelfEmployed                                 -4.017e-02  3.195e-02  -1.257
## FamilyWork                                   -1.311e-01  1.816e-01  -0.722
## minority                                      3.736e-02  4.838e-03   7.722
## `Less than a high school diploma, 2015-19`    -1.926e-04  3.707e-05  -5.196
## `High school diploma only, 2015-19`           -2.048e-04  3.035e-05  -6.747
## `Some college or associate's degree, 2015-19` -3.545e-04  4.581e-05  -7.738
## `Bachelor's degree or higher, 2015-19`        -2.111e-04  3.203e-05  -6.589
## CountyPopulation                                     NA         NA      NA
##                                              Pr(>|z|)
## (Intercept)                                  4.40e-10 ***
## TotalPop                                     2.24e-15 ***
## Men                                           < 2e-16 ***
## VotingAgeCitizen                             0.024650 *
## Professional                                 0.038187 *
## Service                                      0.001414 **
## Office                                       0.763975
## Production                                   0.000489 ***
## Drive                                        0.088423 .
## Carpool                                      0.967751
## Transit                                      0.133519
## OtherTransp                                  0.083352 .
## WorkAtHome                                   0.007389 **
## MeanCommute                                  0.087947 .
## Employed                                      < 2e-16 ***
## PrivateWork                                  0.121737
## SelfEmployed                                 0.208652
```

```
## FamilyWork                                        0.470373
## minority                                          1.15e-14 ***
## `Less than a high school diploma, 2015-19`    2.04e-07 ***
## `High school diploma only, 2015-19`           1.51e-11 ***
## `Some college or associate's degree, 2015-19` 1.01e-14 ***
## `Bachelor's degree or higher, 2015-19`        4.43e-11 ***
## CountyPopulation                                    NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2650.6  on 2496  degrees of freedom
## Residual deviance: 1366.3  on 2474  degrees of freedom
## AIC: 1412.3
##
## Number of Fisher Scoring iterations: 9
```

```
# estimated probability
prob.train <- predict(glm.fit, all.tr, type="response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
prob.test <- predict(glm.fit, all.te, type="response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
prob.train = ifelse(prob.train > 0.5, "1", "0")
prob.test = ifelse(prob.test > 0.5,"1", "0")

records[2,1] <- calc_error_rate(prob.train,all.tr$Poverty)
records[2,2] <- calc_error_rate(prob.test,all.te$Poverty)
records
```

```
##          train.error test.error
## tree       0.1553865     0.1680
## logistic   0.1233480     0.1248
## lasso             NA         NA
```

The results we get for logistic regression is significantly better than the ones we get from decision tree. The training error is 0.12 whereas the test error is 0.12, both are lower than decision tree error rates. When running the summary for the logistic regression model, we can see that education level, employment status and production plays a very significant role in terms of predicting poverty rate.

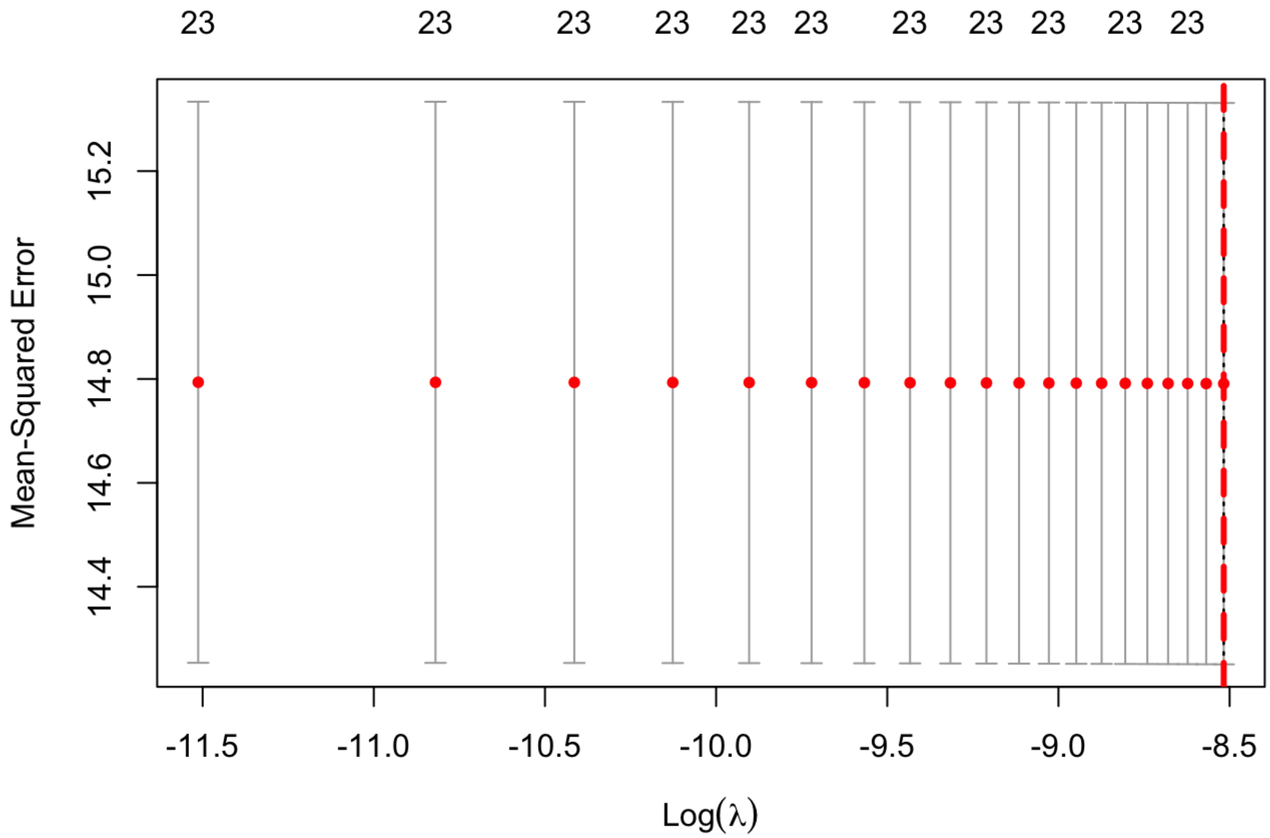
17. You may notice that you get a warning glm.fit: fitted probabilities numerically 0 or 1 occurred. As we discussed in class, this is an indication that we have perfect separation (some linear combination of

variables perfectly predicts the winner). This is usually a sign that we are overfitting. One way to control overfitting in logistic regression is through regularization.

(3 pts) Use the cv.glmnet function from the glmnet library to run a 10-fold cross validation and select the best regularization parameter for the logistic regression with LASSO penalty. Set lambda = seq(1, 20) * 1e-5 in cv.glmnet() function to set pre-defined candidate values for the tuning parameter lambda.

(1 pts) What is the optimal value of lambda in cross validation? (1 pts) What are the non-zero coefficients in the LASSO regression for the optimal value of lambda? (1 pts) How do they compare to the unpenalized logistic regression? (1 pts) Comment on the comparison. (1 pts) Save training and test errors to the records variable.

