Classification and Prediction of Poverty in the United States Using County Level Census and Education Data

Meredith Johnson

```
knitr::opts_chunk$set(message = FALSE, warning = FALSE)
```

```
#install.packages('ggplot2')
#install.packages('tibble')
#install.packages('tidyr')
#install.packages('readr')
#install.packages('purrr')
#install.packages('dplyr')
#install.packages('stringr')
#install.packages('forcats')
#install.packages('crayon')
#install.packages('reshape')
#install.packages('cluster')
#install.packages('rpart')
#install.packages('Matrix')
#install.packages('randomForest')
```

```
library(tidyverse)
library(crayon)
library(reshape)
library(ISLR)
library(tree)
library(maptree)
library(glmnet)
library("ROCR")
library(rpart)
library("FNN")
library(randomForest)
```

In this report, I will study and analyze the Untied States county-level census and education data. In particular, my target is to build and evaluate statistical machine learning models to understand some of the potential causes of poverty.

Data

Census Data

I start with the 2017 United States county-level census data, which is available at US Census Demographic Data. This dataset contains many demographic variables for each county in the U.S.

I load in and clean the **census** dataset by transforming the full state names to abbreviations (to match the subsequent **education** dataset). Specifically, R contains default global variables **state.name** and **state.abb** that store the full names and the associated abbreviations of the 50 states. However, it does not contain District of Columbia (and the associated abbreviation DC). I add it back manually since the **census** dataset contains information in DC. I further remove data from Puerto Rico to ease the visualization later on in the report.

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

The following are the first few rows of the **census** data.

```
head(census)
```

```
## # A tibble: 6 x 31
##   State County     Total~1   Men  Women Hispa~2 White Black Native Asian Pacific
##   <chr> <chr>        <dbl> <dbl>  <dbl>   <dbl> <dbl> <dbl>  <dbl> <dbl>   <dbl>
## 1 AL    Autauga C~   55036 26899  28137     2.7  75.4  18.9    0.3   0.9       0
## 2 AL    Baldwin C~  203360 99527 103833     4.4  83.1   9.5    0.8   0.7       0
## 3 AL    Barbour C~   26201 13976  12225     4.2  45.7  47.8    0.2   0.6       0
## 4 AL    Bibb Coun~   22580 12251  10329     2.4  74.6  22      0.4   0         0
## 5 AL    Blount Co~   57667 28490  29177     9    87.4   1.5    0.3   0.1       0
## 6 AL    Bullock C~   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>, and abbreviated variable names 1: TotalPop, 2: Hispanic
```

Education Data

I also include the education dataset, available at Economic Research Service at USDA. The dataset contains county-level educational attainment for adults age 25 and older in 1970-2019. I specifically use educational attainment information for the time period of 2015-2019.

To clean the data, I remove uninformative columns (as in **FIPS Code**, **2003 Rural-urban Continuum Code**, **2003 Urban Influence Code**, **2013 Rural-urban Continuum Code**, and **2013 Urban Influence Code**). To be consistent with census data, I exclude data from Puerto Rico and rename **Area Name** to **County** in order to match that in the **census** dataset.

```
education <- read_csv("./education.csv", show_col_types = FALSE) %>%
  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')
```

Preliminary Data Analysis

```
print(paste('The dimensions of the $\textbf{census}$ dataset are', nrow(census), 'rows
↪  and', ncol(census), 'columns.'))
```

[1] "The dimensions of the **census** dataset are 3142 rows and 31 columns."

```
if (sum(is.na(census)) == 0) {
  print('There are no missing values in the $\textbf{census}$ dataset.')
} else {
  print(paste('There are', sum(is.na(census)), 'missing values in the $\textbf{census}$
   ↪  dataset.'))
}
```

[1] "There are no missing values in the **census** dataset."

```
if (length(unique(census$State)) == 51) {
  print('There are 51 distinct values contained in the $\textbf{State}$ variable of the
   ↪  $\textbf{census}$ dataset; Thus the data contains all states and a federal
   ↪  district.')
}
```

[1] "There are 51 distinct values contained in the **State** variable of the **census** dataset; Thus the data contains all states and a federal district."

```
print(paste('The dimensions of the $\textbf{education}$ dataset are', nrow(education),
↪  'rows and', ncol(education), 'columns.'))
```

[1] "The dimensions of the **education** dataset are 3143 rows and 42 columns."

```
rows_na <- education[rowSums(is.na(education)) > 0, ]
dis_counties <- length(unique(rows_na$County))
print(paste(dis_counties, 'distinct counties contain missing values in the
↪  $\textbf{education}$ dataset.'))
```

[1] "18 distinct counties contain missing values in the **education** dataset."

```
print(paste('There are', length(unique(education$County)), 'distinct values in the
↪  $\textbf{County}$ column of the $\textbf{education}$ dataset.'))
```

[1] "There are 1877 distinct values in the **County** column of the **education** dataset."

```
if (length(unique(census$County)) == length(unique(education$County))) {
  print('The values of total number of disinct county are the same in the
   ↪  $\textbf{education}$ dataset and in the $\textbf{census}$ dataset.')
} else {
  print('The values of total number of disinct county are not the same in the
   ↪  $\textbf{education}$ dataset and in the $\textbf{census}$ dataset.')
}
```

[1] "The values of total number of disinct county are the same in the **education** dataset and in the **census** dataset."

Data Wrangling

Here, I remove all NA values in education.

```
education = drop_na(education)
nrow(education)
```

```
## [1] 3125
```

In **education**, in addition to **State** and **County**, I 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'**. I mutate the **education** dataset by selecting these 6 features only, and create a new feature which is the **total population** of that county.

```
education <- education %>%
  select(State, County,
         `Less than a high school diploma, 2015-19`,
         `High school diploma only, 2015-19`,
         `Some college or associate's degree, 2015-19`,
         `Bachelor's degree or higher, 2015-19`) %>%
  mutate(Total_Population =`Less than a high school diploma, 2015-19`
         + `High school diploma only, 2015-19`
         + `Some college or associate's degree, 2015-19`
         + `Bachelor's degree or higher, 2015-19`)
head(education)
```

```
## # A tibble: 6 x 7
##   State County         Less than a high school~1 High ~2 Some ~3 Bache~4 Total~5
##   <chr> <chr>                              <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 AL    Autauga County                      4291   12551   10596    9929   37367
## 2 AL    Baldwin County                     13893   41797   47274   48148  151112
## 3 AL    Barbour County                      4812    6396    4676    2080   17964
## 4 AL    Bibb County                         3386    7256    3848    1678   16168
## 5 AL    Blount County                       7763   13299   13519    5210   39791
## 6 AL    Bullock County                      1798    2860    1587     856    7101
## # ... with abbreviated variable names
## #   1: `Less than a high school diploma, 2015-19`,
## #   2: `High school diploma only, 2015-19`,
## #   3: `Some college or associate's degree, 2015-19`,
## #   4: `Bachelor's degree or higher, 2015-19`, 5: Total_Population
```

I construct aggregated data sets from **education** data by creating a state-level summary into a dataset named **education.state**.

```
education.state <- education %>%
  group_by(State) %>%
  summarise(across(`Less than a high school diploma, 2015-19`:`Bachelor's degree or
  ↪  higher, 2015-19`, ~sum(.x)))

head(education.state)
```

```
## # A tibble: 6 x 5
##   State `Less than a high school diploma, 2015-19` High school~1 Some ~2 Bache~3
##   <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
## # ... with abbreviated variable names 1: `High school diploma only, 2015-19`,
## #   2: `Some college or associate's degree, 2015-19`,
## #   3: `Bachelor's degree or higher, 2015-19`
```

I create a data set named **state.level** on the basis of **education.state**, where I create a new feature which is the name of the education degree level with the largest population in that state.

```
col_names = colnames(select(education.state, -State))
state.level <- education.state %>%
  mutate(`name of the education degree level with the largest population` =
           col_names[max.col(select(education.state, -State))])
head(state.level)
```

```
## # A tibble: 6 x 6
##   State Less than a high school diploma, 2015--1 High ~2 Some ~3 Bache~4 name ~5
##   <chr>                                          <dbl>   <dbl>   <dbl>   <dbl> <chr>
## 1 AK                                             32338  126881  162816  137666 Some c~
## 2 AL                                            458922 1022839  993344  845772 High s~
## 3 AR                                            270168  684659  593576  463236 High s~
## 4 AZ                                            604935 1124129 1594817 1392598 Some c~
## 5 CA                                           4418675 5423462 7648680 8980726 Bachel~
## 6 CO                                            314312  810659 1114680 1538936 Bachel~
## # ... with abbreviated variable names
## #   1: `Less than a high school diploma, 2015-19`,
## #   2: `High school diploma only, 2015-19`,
## #   3: `Some college or associate's degree, 2015-19`,
## #   4: `Bachelor's degree or higher, 2015-19`,
## #   5: `name of the education degree level with the largest population`
```

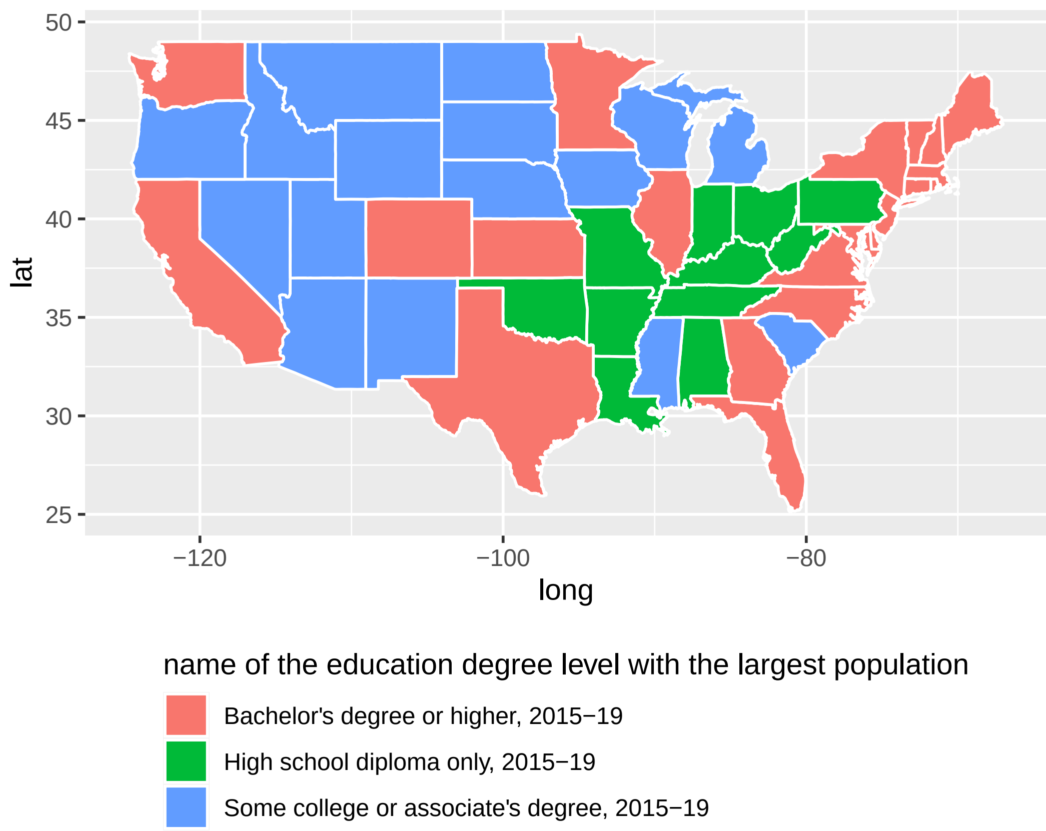
Visualization

Now I color a map of the United States (on the state level) by the education level with highest population for each state.

```
states <- map_data("state")
state.name.low = tolower(state.name)
states_modified <- states %>%
  mutate(region = state.abb[match(`region`, state.name.low)])
```

```
left_join_data <- left_join(states_modified, state.level, by = c('region' = 'State'))
```

```r
ggplot(data = left_join_data) +
  geom_polygon(aes(x = long, y = lat, fill = `name of the education degree level with the
  ↪  largest population`, group = group), color = "white") + coord_fixed(1.3) +
  ↪  theme(legend.position = "bottom", legend.direction="vertical")
```

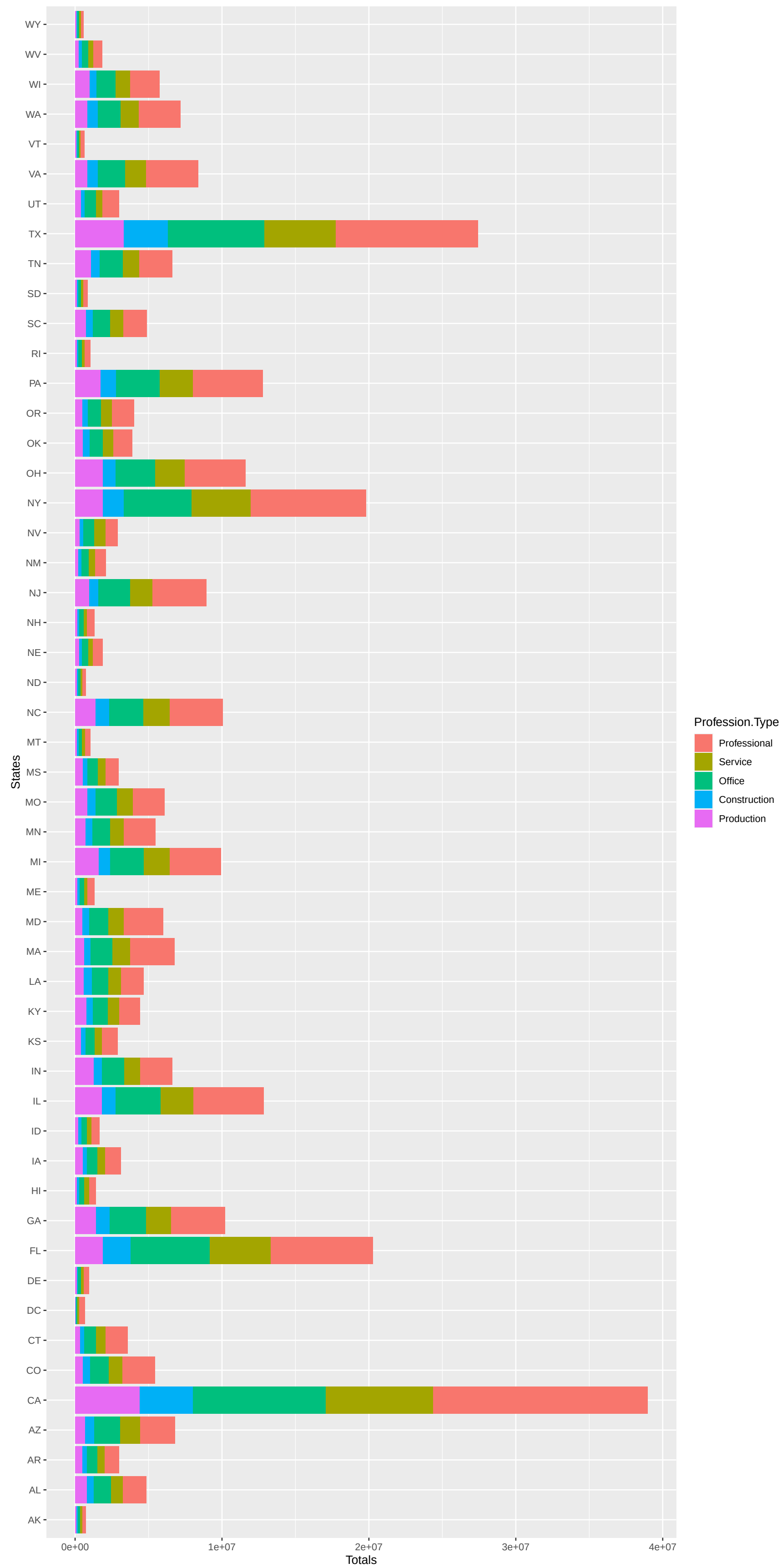


There were no states between 2015-2019 where **'Less than a high school diploma'** was the education attainment level with the largest population in that state.

Now, using the **census** data, I provide a bar graph where population of a state is represented by the length of a bar and percentage of that states' population with a profession belonging to the categories Professional, Service, Office, Construction, and Production is represented by color.

```r
profession.percent <- census %>% select(State, TotalPop, Professional, Service, Office,
↪  Construction, Production)

profession.population <- profession.percent %>% mutate(Professional = TotalPop
↪  *(Professional/100),
                                 Service = TotalPop * (Service/100),
                                 Office = TotalPop * (Office/100),
                                 Construction = TotalPop * (Construction/100),
                                 Production = TotalPop * (Production/100))


profession.bystate <- profession.population %>% select(-TotalPop) %>% group_by(State) %>%
↪  summarize(across(Professional:Production, ~ sum(.x, na.rm = TRUE)))
```

```r
profession <- profession.bystate %>% select(-State)

states <- rep(profession.bystate$State, each = 5)

profession.T = t(profession)
profession.totals <- melt(profession.T) %>% select(-X2)
colnames(profession.totals) <- c("Profession Type", "Totals")

profession.df <- data.frame(States = states, profession.totals)

ggplot(profession.df, aes(fill=Profession.Type, y=Totals, x=States)) +
    geom_bar(position="stack", stat="identity") +
  coord_flip()
```



WY
WV
WI
WA
VT
VA
UT
TX
TN
SD
SC
RI
PA
OR
OK
OH
NY
NV
NM
NJ
NH
NE
ND
NC
MT
MS
MO
MN
MI
ME
MD
MA
LA
KY
KS
IN
IL
ID
IA
HI
GA
FL
DE
DC
CT
CO
CA
AZ
AR
AL
AK
States
0e+00
1e+07
2e+07
3e+07
4e+07
Totals
Profession.Type
Professional
Service
Office
Construction
Production

This visualization of 2017 US **census** data conveys the magnitude of California's population in comparison to other states and reveals that the majority of every US state's population has a job belonging to the job category 'Professional' in 2017.