```
set.seed(123)
n <- nrow(all)
idx.tr <- sample.int(n, 0.8*n)
all.tr <- all[idx.tr, ]

df <- all.tr %>% select(-c(County,State))
idx.tr <- sample.int(n, 0.8*n)
train <- df[idx.tr, ]
test <- df[-idx.tr, ]

train <- na.omit(train)
test <- na.omit(test)

YTrain = train$Poverty
XTrain = train %>% select(-Poverty) %>% scale(center = TRUE, scale = TRUE)

YTest = test$Poverty
XTest = test %>% select(-Poverty) %>% scale(center = TRUE, scale = TRUE)

lasso_lambda = seq(1, 20)*1e-5
lasso.mod <- glmnet(XTrain, YTrain, alpha=1)
cv.out.lasso = cv.glmnet(XTrain, YTrain, nfolds = 10, lambda=lasso_lambda)
plot(cv.out.lasso)
abline(v = log(cv.out.lasso$lambda.min), col="red", lwd=3, lty=2)
```




```
bestlam = cv.out.lasso$lambda.min

XTrain <- as.matrix(XTrain)
XTest <- as.matrix(XTest)

train.pred <- predict(lasso.mod, s=bestlam, newx=XTrain)
records[3,1] <- mean((train.pred - YTrain)^2)
test.pred <- predict(lasso.mod, s=bestlam, newx=XTest)
records[3,2] <- mean((test.pred - YTest)^2)
records
```

```
##            train.error test.error
## tree         0.1553865    0.16800
## logistic     0.1233480    0.12480
## lasso       14.3259354   14.57077
```

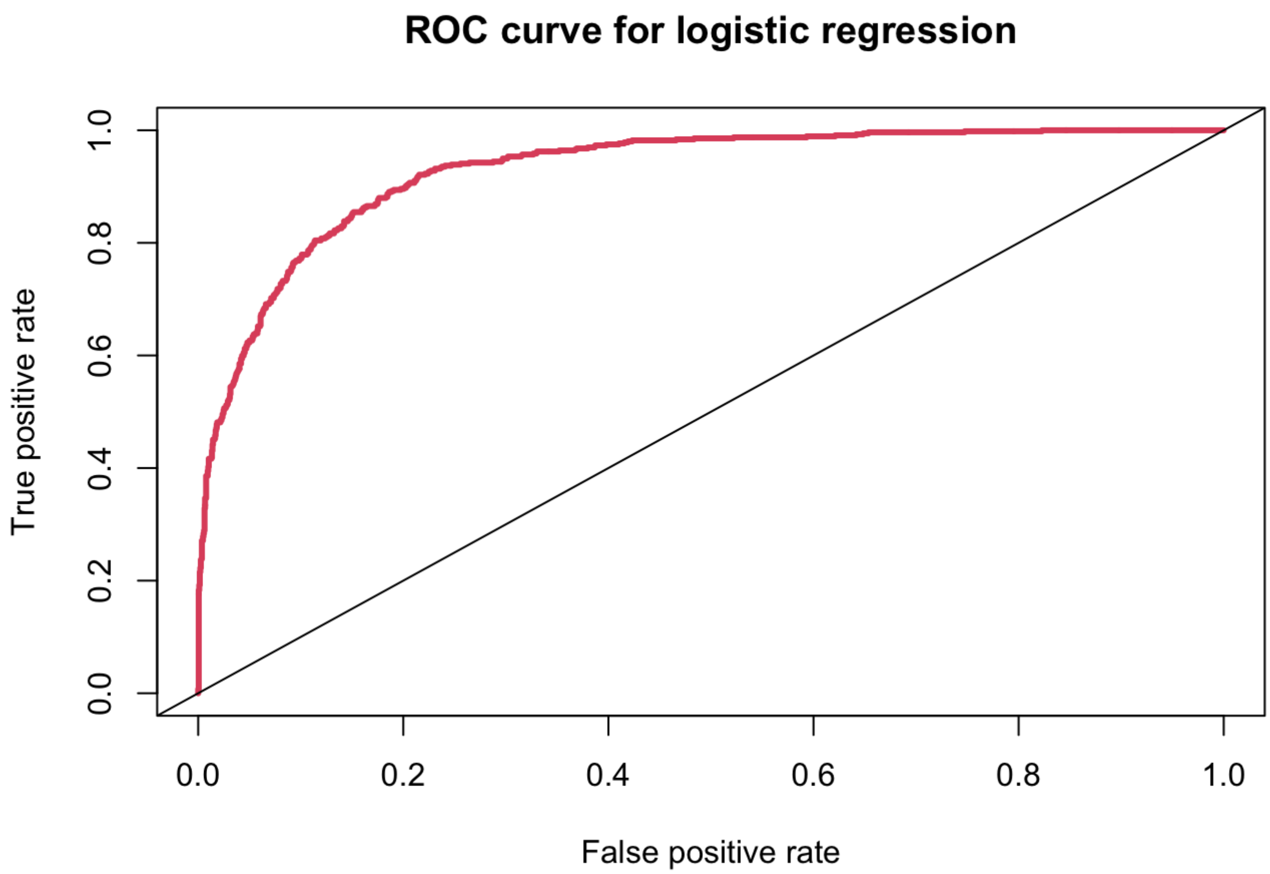
```
lasso.coef=predict(lasso.mod,type="coefficients",s=bestlam)
lasso.coef
```

```
## 24 x 1 sparse Matrix of class "dgCMatrix"
##                                                        s1
## (Intercept)                                  16.062046371
## TotalPop                                      1.190544754
## Men                                          -1.530810363
## VotingAgeCitizen                              0.232282226
## Professional                                  0.134097793
## Service                                       0.673380347
## Office                                        0.004630245
## Production                                    1.167331839
## Drive                                        -0.498151063
## Carpool                                      -0.060954673
## Transit                                       0.089553236
## OtherTransp                                  -0.014207218
## WorkAtHome                                   -0.174388216
## MeanCommute                                  -0.719270892
## Employed                                     -4.121651355
## PrivateWork                                  -0.752023659
## SelfEmployed                                 -0.527089100
## FamilyWork                                    0.101011650
## minority                                      1.629593261
## Less than a high school diploma, 2015-19      0.419766324
## High school diploma only, 2015-19            -0.070155241
## Some college or associate's degree, 2015-19 -1.595971031
## Bachelor's degree or higher, 2015-19          .
## CountyPopulation                              .
```

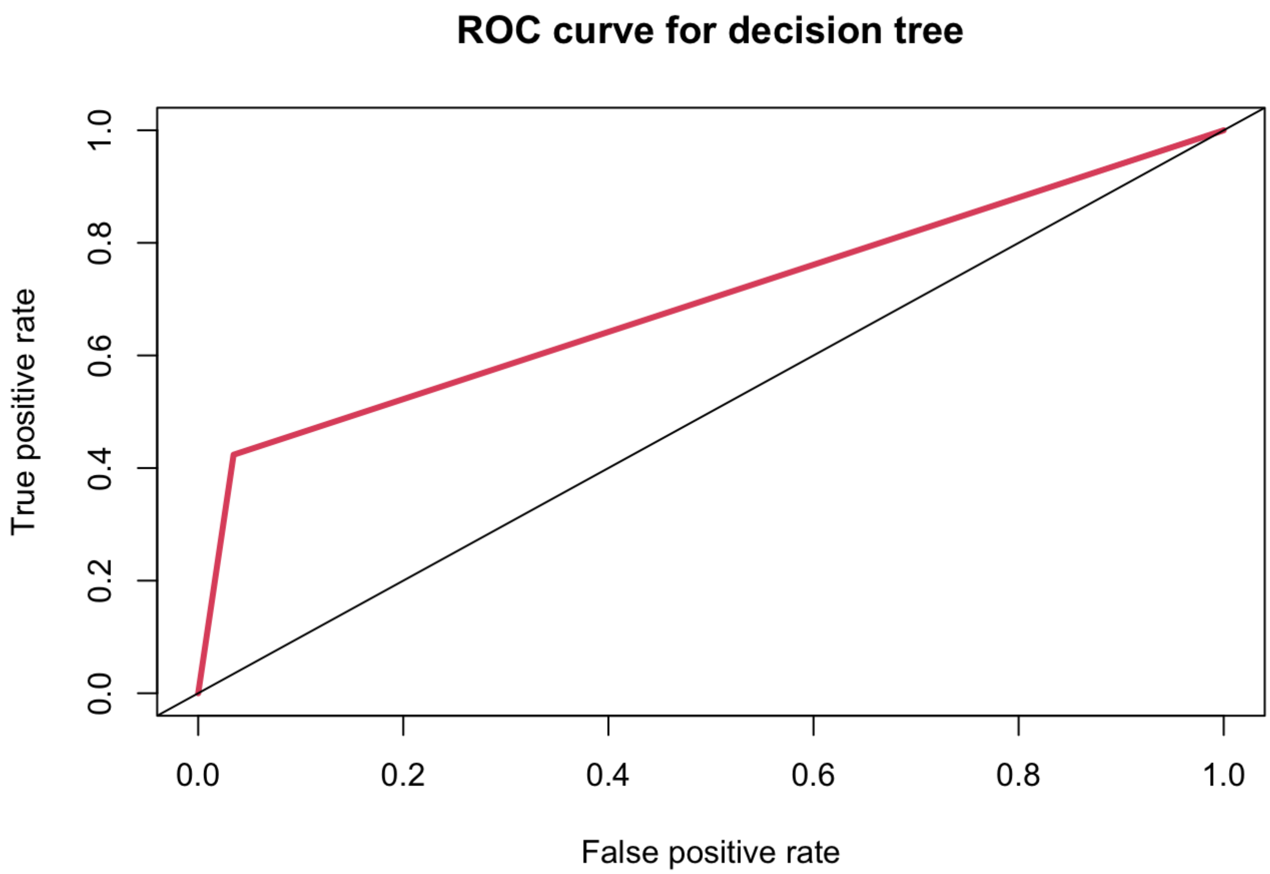
A lot of the coefficients gives non zero values such as total population, minority, production and etc. Compared to the unpenalized logistic regression, LASSO did worse as the training and testing MSE is significantly higher than the train and test error of logistic regression.

18. (6 pts) Compute ROC curves for the decision tree, logistic regression and LASSO logistic regression using predictions on the test data. Display them on the same plot. (2 pts) Based on your classification results, discuss the pros and cons of the various methods. (2 pts) Are the different classifiers more appropriate for answering different kinds of questions about Poverty?