The **census** data contains county-level census information. I clean and aggregate the information by starting with the **census** data, filtering out any rows with missing values, converting **Men**, **Employed**, **VotingAgeCitizen** attributes to percentages, computing a **Minority** attribute by combining the **Hispanic**, **Black**, **Native**, **Asian**, **Pacific** attributes, removing the **Hispanic**, **Black**, **Native**, **Asian**, **Pacific** attributes after creating the **Minority** attribute, and removing the **Walk**, **PublicWork**, **Construction**, **Unemployment** attributes.

```
census.modified <- census %>%
  mutate(Men = (Men/TotalPop)*100,
         Employed = (Employed/TotalPop)*100,
         VotingAgeCitizen = (VotingAgeCitizen/TotalPop)*100,
         Minority = Hispanic+Black+Native+Asian+Pacific) %>%
  select(-c(Hispanic, Black, Native, Asian,
            Pacific,Walk, PublicWork, Construction, Unemployment))
head(census.modified)
```

```
## # A tibble: 6 x 23
##   State County Total~1   Men  Women White Votin~2 Poverty Profe~3 Service Office
##   <chr> <chr>    <dbl> <dbl>  <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  <dbl>
## 1 AL    Autau~   55036  48.9  28137  75.4    74.5    13.7    35.3    18     23.2
## 2 AL    Baldw~  203360  48.9 103833  83.1    76.4    11.8    35.7    18.2   25.6
## 3 AL    Barbo~   26201  53.3  12225  45.7    77.4    27.2    25      16.8   22.6
## 4 AL    Bibb ~   22580  54.3  10329  74.6    78.2    15.2    24.4    17.6   19.7
## 5 AL    Bloun~   57667  49.4  29177  87.4    73.7    15.6    28.5    12.9   23.3
## 6 AL    Bullo~   10478  53.6   4862  21.6    78.4    28.5    19.7    17.1   18.6
## # ... with 12 more variables: Production <dbl>, Drive <dbl>, Carpool <dbl>,
## #   Transit <dbl>, OtherTransp <dbl>, WorkAtHome <dbl>, MeanCommute <dbl>,
## #   Employed <dbl>, PrivateWork <dbl>, SelfEmployed <dbl>, FamilyWork <dbl>,
## #   Minority <dbl>, and abbreviated variable names 1: TotalPop,
## #   2: VotingAgeCitizen, 3: Professional
```

I find several columns to be perfectly collinear, in which case one column should be deleted.

```
tmp <- cor(select(census.modified,-c(State, County)))
diag(tmp) <- 0
which(tmp > 0.99, TRUE)
```

```
##          row col
## Women      3   1
## TotalPop   1   3
```

```
which(tmp < -0.99, TRUE)
```

```
##          row col
## Minority  21   4
## White      4  21
```

From the above result it is evident that **Women** and **TotalPop** are highly correlated while **Minority** and **White** are highly correlated; Therefore I choose to remove the columns **White** and **Women**.

```
census.clean <- census.modified %>%
  select(-c(White, Women))
```

The following are the first five rows of the **census.clean** data.

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

```
## # A tibble: 5 x 21
##   State County      Total~1   Men Votin~2 Poverty Profe~3 Service Office Produ~4
##   <chr> <chr>         <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  <dbl>   <dbl>
## 1 AL    Autauga Co~   55036  48.9    74.5    13.7    35.3    18     23.2    15.4
## 2 AL    Baldwin Co~  203360  48.9    76.4    11.8    35.7    18.2   25.6    10.8
## 3 AL    Barbour Co~   26201  53.3    77.4    27.2    25      16.8   22.6    24.1
## 4 AL    Bibb County   22580  54.3    78.2    15.2    24.4    17.6   19.7    22.4
## 5 AL    Blount Cou~   57667  49.4    73.7    15.6    28.5    12.9   23.3    19.5
## # ... with 11 more variables: Drive <dbl>, Carpool <dbl>, Transit <dbl>,
## #   OtherTransp <dbl>, WorkAtHome <dbl>, MeanCommute <dbl>, Employed <dbl>,
## #   PrivateWork <dbl>, SelfEmployed <dbl>, FamilyWork <dbl>, Minority <dbl>,
## #   and abbreviated variable names 1: TotalPop, 2: VotingAgeCitizen,
## #   3: Professional, 4: Production
```

Dimensionality reduction

I run PCA for the cleaned county level **census** data (with **State** and **County** excluded).

```
pr.out = prcomp(select(census.clean, -c(State, County)), scale = TRUE)
```

I save the first two principle components PC1 and PC2 into a two-column data frame and call it **pc.county**.

```
pc.county <- pr.out$x[, c('PC1','PC2')]
head(pc.county)
```

```
##             PC1        PC2
## [1,] -0.8024539 -0.8526622
## [2,] -0.2135456 -1.7982690
## [3,] -2.4403521  2.0806041
## [4,] -1.8997765  0.8098669
## [5,] -2.4614366 -1.4192899
## [6,] -2.8963593  2.6341947
```

I chose to center and scale the features before running PCA because features need to be centered before PCA is performed. Several groups of features had been recorded on different scale types; For instance: race and commute type were recorded as percentages of the population.

```
loadings = pr.out$rotation[,c("PC1")] %>% abs() %>% sort(decreasing = TRUE)
head(loadings, 3)
```

```
##   WorkAtHome SelfEmployed        Drive
##    0.4267336    0.3605124    0.3578110
```

WorkAtHome, **SelfEmployed**, and **Drive** are the three features with the largest absolute values of the first principal component. This is an indication that **WorkAtHome**, **SelfEmployed**, and **Drive** are the three features that explain the most variance within the population.

```
o <- order(abs(pr.out$rotation[,c("PC1")]), decreasing = TRUE)
pr.out$rotation[o,c("PC1")]
```

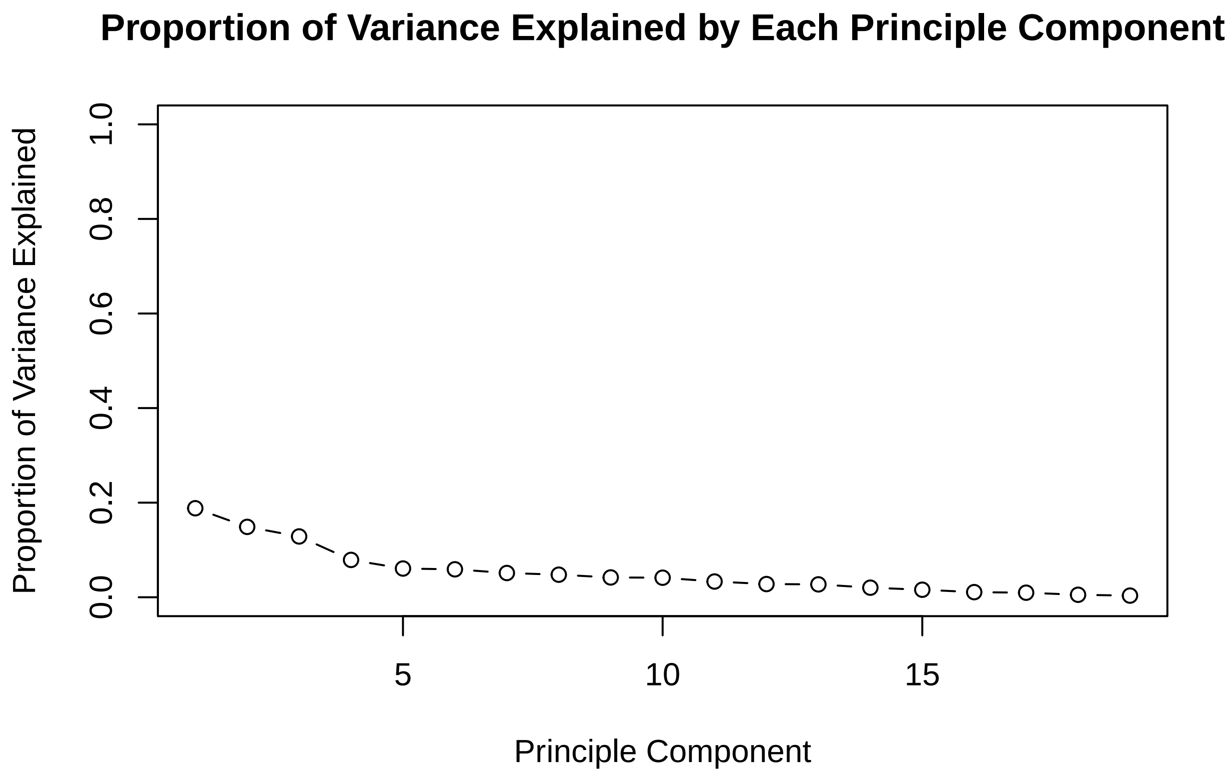
```
##       WorkAtHome     SelfEmployed            Drive     Professional
##       0.42673365       0.36051238      -0.35781105       0.34469432
##       Production      PrivateWork         Employed          Poverty
##      -0.29166926      -0.27012383       0.26003242      -0.24039363
##       FamilyWork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

In respect to the five features having the principle component loadings with the largest absolute values, **WorkAtHome**, **SelfEmployed**, and **Professional** possess a positive absolute value while **Drive** and **PrivateWork** possess principle component loadings with negative absolute values. Positive loadings indicate features and principal component that are positively correlated: an increase in one results in an increase in the other while the opposite is true for negative loadings. Features that are positively correlated with the first principle component are likely to be positively correlated with each other because the first principle component contains the most variance in the data. Negative correlation between features and the first principle component indicate contrast between those features and the first principle component. Therefore, features that have opposite signs are negatively correlated: an increase in one results in a decrease in the other.

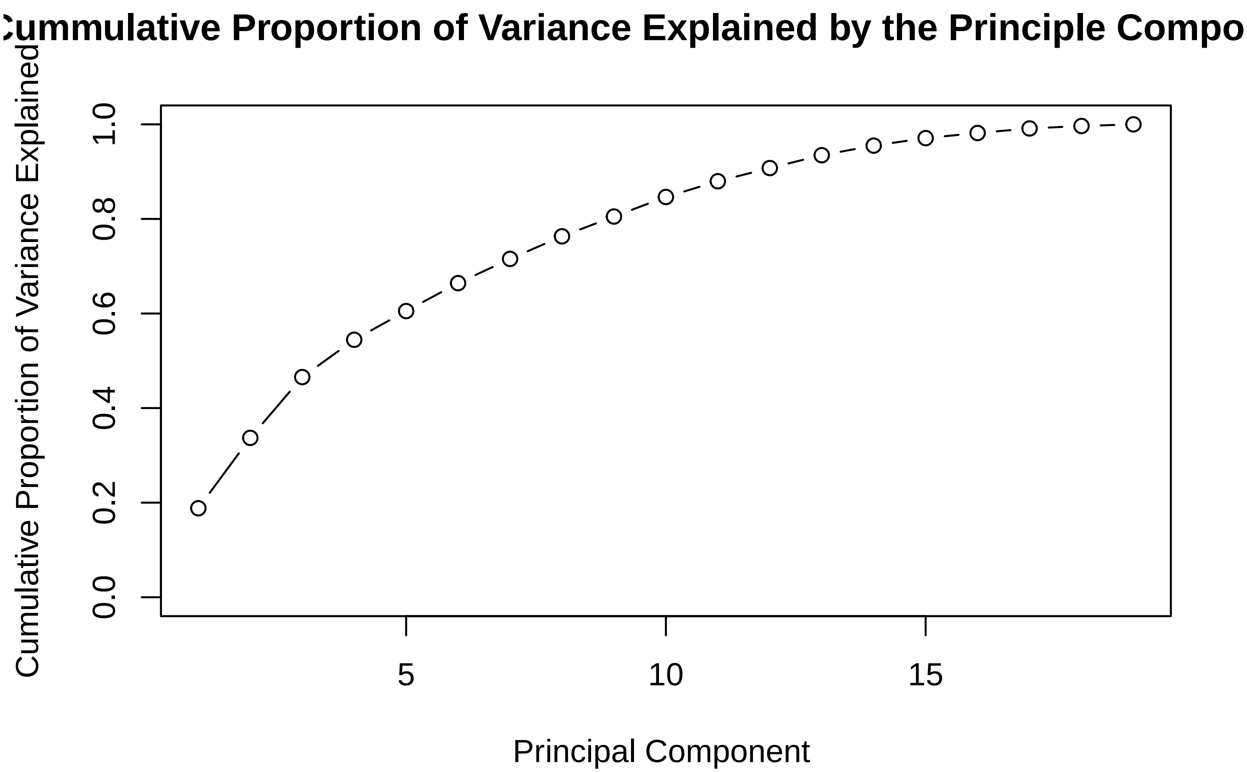
```
pr.var = pr.out$sdev^2
pve = pr.var/sum(pr.var)
min.pc <- min(which(cumsum(pve) > .9))
paste0('The minimum number of principle components needed to capture 90% of the variance
↪  for the analysis is ', min.pc, '.')
```

[1] "The minimum number of principle components needed to capture 90% of the variance for the analysis is 12."

```
plot(pve, xlab = "Principle Component", ylab = "Proportion of Variance Explained",
     ylim = c(0,1), type =  'b', main = "Proportion of Variance Explained by Each
      ↪  Principle Component")
```




```
plot(cumsum(pve), xlab="Principal Component ",
ylab=" Cumulative Proportion of Variance Explained ", ylim=c(0,1), type='b', main =
↪  "Cummulative Proportion of Variance Explained by the Principle Components")
```



Clustering

Here, I attempt two clustering approaches and compare the value of both approaches by analyzing which one puts Santa Barbara County in a more appropriate cluster.

Using **census.clean**, I perform hierarchical clustering with complete linkage.

```
census.clean.dist = dist(select(census.clean, -c(State, County)), method = "euclidean")
census.clean.hclust = hclust(census.clean.dist)
```

I cut the tree to partition the observations into 10 clusters.

```
clus = cutree(census.clean.hclust, 10)
table(clus)
```

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

Next, I re-run the hierarchical clustering algorithm using the first 2 principal components from **pc.county** as inputs instead of the original features.

```
pc.county.dist = dist(pc.county, method = "euclidean")
pc.county.hclust = hclust(pc.county.dist)
clus2 = cutree(pc.county.hclust, 10)
table(clus2)
```

```
## clus2
##    1    2    3    4    5    6    7    8    9   10
## 1734  272   42   79  772   19  109    1  100   14
```

Now, I compare the results of both approaches by investigating the clusters that contain Santa Barbara County.

```r
index = which(census.clean$County == "Santa Barbara County")
clus[index]
```

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

```r
clus2[index]
```

```
## [1] 5
```

```r
groups = which(clus == 1)
groups2 = which(clus2 == 5)
```

```r
head(census.clean[groups,], 20)
```

```
## # A tibble: 20 x 21
##    State County     Total~1   Men Votin~2 Poverty Profe~3 Service Office Produ~4
##    <chr> <chr>        <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  <dbl>   <dbl>
##  1 AL    Autauga C~   55036  48.9    74.5    13.7    35.3    18     23.2    15.4
##  2 AL    Baldwin C~  203360  48.9    76.4    11.8    35.7    18.2   25.6    10.8
##  3 AL    Barbour C~   26201  53.3    77.4    27.2    25      16.8   22.6    24.1
##  4 AL    Bibb Coun~   22580  54.3    78.2    15.2    24.4    17.6   19.7    22.4
##  5 AL    Blount Co~   57667  49.4    73.7    15.6    28.5    12.9   23.3    19.5
##  6 AL    Bullock C~   10478  53.6    78.4    28.5    19.7    17.1   18.6    30.6
##  7 AL    Butler Co~   20126  46.8    76.8    24.4    26.9    17.3   18.5    25.7
##  8 AL    Calhoun C~  115527  48.1    76.5    18.6    29      17.5   23.7    19.4
##  9 AL    Chambers ~   33895  48.1    77.5    18.8    24.3    13.5   23      27.6
## 10 AL    Cherokee ~   25855  49.7    79.8    16.1    28.8    14.8   18.1    26.5
## 11 AL    Chilton C~   43805  49.2    72.5    19.4    25.3    14.5   23.7    21
## 12 AL    Choctaw C~   13188  47.6    79.3    22.3    23.6    15.4   22      21.9
## 13 AL    Clarke Co~   24625  47.3    77.5    25.3    21.6    14.3   24.8    25.6
## 14 AL    Clay Coun~   13407  48.3    77.1    19.1    22.2    14.6   18.4    31.9
## 15 AL    Cleburne ~   14939  49.3    75.8    19.1    25.7    11.4   23.2    21.7
## 16 AL    Coffee Co~   51073  49.4    74.5    16.1    31.6    17.3   21.4    16.7
## 17 AL    Colbert C~   54435  48.0    77.5    16.8    27.1    15.4   26      20.3
## 18 AL    Conecuh C~   12649  47.8    77.6    26.4    15.9    19.7   24.3    27.4
## 19 AL    Coosa Cou~   10955  50.0    82.1    14.4    17.6    23.2   23.7    20.9
## 20 AL    Covington~   37519  48.3    77.7    17.6    29.2    14.3   22.2    18.4
## # ... with 11 more variables: Drive <dbl>, Carpool <dbl>, Transit <dbl>,
## #   OtherTransp <dbl>, WorkAtHome <dbl>, MeanCommute <dbl>, Employed <dbl>,
## #   PrivateWork <dbl>, SelfEmployed <dbl>, FamilyWork <dbl>, Minority <dbl>,
## #   and abbreviated variable names 1: TotalPop, 2: VotingAgeCitizen,
## #   3: Professional, 4: Production
```