```
all.tr <- select(all.tr,-c(State,County))
all.tr = all.tr %>%
  mutate(Poverty=as.factor(ifelse(Poverty>20,"1","0")))
poverty.test <- all.tr$Poverty
# generating ROC curve for logistic regression
prob.training = predict(glm.fit, type="response")
prediction.log = prediction(as.numeric(prob.training),as.numeric(poverty.test))
perf.log = performance(prediction.log, measure="tpr", x.measure="fpr")
plot(perf.log, col=2, lwd=3, main="ROC curve for logistic regression")
abline(0,1)
```




```
# generating ROC curve for decision tree
pruned <- predict(pruned.tree,type="class")
prediction.tree = prediction(as.numeric(pruned),as.numeric(poverty.test))
perf.log = performance(prediction.tree, measure="tpr", x.measure="fpr")
plot(perf.log, col=2, lwd=3, main="ROC curve for decision tree")
abline(0,1)
```




```
# generating ROC curve for lasso
# lassoed <- predict(,type="class")
# prediction.lasso = prediction(as.numeric(),as.numeric(poverty.test))
# perf.log = performance(prediction.tree, measure="tpr", x.measure="fpr")
# plot(perf.log, col=2, lwd=3, main="ROC curve for lasso")
# abline(0,1)
```

Taking it further

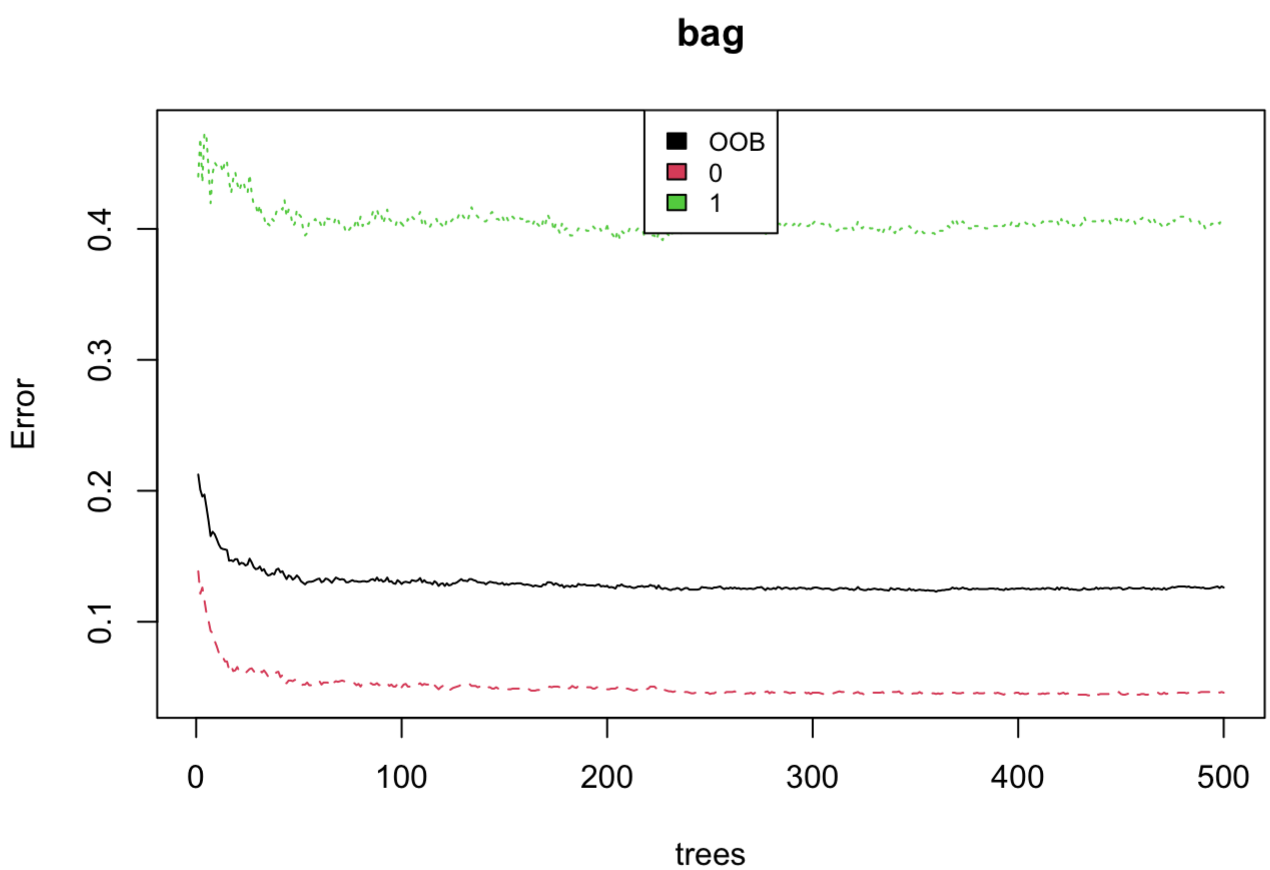
19. (9 pts) Explore additional classification methods. Consider applying additional two classification methods from KNN, LDA, QDA, SVM, random forest, boosting, neural networks etc. (You may research and use methods beyond those covered in this course). How do these compare to the tree method, logistic regression, and the lasso logistic regression?

Method 1: Bagging and Random Forest

```
bag = randomForest(poverty ~ ., data=all2.tr,importance=TRUE)
bag
```

```
##
## Call:
##  randomForest(formula = poverty ~ ., data = all2.tr, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 12.62%
## Confusion matrix:
##      0   1 class.error
## 0 1851  89  0.04587629
## 1  226 331  0.40574506
```

```
plot(bag)
legend("top", colnames(bag$err.rate),col=1:4,cex=0.8,fill=1:4)
```




```
yhat.bag = predict(bag, newdata = all2.te, type = "response")
test.bag.err = mean(yhat.bag != all2.te$poverty)
test.bag.err
```

```
## [1] 0.1184
```

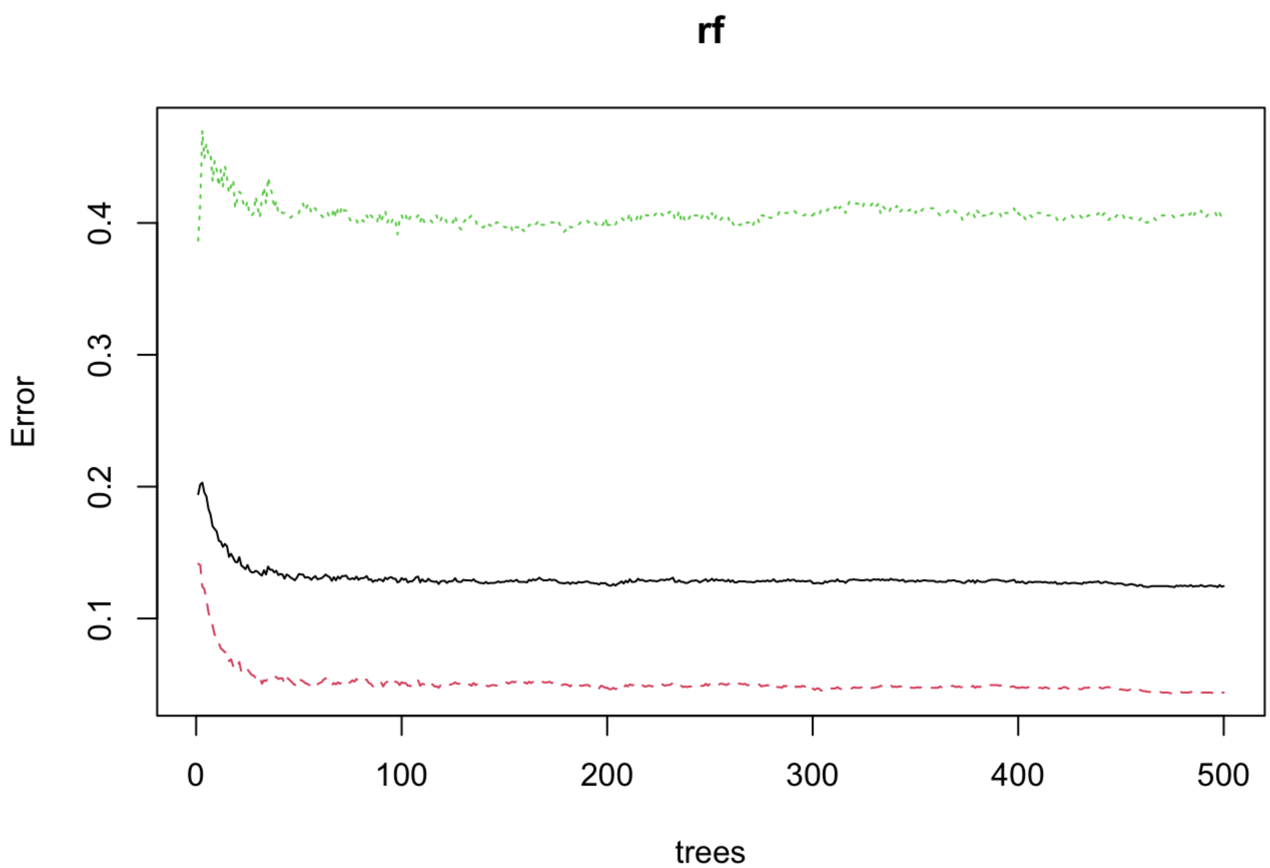
```
prob.bag = predict(bag, newdata = all2.te, type = "prob")
head(prob.bag)
```

```
##        0     1
## 3  0.244 0.756
## 6  0.414 0.586
## 12 0.156 0.844
## 22 0.792 0.208
## 26 0.778 0.222
## 42 0.904 0.096
```

```
set.seed(123)
rf = randomForest(poverty ~ ., data=all2.tr, importance=TRUE)
rf
```

```
##
## Call:
##  randomForest(formula = poverty ~ ., data = all2.tr, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 12.45%
## Confusion matrix:
##      0   1 class.error
## 0 1855  85  0.04381443
## 1  226 331  0.40574506
```

```
plot(rf)
```



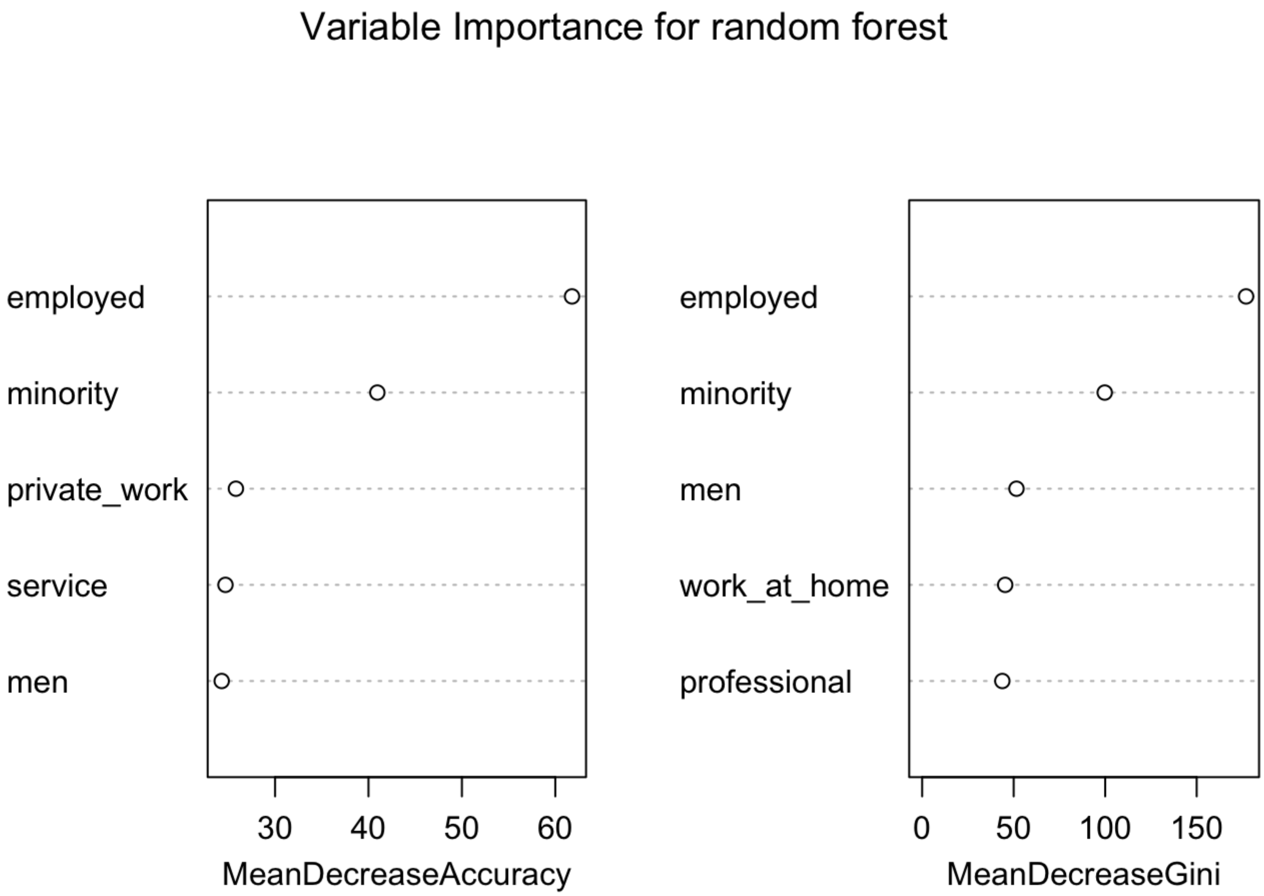

```
yhat.rf = predict(rf, newdata = all2.te)
test.rf.err = mean(yhat.rf != all2.te$poverty)
test.rf.err
```