```r
var(census.clean[groups,6])
```

```
##          Poverty
## Poverty 43.44052
```

```r
head(census.clean[groups2,], 20)
```

```
## # A tibble: 20 x 21
##    State County     Total~1   Men Votin~2 Poverty Profe~3 Service Office Produ~4
##    <chr> <chr>        <dbl> <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  <dbl>   <dbl>
##  1 AK    Aleutians~    5784  61.2    61.6     7.5    17.4    14.9   17      36.2
##  2 AK    Anchorage~  298225  51.1    71.7     8.1    40      17.4   24.3     9.5
##  3 AK    Fairbanks~  100031  54.0    73.7     7.7    37.6    15.6   23.2     9.6
##  4 AK    Ketchikan~   13745  51.4    74.5    10.6    29.7    17.2   27.5    13.7
##  5 AK    Matanuska~  101135  52.2    71.3     9.8    32.1    17.3   22      10.1
##  6 AK    Valdez-Co~    9439  51.9    75.3     7.4    27      17.2   21.8     9.8
##  7 AZ    Cochise C~  126516  50.7    72.8    18.1    34.4    24.3   24.1     7.2
##  8 AZ    Coconino ~  138639  49.2    75.8    21      36.5    22.7   22.5     9.9
##  9 AZ    Pima Coun~ 1007257  49.2    72.1    18.3    36.4    21.5   24.9     8.5
## 10 AZ    Yavapai C~  220972  48.9    79.8    14.7    31.9    22.7   24.5     9.8
## 11 AR    Newton Co~    7898  50.4    79.6    17.8    27.8    21.1   18.6    16.5
## 12 AR    Perry Cou~   10320  49.6    77.4    17.8    28.7    18.4   24.1    13.7
## 13 AR    Searcy Co~    7925  50.9    79.5    17.4    25      10.1   23.2    21.9
## 14 CA    Amador Co~   37306  53.6    82.2    10.6    32.7    23.2   24.1    10.1
## 15 CA    Butte Cou~  225207  49.5    76.4    20.5    35.9    22.2   22.7    10.1
## 16 CA    Calaveras~   45057  49.5    79.4    12.8    35.6    18.4   22.9    11.5
## 17 CA    Contra Co~ 1123678  48.8    66.2     9.8    43      18     23.1     7.9
## 18 CA    El Dorado~  185015  49.9    75.3     9.8    41.4    19     24.2     7
## 19 CA    Humboldt ~  135490  49.8    77.7    20.8    33.9    23.1   23.6     9.1
## 20 CA    Inyo Coun~   18195  50.4    74.9    10.2    31.5    23     22.4     8.7
## # ... with 11 more variables: Drive <dbl>, Carpool <dbl>, Transit <dbl>,
## #   OtherTransp <dbl>, WorkAtHome <dbl>, MeanCommute <dbl>, Employed <dbl>,
## #   PrivateWork <dbl>, SelfEmployed <dbl>, FamilyWork <dbl>, Minority <dbl>,
## #   and abbreviated variable names 1: TotalPop, 2: VotingAgeCitizen,
## #   3: Professional, 4: Production
```

```r
var(census.clean[groups2, 6])
```

```
##          Poverty
## Poverty 20.67398
```

The second approach seems to put Santa Barbara County in a more appropriate cluster. The first approach uses all of the information contained in the data and organizes the majority of the data points into one cluster; This creates more variance within the attributes of the data set which is demonstrated by the larger variance of the attribute **poverty** in the cluster containing Santa Barbara Country produced by the first approach. Large variance of attributes within clusters does not allow for a meaningful analysis. The second approach organizes Counties into more evenly distributed clusters.

Modeling

Here, I attempt to answer the question: *Can I use census information as well as the education information in a county to predict the level of poverty in that county?*

For simplicity, I transform **Poverty** into a binary categorical variable: high and low, and conduct its classification. The variable **Poverty** originally represents the percentage of the population that is below the poverty level.

In order to build classification models, I first need to combine the **education** and **census.clean** data and remove all NAs.

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

Here, I 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. I also remove features that I think are uninformative in classification tasks.

```
all <- all %>% mutate(Poverty =as.factor(ifelse(Poverty > 20, 1, 0))) %>% select(-State,
↪  -County, -Total_Population)
head(all)
```

```
## # A tibble: 6 x 23
##   TotalPop   Men VotingAg~1 Poverty Profe~2 Service Office Produ~3 Drive Carpool
##      <dbl> <dbl>      <dbl> <fct>     <dbl>   <dbl>  <dbl>   <dbl> <dbl>   <dbl>
## 1    55036  48.9       74.5 0          35.3    18     23.2    15.4  86        9.6
## 2   203360  48.9       76.4 0          35.7    18.2   25.6    10.8  84.7      7.6
## 3    26201  53.3       77.4 1          25      16.8   22.6    24.1  83.4     11.1
## 4    22580  54.3       78.2 0          24.4    17.6   19.7    22.4  86.4      9.5
## 5    57667  49.4       73.7 0          28.5    12.9   23.3    19.5  86.8     10.2
## 6    10478  53.6       78.4 1          19.7    17.1   18.6    30.6  73.1     15.7
## # ... with 13 more variables: Transit <dbl>, OtherTransp <dbl>,
## #   WorkAtHome <dbl>, MeanCommute <dbl>, Employed <dbl>, PrivateWork <dbl>,
## #   SelfEmployed <dbl>, FamilyWork <dbl>, Minority <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>, and abbreviated variable ...
```

I partition the dataset into 80% training and 20% test data.

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

I use the following code to define 10 cross-validation folds:

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

I use the following error rate function as well as the object **records** to record the classification performance of each method in the subsequent report.

```
calc_error_rate = function(predicted.value, true.value){
  return(mean(true.value!=predicted.value))
}
records = matrix(NA, nrow=3, ncol=2)
colnames(records) = c("train.error","test.error")
rownames(records) = c("tree","logistic","lasso")
```

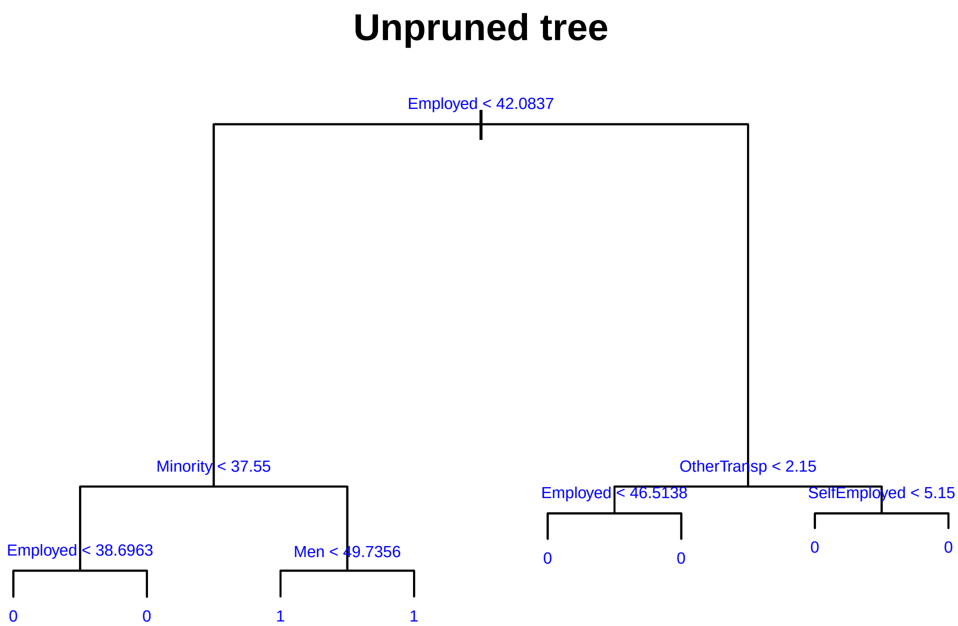
Classification

Here, I train a decision tree using cv.tree().