```
## [1] 0.1168
```

```
importance(rf)
```

```
##                                                  0          1
## state                                   15.0249598 13.4079223
## county                                   0.7059549 -1.0456816
## total_pop                               16.7924449 -0.7053462
## men                                     21.4252609 11.0816693
## professional                            17.2728397  7.2129234
## service                                 21.2514366 11.8328502
## office                                  11.0185630  2.4868037
## production                              13.9434647  5.7584841
## drive                                   15.9146340  2.2908932
## carpool                                 13.5777104 -3.0298145
## work_at_home                            19.6172145 12.8185705
## employed                                31.3173778 65.6921050
## private_work                            21.0668665 13.7606735
## self_employed                           17.2043055  4.4987357
## family_work                              7.6304357  4.0683724
## minority                                26.8857130 41.2285567
## less_than_a_high_school_diploma_2015_19   17.2915230  6.5015545
## high_school_diploma_only_2015_19        15.5330213  2.3464036
## some_college_or_associates_degree_2015_19 14.9873739  7.0414947
## bachelors_degree_or_higher_2015_19      15.6074105  9.9993460
## county_population                       12.8174479  3.4821595
##                                         MeanDecreaseAccuracy MeanDecreaseGini
## state                                            18.84867266         31.58579
## county                                           -0.05672811         23.38964
## total_pop                                        17.45338734         23.99961
## men                                              24.28164366         51.53664
## professional                                     20.05157045         43.79899
## service                                          24.68558429         40.43870
## office                                           11.63071619         23.16979
## production                                       15.92246926         29.15493
## drive                                            16.24507893         24.65811
## carpool                                          10.16430342         22.32672
## work_at_home                                     23.26769966         45.35844
## employed                                         61.79132045        176.99471
## private_work                                     25.80503188         37.66197
## self_employed                                    18.22813076         30.32063
## family_work                                       9.02779418         14.43728
## minority                                         40.94750367         99.76200
## less_than_a_high_school_diploma_2015_19          18.93792121         34.87587
## high_school_diploma_only_2015_19                 16.14000004         23.48218
## some_college_or_associates_degree_2015_19        16.82982334         28.89923
## bachelors_degree_or_higher_2015_19               18.06172928         32.60310
## county_population                                14.71982028         24.52926
```

```
varImpPlot(rf, sort=T,
           main="Variable Importance for random forest", n.var=5)
```



The test set error rate is 0.11; this indicates that random forests did provide a slight improvement over bagging (test error 0.12) in this case. The variable importance results indicate that across all of the trees considered in the random forest, employed and minority are by far the two most important variables in terms of Model Accuracy and Gini index.

Method 2: KNN

```
set.seed(123)
idx.tr <- sample.int(n, 0.8*n)
all.tr <- all[idx.tr, ]

all.tr = all.tr %>%
  mutate(Poverty=as.factor(ifelse(Poverty>20,"1","0")))

df <- all.tr %>% select(-c(County,State))
idx.tr <- sample.int(n, 0.8*n)
train <- df[idx.tr, ]
test <- df[-idx.tr, ]

# train <- train %>%
#   mutate(Poverty=as.factor(ifelse(Poverty>20,"1","0")))
# test <- test %>%
#   mutate(Poverty=as.factor(ifelse(Poverty>20,"1","0")))

train <- na.omit(train)
test <- na.omit(test)

YTrain = train$Poverty
XTrain = train %>% select(-Poverty) %>% scale(center = TRUE, scale = TRUE)

YTest = test$Poverty
XTest = test %>% select(-Poverty) %>% scale(center = TRUE, scale = TRUE)

# train classifier and make predictions on training data
pred.YTtrain = knn(train=XTrain, test=XTrain, cl=YTrain)
# confusion matrix
conf.train = table(predicted=pred.YTtrain, true=YTrain)

# trainning error rate
1 - sum(diag(conf.train)/sum(conf.train))
```

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

```
pred.YTest = knn(train=XTrain, test=XTest, cl=YTrain)
# Get confusion matrix
conf.test = table(predicted=pred.YTest, true=YTest)
# Test error rate
1 - sum(diag(conf.test)/sum(conf.test))
```

```
## [1] 0.1988304
```

The test error rate is slightly higher than the training error rate, which is expected. The test error rate obtained by 2-NN classifier is quite ideal as 19.8% of the test observations are incorrectly predicted. If we compare both methods to LASSO, logistic and decision tree, latter models provided better or lower error rates. However, bagging and random forest have a fairly similar output to logistic regression. Both giving low train and test error rates.

20. (9 pts) Tackle at least one more interesting question. Creative and thoughtful analysis will be rewarded! Some possibilities for further exploration are:

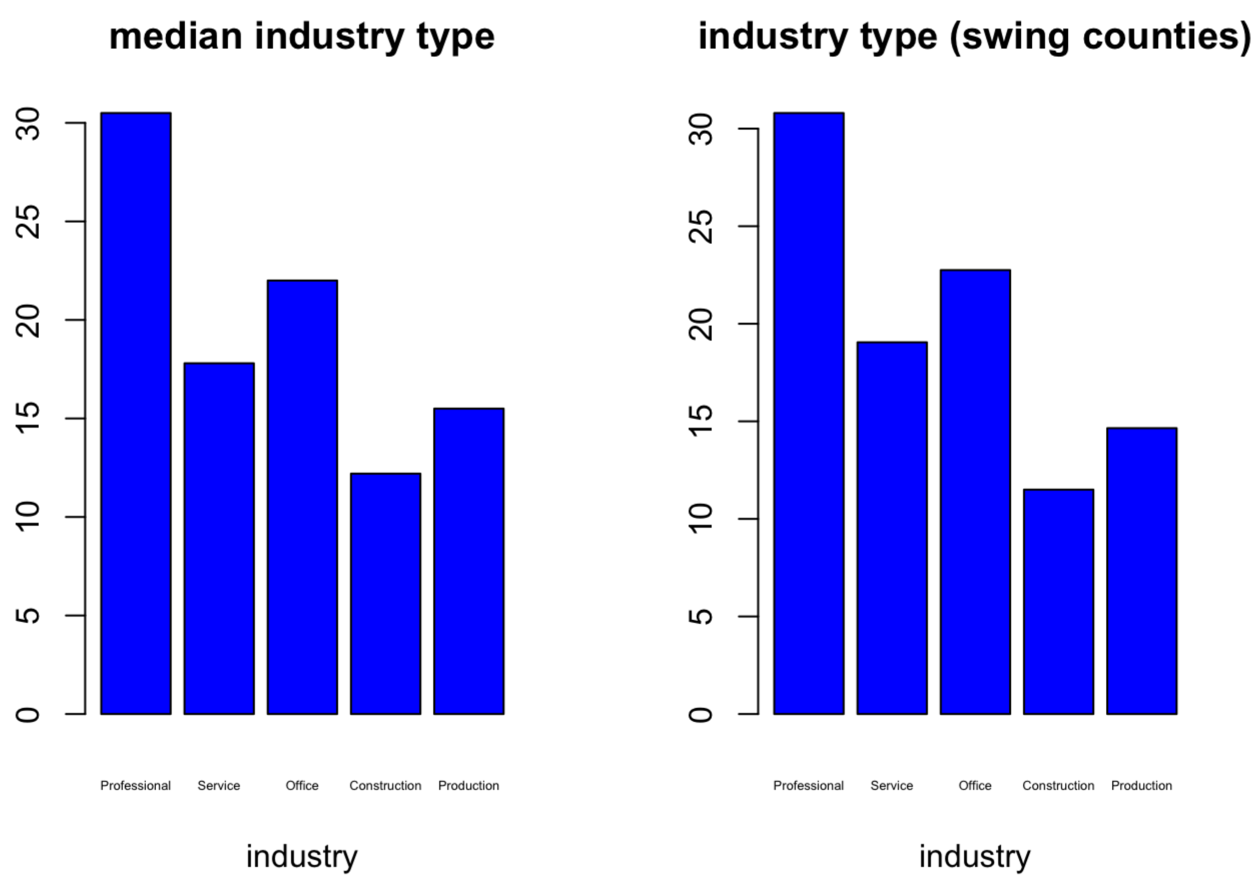
Swing counties are battleground counties that can make or break an election win. They are called swing counties because they seesaw back and forth between voting for Democratic and Republican parties. While the question of what makes them so difficult to predict has since been removed from this project in an updated version, we feel that it still poses an interesting question. That in mind, we will modify the question to instead encompass Donald Trump's 2016 election win over Hillary Clinton, as 1) it was a surprise almost no one saw coming, and 2) the provided datasets for this project only encompass the time up until 2019. We will perform exploratory analysis using a convenience sample from Ballotpedia's "List of Pivot Counties - the 206 counties that voted Obama-Obama-Trump," taking the first 20 unique county names and reconciling them with our provided datasets.

Exploratory Analysis of Swing Counties

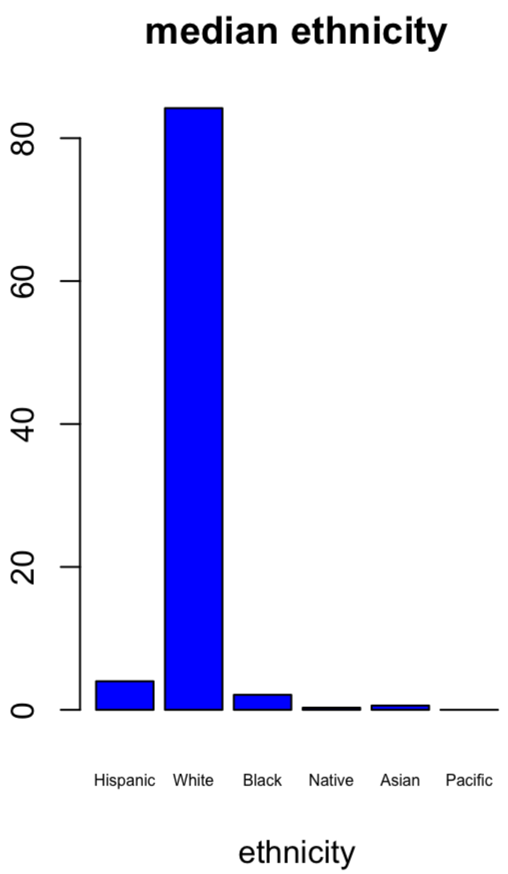
```
# take convenience sample of 20 counties from Ballotpedia's "Election results 2020: Pivo
t counties in the 2020 presidential election" (first twenty w/o duplicate names)