```
all.rename <- all %>% dplyr::rename(LessThanHighSchool =
                                      "Less than a high school diploma, 2015-19",
                                    HighSchool = "High school diploma only, 2015-19",
                                    SomeCollege = "Some college or associate's degree,
                                     ↪  2015-19",
                                    BachelorsOrHigher = "Bachelor's degree or higher,
                                     ↪  2015-19")

all.rename.tr <- all.rename[idx.tr, ]
all.rename.te <- all.rename[-idx.tr, ]

tree.all = tree(Poverty~., data = all.rename.tr)
summary(tree.all)
```

```
##
## Classification tree:
## tree(formula = Poverty ~ ., data = all.rename.tr)
## Variables actually used in tree construction:
## [1] "Employed"     "Minority"     "Men"          "OtherTransp"  "SelfEmployed"
## Number of terminal nodes:  8
## Residual mean deviance:  0.651 = 1620 / 2489
## Misclassification error rate: 0.1554 = 388 / 2497
```

```
plot(tree.all)
text(tree.all, pretty=0, col = "blue", cex = .5)
title("Unpruned tree")
```



I prune the tree to minimize misclassification error and use the folds from above for cross-validation.

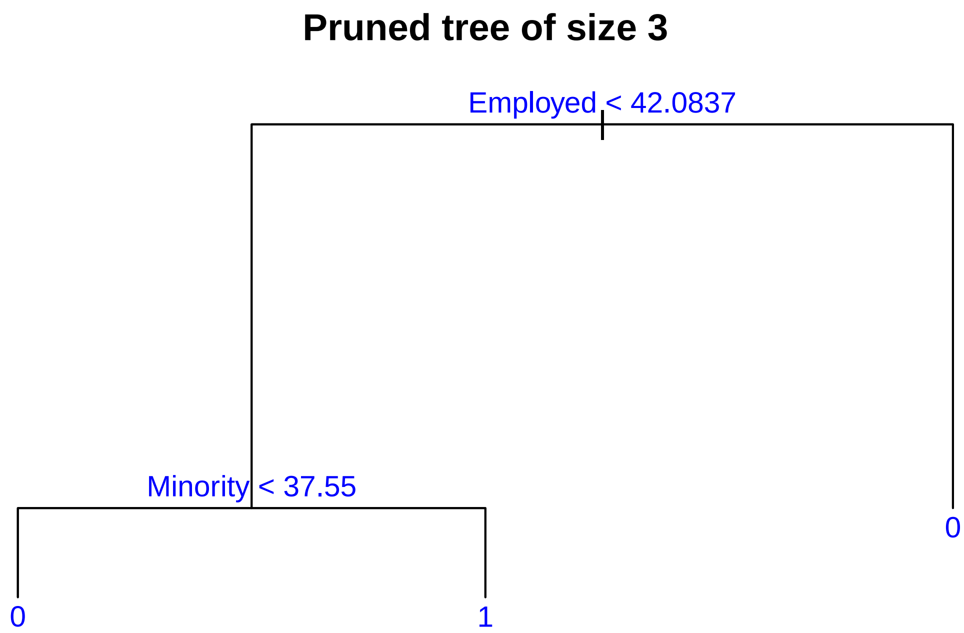
```
set.seed(1)

cv = cv.tree(tree.all, folds, FUN = prune.misclass, K = 10)

best_size = min(cv$size[cv$dev == min(cv$dev)])
print(paste("Smallest tree size that results in the minimum misclassification rate:",
↪  best_size))
```

[1] "Smallest tree size that results in the minimum misclassification rate: 3"

```
pt.cv = prune.misclass (tree.all, best=best_size)
```

I provide a visualization of the trees before and after pruning.

```
plot(pt.cv)
text(pt.cv, pretty=0, col = "blue", cex = .9)
title("Pruned tree of size 3")
```



I save training and test errors to the **records** object.

```
tree.prob.train = predict(pt.cv, type="class")
tree.prob.test = predict(pt.cv, newdata = all.rename.te, type="class")

tree.train.error = calc_error_rate(tree.prob.train, all.rename.tr$Poverty)
tree.test.error = calc_error_rate(tree.prob.test, all.rename.te$Poverty)
records["tree", ] <- c(tree.train.error, tree.test.error)
records
```

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

The pruning of the decision tree indicates that the most significant predictors of a state retaining a greater than 20% poverty rate are that state having a less than 42% employment rate and greater than 37.55% minority population.

Conculstions Drawn from the Decision Tree: This decision tree indicates that counties with larger minority populations as well as less employment are more likely to be in poverty; A population that is less employed has less income and insufficient income is indicative of poverty. Likewise, The Decision Tree indicates that there are systemic factors amongst counties with larger minority populations that contribute to those counties being in poverty.

Here, I run a logistic regression to predict Poverty in each county.

```r
glm.fit = glm(Poverty ~ ., data=all.rename.tr, family=binomial)
```

I save training and test errors to the **records** variable.

```r
log.prob.train = predict(glm.fit, type="response")
log.prob.test = predict(glm.fit, newdata = all.rename.te, type="response")

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

log.train.error = calc_error_rate(log.prob.train, all.rename.tr$Poverty)
log.test.error = calc_error_rate(log.prob.test, all.rename.te$Poverty)
records["logistic", ] <- c(log.train.error, log.test.error)
```

Here, I display the significant variables of poverty in each county.

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

```
##
## Call:
## glm(formula = Poverty ~ ., family = binomial, data = all.rename.tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3122  -0.4188  -0.1575  -0.0029   3.4222
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         2.751e+01  4.409e+00   6.239 4.40e-10 ***
## TotalPop            1.579e-04  1.992e-05   7.927 2.24e-15 ***
## Men                -3.468e-01  2.977e-02 -11.649  < 2e-16 ***
## VotingAgeCitizen    4.438e-02  1.975e-02   2.247 0.024650 *
## Professional        5.262e-02  2.539e-02   2.073 0.038187 *
## Service             9.230e-02  2.892e-02   3.192 0.001414 **
## Office              9.219e-03  3.070e-02   0.300 0.763975
## Production          8.075e-02  2.316e-02   3.487 0.000489 ***
## Drive              -5.084e-02  2.984e-02  -1.704 0.088423 .
## Carpool            -1.510e-03  3.736e-02  -0.040 0.967751
## Transit             9.689e-02  6.458e-02   1.500 0.133519
## OtherTransp        -1.171e-01  6.760e-02  -1.732 0.083352 .
## WorkAtHome         -1.293e-01  4.826e-02  -2.679 0.007389 **
## MeanCommute        -2.794e-02  1.638e-02  -1.706 0.087947 .
## Employed           -2.975e-01  1.977e-02 -15.048  < 2e-16 ***
## PrivateWork        -2.587e-02  1.672e-02  -1.548 0.121737
## SelfEmployed       -4.017e-02  3.195e-02  -1.257 0.208652
## FamilyWork         -1.311e-01  1.816e-01  -0.722 0.470373
## Minority            3.736e-02  4.838e-03   7.722 1.15e-14 ***
## LessThanHighSchool -1.926e-04  3.707e-05  -5.196 2.04e-07 ***
## HighSchool         -2.048e-04  3.035e-05  -6.747 1.51e-11 ***
## SomeCollege        -3.545e-04  4.581e-05  -7.738 1.01e-14 ***
## BachelorsOrHigher  -2.111e-04  3.203e-05  -6.589 4.43e-11 ***
```

```
## ---
## 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
```

TotalPop, **Men**, **Production**, **Employed**, **Minority**, **'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'** are the most significant variables. Among these variables, **Men**, **Employed**, and **Minority** were also present in the decision tree analysis. Among the most significant logistic regression variables, **Men**, **Employed**, and **Minority** where some of the most significant; therefore, I find the significant logistic regression variables to be fairly consistent with the significant decision tree analysis variables.

The variable **Men** has a coefficient of -0.3468. For every one unit change in **Men**, the log odds of **Poverty** being greater than 20 decreases by 0.3468, holding other variables fixed. This is an indication that as a county's population of men increases, that county becomes less likely to have more than 20% of its inhabitants under the poverty level. The variable **Employed** has a coefficient of -0.2975. For every one unit change in **Employed**, the log odds of **Poverty** being greater than 20 decreases by 0.2975, holding other variables fixed. This is an indication that as a county's employed population increases, that county becomes less likely to have more than 20% of its inhabitants under the poverty level. The variable **Minority** has a coefficient of 0.03736. For every one unit change in **Minority**, the log odds of **Poverty** being greater than 20 increases by 0.03736, holding other variables fixed. This is an indication that as a county's minority population increases, that county becomes more likely to have more than 20% of its inhabitants under the poverty level.

It is possible to get a warning **glm.fit**: fitted probabilities numerically 0 or 1 occurred. This is an indication that there is perfect separation (some linear combination of variables perfectly predicts the winner). This is usually a sign that there is overfitting. One way to control overfitting in logistic regression is through regularization.

I 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. I set **lambda = seq(1, 20) * 1e-5** in **cv.glmnet()** function to set pre-defined candidate values for the tuning parameter λ.

```
set.seed(123)
x <- model.matrix(Poverty~., all.rename)
y <- all$Poverty

x.train = x[idx.tr, ]
y.train = y[idx.tr]

# The rest as test data
x.test = x[-idx.tr, ]
y.test = y[-idx.tr]

set.seed(123)

cv.out.lasso = cv.glmnet(x.train, y.train, nfolds = 10, lambda = seq(1, 20) * 1e-5, alpha
↪  = 1, family = "binomial")
```

```
bestlam.lasso = cv.out.lasso$lambda.min
print(paste("Optimal value of tuning parameter lambda:", bestlam.lasso))
```

[1] "Optimal value of tuning parameter lambda: 1e-05" Here I display the non-zero coefficients in the LASSO regression for the optimal value of λ?

```
lasso.fit=glmnet(x.train,y.train,alpha=1,lambda=bestlam.lasso, family = "binomial")
lasso.coef=predict(lasso.fit,type="coefficients",s=bestlam.lasso)
lasso.coef
```

```
## 24 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)        27.7054632253
## (Intercept)          .
## TotalPop            0.0001466950
## Men                -0.3449296557
## VotingAgeCitizen    0.0427426018
## Professional        0.0532052506
## Service             0.0917340760
## Office              0.0089808679
## Production          0.0810747061
## Drive              -0.0525539034
## Carpool            -0.0030781228
## Transit             0.0867405540
## OtherTransp        -0.1160403631
## WorkAtHome         -0.1310196051
## MeanCommute        -0.0281229260
## Employed           -0.2958985709
## PrivateWork        -0.0265232086
## SelfEmployed       -0.0430289031
## FamilyWork         -0.1322435842
## Minority            0.0372817443
## LessThanHighSchool -0.0001758836
## HighSchool         -0.0001928097
## SomeCollege        -0.0003316877
## BachelorsOrHigher  -0.0001938901
```

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

```
##
## Call:
## glm(formula = Poverty ~ ., family = binomial, data = all.rename.tr)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3122  -0.4188  -0.1575  -0.0029   3.4222
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         2.751e+01  4.409e+00   6.239 4.40e-10 ***
## TotalPop            1.579e-04  1.992e-05   7.927 2.24e-15 ***
```

```
## Men                 -3.468e-01  2.977e-02 -11.649  < 2e-16 ***
## VotingAgeCitizen     4.438e-02  1.975e-02   2.247 0.024650 *
## Professional         5.262e-02  2.539e-02   2.073 0.038187 *
## Service              9.230e-02  2.892e-02   3.192 0.001414 **
## Office               9.219e-03  3.070e-02   0.300 0.763975
## Production           8.075e-02  2.316e-02   3.487 0.000489 ***
## Drive               -5.084e-02  2.984e-02  -1.704 0.088423 .
## Carpool             -1.510e-03  3.736e-02  -0.040 0.967751
## Transit              9.689e-02  6.458e-02   1.500 0.133519
## OtherTransp         -1.171e-01  6.760e-02  -1.732 0.083352 .
## WorkAtHome          -1.293e-01  4.826e-02  -2.679 0.007389 **
## MeanCommute         -2.794e-02  1.638e-02  -1.706 0.087947 .
## Employed            -2.975e-01  1.977e-02 -15.048  < 2e-16 ***
## PrivateWork         -2.587e-02  1.672e-02  -1.548 0.121737
## SelfEmployed        -4.017e-02  3.195e-02  -1.257 0.208652
## FamilyWork          -1.311e-01  1.816e-01  -0.722 0.470373
## Minority             3.736e-02  4.838e-03   7.722 1.15e-14 ***
## LessThanHighSchool -1.926e-04  3.707e-05  -5.196 2.04e-07 ***
## HighSchool          -2.048e-04  3.035e-05  -6.747 1.51e-11 ***
## SomeCollege         -3.545e-04  4.581e-05  -7.738 1.01e-14 ***
## BachelorsOrHigher  -2.111e-04  3.203e-05  -6.589 4.43e-11 ***
## ---
## 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
```

The coefficients for lasso and unpenalized logistic regression are very similar with some differences, and they have the same training error. Lasso and logistic regression share all the same significant variables. The similarities in coefficients may explain their same training errors.

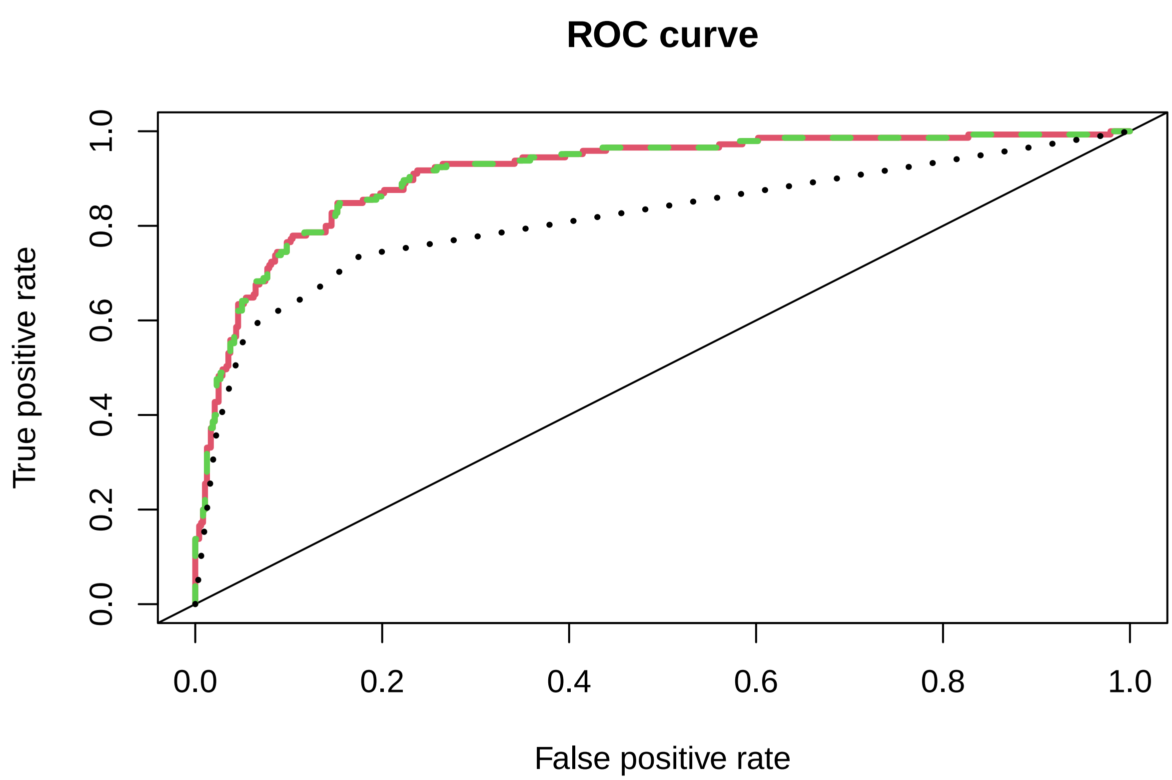
Here I save the training and test errors to the **records** variable.

```r
lasso.prob.train = predict(lasso.fit, s = bestlam.lasso, newx = x[idx.tr,], type =
↪  "class")
lasso.prob.test = predict(lasso.fit, s = bestlam.lasso, newx = x[-idx.tr,], type =
↪  "class")
lasso.train.error = calc_error_rate(lasso.prob.train, y.train)
lasso.test.error = calc_error_rate(lasso.prob.test, y.test)
records["lasso", ] <- c(lasso.train.error, lasso.test.error)
records
```

```
##          train.error test.error
## tree       0.1553865     0.1680
## logistic   0.1233480     0.1248
## lasso      0.1233480     0.1232
```

Next, I compute ROC curves for the decision tree, logistic regression and LASSO logistic regression using predictions on the test data and then display them on the same plot.