pivot.counties<-c('Woodruff County','Conejos County','Huerfano County','Las Animas Count
y','Pueblo County','Pinellas County','St. Lucie County','Dooly County','Peach County','T
wiggs County','Jo Daviess County','Whiteside County','LaPorte County','Porter County','A
llamakee County','Aroostook County','Kennebec County','Penobscot County','Eaton County',
'Gogebic County')

pivot_counties <- filter(census,County %in% pivot.counties) # extract pivot counties fro
m census data
head(pivot_counties)
```

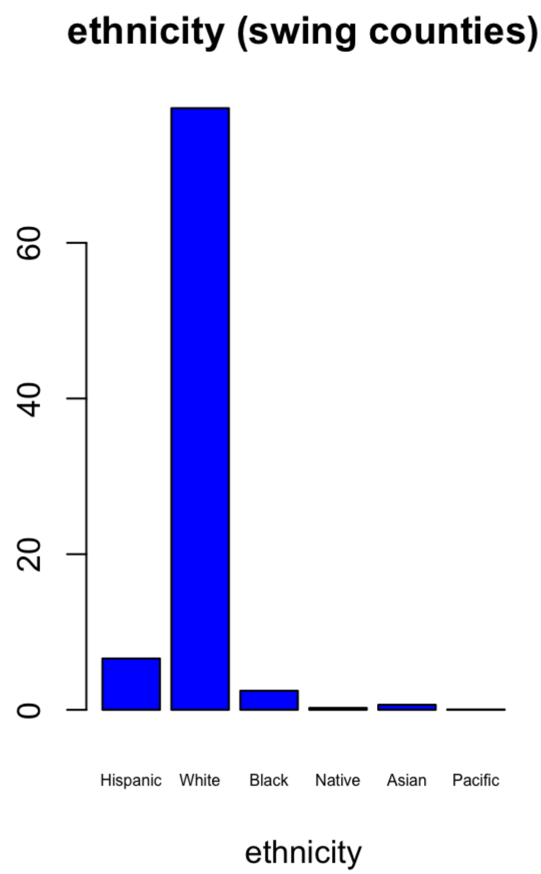
```
## # A tibble: 6 x 31
##   State County  TotalPop    Men  Women Hispanic White Black Native Asian Pacific
##   <chr> <chr>      <dbl>  <dbl>  <dbl>    <dbl> <dbl> <dbl>  <dbl> <dbl>   <dbl>
## 1 AR    Woodru…     6763   3193   3570      0.5  69.2  27.5    0     1.5     0.1
## 2 CO    Conejo…     8147   4084   4063     53.4  43.9   0.3    1.5   0.1     0.1
## 3 CO    Huerfa…     6498   3230   3268     34.7  64.2   0.2    0.3   0.1     0  
## 4 CO    Las An…    14151   7323   6828     42.4  52.7   0.8    2.1   0.8     0  
## 5 CO    Pueblo…   163368  80330  83038     42.7  52.4   1.6    0.5   0.7     0.1
## 6 FL    Pinell…   949842 456017 493825      9.2  74.7  10      0.2   3.3     0.1
## # … with 20 more variables: VotingAgeCitizen <dbl>, Poverty <dbl>,
## #   Professional <dbl>, Service <dbl>, Office <dbl>, Construction <dbl>,
## #   Production <dbl>, Drive <dbl>, Carpool <dbl>, Transit <dbl>, Walk <dbl>,
## #   OtherTransp <dbl>, WorkAtHome <dbl>, MeanCommute <dbl>, Employed <dbl>,
## #   PrivateWork <dbl>, PublicWork <dbl>, SelfEmployed <dbl>, FamilyWork <dbl>,
## #   Unemployment <dbl>
```

```
industry.median <- c(median(census$Professional),median(census$Service),median(census$Of
fice),median(census$Construction), median(census$Production))
industry.median2 <- c(median(pivot_counties$Professional),median(pivot_counties$Servic
e),median(pivot_counties$Office),median(pivot_counties$Construction), median(pivot_count
ies$Production))
industry.name <- c('Professional','Service','Office','Construction','Production')

op <- par(mfrow = c(1,2))
barplot(industry.median,names.arg=industry.name,main="median industry type",
        xlab="industry",cex.names=0.4,col=c("blue"))
barplot(industry.median2,names.arg=industry.name,main="industry type (swing counties)",
        xlab="industry",cex.names=0.4,col=c("blue"))
```



```
ethnicity.median <- c(median(census$Hispanic),median(census$White),median(census$Black),
median(census$Native), median(census$Asian),median(census$Pacific))
ethnicity.median2 <- c(median(pivot_counties$Hispanic),median(pivot_counties$White),medi
an(pivot_counties$Black),median(pivot_counties$Native), median(pivot_counties$Asian),med
ian(pivot_counties$Pacific))
industry.name <- c('Hispanic','White','Black','Native','Asian','Pacific')

op <- par(mfrow = c(1,2))
barplot(ethnicity.median,names.arg=industry.name,main="median ethnicity",
        xlab="ethnicity",cex.names=0.5,col=c("blue"))
barplot(ethnicity.median2,names.arg=industry.name,main="ethnicity (swing counties)",
        xlab="ethnicity",cex.names=0.5,col=c("blue"))
```