```r
#logistic
log.prob.test2 = predict(glm.fit, all.rename.te, type = "response")

log.prediction = prediction(log.prob.test2, all.rename.te$Poverty)
log.perf = performance(log.prediction, measure="tpr", x.measure="fpr")
plot(log.perf, col=2, lwd=3, main="ROC curve")
abline(0,1)

#lasso
lasso.prob.test2 = predict(lasso.fit, newx = x.test, type = "response")

lasso.prediction = prediction(lasso.prob.test2, y.test)
lasso.perf = performance(lasso.prediction, measure="tpr", x.measure="fpr")
lines(lasso.perf@x.values[[1]], lasso.perf@y.values[[1]], col = 3, lwd = 3, lty = 2 )

#tree
tree.all.2 = rpart(Poverty~., data = all.rename.tr, method = "class")
tree.prob.test2 = predict(tree.all.2, all.rename.te, type = "prob")[,2]
tree.pred = prediction(tree.prob.test2, all.rename.te$Poverty)
tree.perf = performance(tree.pred, measure = "tpr", x.measure = "fpr")
lines(tree.perf@x.values[[1]], tree.perf@y.values[[1]], col = 1, lwd = 3, lty = 3)
```



The ROC Curve demonstrates the extreme similarity of performance between Lasso and the unpenalized logistic regression. Both Lasso and Logistic regression preform relatively well while the decision tree method results in much less area under the ROC curve than the other two methods, which indicates less powerful performance. The pro of Lasso and Logistic Regression is that they preform better but the con is that they are less interpretable. The pro of Decision Trees is that they are more interpretable but do not preform as accurately.

However, the different classifiers are more appropriate for answering different kinds of questions about Poverty; Decision Tree analysis is more appropriate for visualization: it is very easy to understand the influence of predictors on the response variable even to people other than statisticians, while understanding of influence of predictors on the response variable for Lasso and Logistic Regression requires some knowledge of statistics. Decision Tree analysis maybe more appropriate for answering what populations greater than a calculated percentage live in a state with poverty greater than 20%, while Lasso and Logistic Regression may be more appropriate for predicting which states have poverty greater than 20% in relation to the population of those states.

Here, I use Random Forest and KNN as additional classification methods.

```
set.seed(123)
YTrain = all.rename.tr$Poverty
XTrain = all.rename.tr %>% select(-Poverty) %>% scale(center = TRUE, scale = TRUE)
YTest = all.rename.te$Poverty
XTest = all.rename.te%>% select(-Poverty) %>% scale(center = TRUE, scale = TRUE)
pred.YTtrain = knn(train = XTrain, test = XTrain, cl = YTrain, k = 2)

conf.train = table(predicted = pred.YTtrain, true = YTrain)
conf.train
```

```
##          true
## predicted    0    1
##         0 1940  240
##         1    0  317
```

```
1-sum(diag(conf.train)/sum(conf.train))
```

```
## [1] 0.09611534
```

```
pred.YTest = knn(train = XTrain, test = XTest, cl = YTrain, k = 2)
```

```
conf.test = table(predicted = pred.YTest, true = YTest)
conf.test
```

```
##          true
## predicted   0   1
##         0 461  88
##         1  19  57
```

```
knn.error = 1-sum(diag(conf.test)/sum(conf.test))
print(paste("the test error rate of KNN:", knn.error))
```

[1] "the test error rate of KNN: 0.1712"

```
rf = randomForest(Poverty~., data = all.rename.tr, mtry = 5, importance = TRUE)
rf
```

```
##
## Call:
##  randomForest(formula = Poverty ~ ., data = all.rename.tr, mtry = 5,      importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 5
##
##         OOB estimate of  error rate: 12.82%
## Confusion matrix:
##      0   1 class.error
## 0 1844  96  0.04948454
## 1  224 333  0.40215440
```

```
yhat.bag = predict(rf, newdata = all.rename.te, type = "class")
test.bag.err = mean(yhat.bag != all.rename.te$Poverty)
print(paste("the test error rate of random forest:", test.bag.err))
```

[1] "the test error rate of random forest: 0.1264"

```
records
```

```
##          train.error test.error
## tree       0.1553865     0.1680
## logistic   0.1233480     0.1248
## lasso      0.1233480     0.1232
```

As we can see from the above outputs, utilized methods in the order of least to greatest test error rate are Lasso, Logistic, Random Forest, Tree, and KNN. Therefore, Lasso and Logistic Regression remain more accurate than the additional chosen methods of Random Forest and KNN.

Prediction

Here I use regression models to predict the actual value of **Poverty** (before I transformed Poverty to a binary variable) by county as well as compare and contrast the results with the classification models.

```
all.num <- census.clean %>%
  left_join(education, by = c("State"="State", "County"="County")) %>%
  na.omit
all.num <- all.num %>% select(-c("State", "County"))
all.num.tr <- all.num[idx.tr, ]
all.num.te <- all.num[-idx.tr, ]
regression <- lm(Poverty ~., data = all.num.tr)
```

```
summary(regression)
```

```
##
## Call:
## lm(formula = Poverty ~ ., data = all.num.tr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -16.7042  -2.2381  -0.2397   1.9311  20.2446
##
## Coefficients: (1 not defined because of singularities)
##                                                  Estimate Std. Error t value
## (Intercept)                                     8.171e+01  4.611e+00  17.722
## TotalPop                                        3.631e-05  6.799e-06   5.340
## Men                                            -6.644e-01  3.388e-02 -19.607
## VotingAgeCitizen                                6.282e-02  2.047e-02   3.069
## Professional                                    2.146e-02  2.586e-02   0.830
## Service                                         1.873e-01  3.203e-02   5.847
## Office                                         -1.517e-02  3.475e-02  -0.436
## Production                                      1.974e-01  2.678e-02   7.371
## Drive                                          -9.755e-02  3.023e-02  -3.227
## Carpool                                        -3.570e-02  4.158e-02  -0.859
## Transit                                         1.185e-02  4.781e-02   0.248
## OtherTransp                                    -1.151e-01  7.248e-02  -1.588
## WorkAtHome                                     -6.833e-02  5.072e-02  -1.347
## MeanCommute                                    -1.346e-01  1.682e-02  -8.002
## Employed                                       -6.280e-01  1.750e-02 -35.887
## PrivateWork                                    -7.448e-02  1.756e-02  -4.241
## SelfEmployed                                   -1.128e-01  3.308e-02  -3.409
## FamilyWork                                      2.751e-01  1.868e-01   1.473
## Minority                                        8.249e-02  5.702e-03  14.467
## `Less than a high school diploma, 2015-19`     -4.737e-05  1.399e-05  -3.386
## `High school diploma only, 2015-19`            -5.072e-05  1.278e-05  -3.970
## `Some college or associate's degree, 2015-19` -7.240e-05  1.401e-05  -5.168
## `Bachelor's degree or higher, 2015-19`         -4.344e-05  9.269e-06  -4.687
## Total_Population                                       NA         NA      NA
##                                                Pr(>|t|)
## (Intercept)                                     < 2e-16 ***
## TotalPop                                       1.01e-07 ***
## Men                                             < 2e-16 ***
## VotingAgeCitizen                               0.002172 **
## Professional                                   0.406829
## Service                                        5.67e-09 ***
## Office                                         0.662548
## Production                                     2.30e-13 ***
## Drive                                          0.001266 **
## Carpool                                        0.390620
## Transit                                        0.804323
## OtherTransp                                    0.112395
## WorkAtHome                                     0.178067
## MeanCommute                                    1.86e-15 ***
## Employed                                        < 2e-16 ***
## PrivateWork                                    2.31e-05 ***
## SelfEmployed                                   0.000661 ***
## FamilyWork                                     0.140929
## Minority                                        < 2e-16 ***
## `Less than a high school diploma, 2015-19`     0.000722 ***
## `High school diploma only, 2015-19`            7.41e-05 ***
## `Some college or associate's degree, 2015-19` 2.55e-07 ***
## `Bachelor's degree or higher, 2015-19`         2.92e-06 ***
## Total_Population                                     NA
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.768 on 2474 degrees of freedom
## Multiple R-squared:  0.6685, Adjusted R-squared:  0.6656
## F-statistic: 226.8 on 22 and 2474 DF,  p-value: < 2.2e-16
```

```r
pred.regression = predict(regression, newdata = all.num.te, type = "response")
d <- data.frame(pred = pred.regression, actual = all.num.te$Poverty)
mean((d$actual - d$pred)^2)
```

```
## [1] 15.23157
```

I prefer the regression method because poverty rate is a much more flexible and useful indicator than simply "poverty or not." I may introduce bias into the model by designating a poverty line. A complimentary use for both methods may be to use classification methods to identify which counties may be most at risk for poverty and then use regression to predict the poverty rate for those counties that are deemed most at risk by classification.

Conclusion

All methods indicate **Men**, **Employment**, and **Minority** to be significant predictors of **Poverty** in a county; With **Men** and **Employment** being negatively correlated while **Minority** is positively correlated with **Poverty**. These results are logical because **Employment** is a direct implication of income; According to the Bureau of Labor Statistics "In 2017, women who were full-time wage and salary workers had median usual weekly earnings that were 82 percent of those of male full-time wage and salary workers"; And the American Psycological Association states that "Discrimination and marginalization can serve as a hindrance to upward mobility for ethnic and racial minorities seeking to escape poverty."

All of the methods found the variables **'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'** to be significant predictors which is also logical because education is known to be tied to income and social mobility. Our results could indicate that government assistance should be given to counties having large minority and unemployment populations. Additional data in counties with high poverty rates and large minority populations could be gathered in order to determine what characteristics of counties with large minority populations contribute to poverty; Likewise, additional data in counties with with high unemployment rates could be gathered in order to determine the causes of those high unemployment rates.