```
print("Citizen voting age - all vs. swing counties (bottom)")
```

```
## [1] "Citizen voting age - all vs. swing counties (bottom)"
```

```
median(census$VotingAgeCitizen)
```

```
## [1] 19479.5
```

```
median(pivot_counties$VotingAgeCitizen)
```

```
## [1] 31768
```

```
print("Poverty - all vs. swing counties (bottom)")
```

```
## [1] "Poverty - all vs. swing counties (bottom)"
```

```
median(census$Poverty)
```

```
## [1] 15.2
```

```
median(pivot_counties$Poverty)
```

```
## [1] 16.6
```

```
print("Unemployment - all vs. swing counties (bottom)")
```

```
## [1] "Unemployment - all vs. swing counties (bottom)"
```

```
median(census$Unemployment)
```

```
## [1] 6.1
```

```
median(pivot_counties$Unemployment)
```

```
## [1] 6.6
```

```
print("Family Work - all vs. swing counties (bottom)")
```

```
## [1] "Family Work - all vs. swing counties (bottom)"
```

```
median(census$FamilyWork)
```

```
## [1] 0.2
```

```
median(pivot_counties$FamilyWork)
```

```
## [1] 0.15
```

```
census2 = census
pivot.counties2 = pivot_counties
census2 = census2 %>% mutate(Men = Men/TotalPop)
pivot.counties2 = pivot.counties2 %>% mutate(Men=Men/TotalPop)

head(census2)
```

```
## # A tibble: 6 x 31
##   State County   TotalPop   Men  Women Hispanic White Black Native Asian Pacific
##   <chr> <chr>       <dbl> <dbl>  <dbl>    <dbl> <dbl> <dbl>  <dbl> <dbl>   <dbl>
## 1 AL    Autauga…    55036 0.489  28137      2.7  75.4  18.9    0.3   0.9       0
## 2 AL    Baldwin…   203360 0.489 103833      4.4  83.1   9.5    0.8   0.7       0
## 3 AL    Barbour…    26201 0.533  12225      4.2  45.7  47.8    0.2   0.6       0
## 4 AL    Bibb Co…    22580 0.543  10329      2.4  74.6  22      0.4   0         0
## 5 AL    Blount …    57667 0.494  29177      9    87.4   1.5    0.3   0.1       0
## 6 AL    Bullock…    10478 0.536   4862      0.3  21.6  75.6    1     0.7       0
## # … with 20 more variables: VotingAgeCitizen <dbl>, Poverty <dbl>,
## #   Professional <dbl>, Service <dbl>, Office <dbl>, Construction <dbl>,
## #   Production <dbl>, Drive <dbl>, Carpool <dbl>, Transit <dbl>, Walk <dbl>,
## #   OtherTransp <dbl>, WorkAtHome <dbl>, MeanCommute <dbl>, Employed <dbl>,
## #   PrivateWork <dbl>, PublicWork <dbl>, SelfEmployed <dbl>, FamilyWork <dbl>,
## #   Unemployment <dbl>
```

```
head(pivot.counties2)
```

```
## # A tibble: 6 x 31
##   State County   TotalPop   Men  Women Hispanic White Black Native Asian Pacific
##   <chr> <chr>       <dbl> <dbl>  <dbl>    <dbl> <dbl> <dbl>  <dbl> <dbl>   <dbl>
## 1 AR    Woodruf…     6763 0.472   3570      0.5  69.2  27.5    0      1.5     0.1
## 2 CO    Conejos…     8147 0.501   4063     53.4  43.9   0.3    1.5    0.1     0.1
## 3 CO    Huerfan…     6498 0.497   3268     34.7  64.2   0.2    0.3    0.1     0
## 4 CO    Las Ani…    14151 0.517   6828     42.4  52.7   0.8    2.1    0.8     0
## 5 CO    Pueblo …   163368 0.492  83038     42.7  52.4   1.6    0.5    0.7     0.1
## 6 FL    Pinella…   949842 0.480 493825      9.2  74.7  10      0.2    3.3     0.1
## # … with 20 more variables: VotingAgeCitizen <dbl>, Poverty <dbl>,
## #   Professional <dbl>, Service <dbl>, Office <dbl>, Construction <dbl>,
## #   Production <dbl>, Drive <dbl>, Carpool <dbl>, Transit <dbl>, Walk <dbl>,
## #   OtherTransp <dbl>, WorkAtHome <dbl>, MeanCommute <dbl>, Employed <dbl>,
## #   PrivateWork <dbl>, PublicWork <dbl>, SelfEmployed <dbl>, FamilyWork <dbl>,
## #   Unemployment <dbl>
```

```
print("% Men - all vs. swing counties (bottom)")
```

```
## [1] "% Men - all vs. swing counties (bottom)"
```

```
median(census2$Men);median(pivot.counties2$Men)
```

```
## [1] 0.4960161
```

```
## [1] 0.4940075
```

```
glm.fit = glm(Poverty ~ Hispanic+White+Black+Asian+Pacific+Native,data=census2)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Poverty ~ Hispanic + White + Black + Asian + Pacific +
##     Native, data = census2)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -18.0985   -3.5288   -0.5362    2.8801   26.1647
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.38173    6.14585   0.550  0.58219
## Hispanic     0.18286    0.06222   2.939  0.00332 **
## White        0.10004    0.06256   1.599  0.10994
## Black        0.33664    0.06247   5.389 7.60e-08 ***
## Asian       -0.35890    0.08292  -4.328 1.55e-05 ***
## Pacific      0.68190    0.16876   4.041 5.46e-05 ***
## Native       0.37522    0.06728   5.577 2.65e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 27.42201)
##
##     Null deviance: 135088  on 3141  degrees of freedom
## Residual deviance:  85968  on 3135  degrees of freedom
## AIC: 19330
##
## Number of Fisher Scoring iterations: 2
```

```
education2 <- clean_names(education)
pivot.counties2 <- filter(education2,county %in% pivot.counties) # extract pivot countie
s from census data
head(pivot.counties2)
```

```
## # A tibble: 6 x 7
##   state county    less_than_a_high_sc… high_school_diplom… some_college_or_asso…
##   <chr> <chr>                    <dbl>               <dbl>                 <dbl>
## 1 AR    Woodruff…                  853                2041                  1113
## 2 CO    Conejos …                  636                1794                  1747
## 3 CO    Huerfano…                  396                1546                  2171
## 4 CO    Las Anim…                 1279                3103                  4087
## 5 CO    Pueblo C…                11721               33019                 44046
## 6 FL    Pinellas…                64593              207322                234393
## # … with 2 more variables: bachelors_degree_or_higher_2015_19 <dbl>,
## #   county_population <dbl>
```

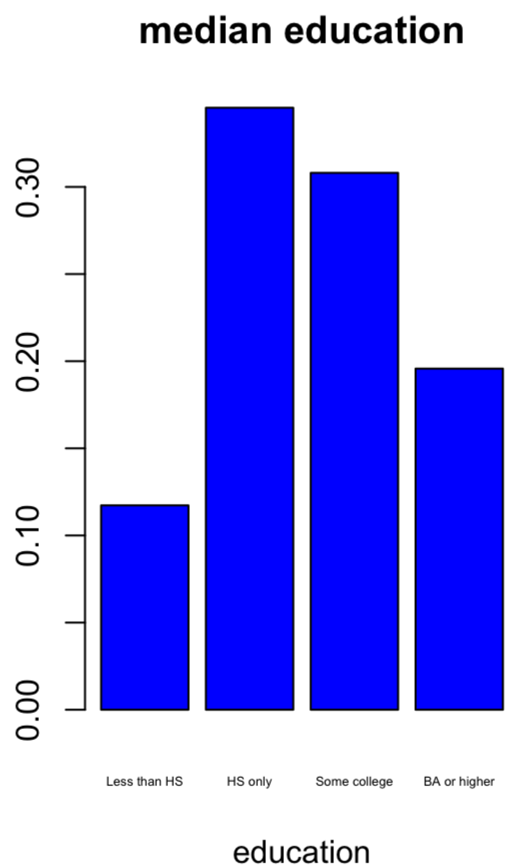
```
# mutate the education levels in both data sets to percentages for better comparability

pivot.counties2 = pivot.counties2 %>%
  mutate(less_than_a_high_school_diploma_2015_19=less_than_a_high_school_diploma_2015_1
9/county_population,high_school_diploma_only_2015_19=high_school_diploma_only_2015_19/co
unty_population,some_college_or_associates_degree_2015_19=some_college_or_associates_deg
ree_2015_19/county_population,bachelors_degree_or_higher_2015_19=bachelors_degree_or_hig
her_2015_19/county_population
)

education2 = education2 %>%
  mutate(less_than_a_high_school_diploma_2015_19=less_than_a_high_school_diploma_2015_1
9/county_population,high_school_diploma_only_2015_19=high_school_diploma_only_2015_19/co
unty_population,some_college_or_associates_degree_2015_19=some_college_or_associates_deg
ree_2015_19/county_population,bachelors_degree_or_higher_2015_19=bachelors_degree_or_hig
her_2015_19/county_population
)

head(pivot.counties2)
```

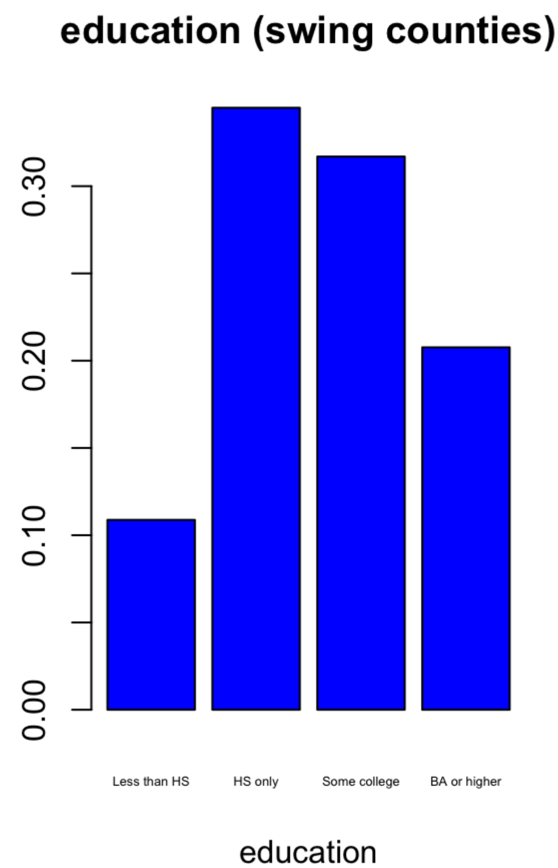
```
## # A tibble: 6 x 7
##   state county    less_than_a_high_sc… high_school_diplom… some_college_or_asso…
##   <chr> <chr>                    <dbl>               <dbl>                 <dbl>
## 1 AR    Woodruff…               0.183                0.438                 0.239
## 2 CO    Conejos …               0.120                0.339                 0.330
## 3 CO    Huerfano…               0.0759               0.296                 0.416
## 4 CO    Las Anim…               0.121                0.295                 0.388
## 5 CO    Pueblo C…               0.103                0.291                 0.388
## 6 FL    Pinellas…               0.0871               0.280                 0.316
## # … with 2 more variables: bachelors_degree_or_higher_2015_19 <dbl>,
## #   county_population <dbl>
```

```
head(education2)
```

```
## # A tibble: 6 x 7
##   state county   less_than_a_high_sch… high_school_diplom… some_college_or_asso…
##   <chr> <chr>                    <dbl>               <dbl>                 <dbl>
## 1 AL    Autauga…                0.115                0.336                 0.284
## 2 AL    Baldwin…                0.0919               0.277                 0.313
## 3 AL    Barbour…                0.268                0.356                 0.260
## 4 AL    Bibb Co…                0.209                0.449                 0.238
## 5 AL    Blount …                0.195                0.334                 0.340
## 6 AL    Bullock…                0.253                0.403                 0.223
## # … with 2 more variables: bachelors_degree_or_higher_2015_19 <dbl>,
## #   county_population <dbl>
```

```
counties.median <- c(median(education2$less_than_a_high_school_diploma_2015_19),median(e
ducation2$high_school_diploma_only_2015_19),median(education2$some_college_or_associates
_degree_2015_19),median(education2$bachelors_degree_or_higher_2015_19))

pivot.counties.median <- c(median(pivot.counties2$less_than_a_high_school_diploma_2015_1
9),median(pivot.counties2$high_school_diploma_only_2015_19), median(pivot.counties2$some
_college_or_associates_degree_2015_19),median(pivot.counties2$bachelors_degree_or_higher
_2015_19))

education.level <- c('Less than HS','HS only','Some college','BA or higher')

op <- par(mfrow = c(1,2))
barplot(counties.median,names.arg=education.level,main="median education",
        xlab="education",cex.names=0.4,col=c("blue"))
barplot(pivot.counties.median,names.arg=education.level,main="education (swing countie
s)",
        xlab="education",cex.names=0.4,col=c("blue"))
```





Exploratory analysis of the census and education data does not necessarily reveal a compelling explanation for what might make swing counties so difficult to predict. Comparing and contrasting the medians of key features reveals mostly equivalent results between counties as a whole and swing counties. However, we note that swing counties typically score higher in 1) unemployment, 2) poverty, 3) citizen voting age, and 4) Hispanic ethnicity. It is generally accepted that older demographics of voters tend to hold more conservative values because they have reached a more financially stable point in life. In this vein, the higher rates of poverty and unemployment in swing counties could be offsetting these more conservative values, thus making it difficult to predict the voting outcome of these counties. Furthermore, the Hispanic ethnicity in particular is noted to fall fairly close to the middle in the glm model on poverty which could offset poverty differences for the other ethnicity. In essence, it is likely a myriad of variables working in tandem that make the swing counties unpredictable.

21. (9 pts) (Open ended) Interpret and discuss any overall insights gained in this analysis and possible explanations. Use any tools at your disposal to make your case: visualize errors on the map, discuss what does/doesn't seems reasonable based on your understanding of these methods, propose possible directions (collecting additional data, domain knowledge, etc).

There are various studies that depict the hardships of achieving upwards mobility. Like our exploratory analysis in Question 20, these studies often seek to explore the effects of variables such as race, gender, location, and education level. For instance, it is common knowledge that there is a gender-wage disparity and that minority ethnic groups tend to not receive the same quality of education. The difficulty in assessing these effects, however, is predicated upon many of them overlapping. For instance, a person who lives in a so-called “poor area” is often simultaneously exposed to lower-quality education, which, in turn, affects their self-esteem and potential for future success. This feedback loop is further compounded by external influences (eg. friends or neighbors who might rope other individuals into dangerous activities), which make it harder for that individual to escape the poverty cycle. Taken together, this negative feedback loop points to a systematic imbalance in today's society wherein some people are set up to fail or will never have the same opportunities that others are afforded.

Similarly, this is the precedent for which Democratic and Republican lines are often drawn. Democratic core values are predicated upon ideas such as social equality, equal opportunity, and minority rights. By contrast, Republican values tend to emphasize the free market, deregulation, and restrictions on immigration - the idea being that the metaphorical lifeboat can only hold so many people before it sinks. Both sides present a valid argument, leading to spit lines which can be incredibly close in battleground counties. That in mind, it is likely a myriad of variables working in tandem that make the swing counties unpredictable. Using models specifically on poverty like we have in this project is unlikely to capture the nuances beyond broad conjecture that account for unpredictability of swing counties. Additionally, the poverty variable was often set to a binary indicator for this project with the value of 20 given arbitrarily. Future analysis might benefit from learning methods that explore the range of poverty in full.

Methods Avoided: We avoided SVM for question 19 as support vector machine does not work well with large datasets.