PSTAT 131 Final Project

Yuqing Xia Meredith Johnson

11/30/2021

Worked on by Yuqing Xia and Meredith Johnson

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'dplyr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

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

```
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") %>%
  select(-CountyId,-ChildPoverty,-Income,-IncomeErr,-IncomePerCap,-IncomePerCapErr) %>%
  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.
```

```
education <- read_csv("./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.
```

1. Report the dimension of census

```
dim(census)
```

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

```
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>
```

It has 31 columns and 3142 rows.

Are there missing values in the data set?

```
summary(census)
```

```
##     State              County             TotalPop             Men
##  Length:3142        Length:3142        Min.   :      74   Min.   :     39
##  Class :character   Class :character   1st Qu.:   10945   1st Qu.:   5514
##  Mode  :character   Mode  :character   Median :   25692   Median :  12798
##                                        Mean   :  102166   Mean   :  50292
##                                        3rd Qu.:   67445   3rd Qu.:  33481
##                                        Max.   :10105722   Max.   :4979641
##      Women            Hispanic          White            Black
##  Min.   :     35   Min.   : 0.000   Min.   :  0.60   Min.   : 0.000
##  1st Qu.:   5460   1st Qu.: 2.100   1st Qu.: 65.10   1st Qu.: 0.600
##  Median :  12885   Median : 4.000   Median : 84.20   Median : 2.100
##  Mean   :  51873   Mean   : 9.122   Mean   : 76.76   Mean   : 8.896
##  3rd Qu.:  34108   3rd Qu.: 9.300   3rd Qu.: 92.90   3rd Qu.: 9.875
##  Max.   :5126081   Max.   :99.200   Max.   :100.00   Max.   :86.900
##      Native           Asian          Pacific          VotingAgeCitizen
##  Min.   : 0.000   Min.   : 0.00   Min.   : 0.00000   Min.   :     59
##  1st Qu.: 0.100   1st Qu.: 0.30   1st Qu.: 0.00000   1st Qu.:   8279
##  Median : 0.300   Median : 0.60   Median : 0.00000   Median :  19480
##  Mean   : 1.812   Mean   : 1.32   Mean   : 0.08546   Mean   :  72223
##  3rd Qu.: 0.600   3rd Qu.: 1.20   3rd Qu.: 0.10000   3rd Qu.:  51224
##  Max.   :90.300   Max.   :41.80   Max.   :33.70000   Max.   :6218279
##     Poverty       Professional      Service          Office
##  Min.   : 2.40   Min.   :11.40   Min.   : 0.00   Min.   : 4.80
##  1st Qu.:11.30   1st Qu.:27.30   1st Qu.:15.70   1st Qu.:19.90
##  Median :15.20   Median :30.50   Median :17.80   Median :22.00
##  Mean   :15.99   Mean   :31.54   Mean   :18.12   Mean   :21.78
##  3rd Qu.:19.40   3rd Qu.:34.90   3rd Qu.:20.07   3rd Qu.:23.80
##  Max.   :52.00   Max.   :69.00   Max.   :46.40   Max.   :37.20
##   Construction     Production        Drive          Carpool
##  Min.   : 0.00   Min.   : 0.00   Min.   : 4.60   Min.   : 0.000
##  1st Qu.: 9.80   1st Qu.:11.60   1st Qu.:77.20   1st Qu.: 8.100
##  Median :12.20   Median :15.50   Median :81.00   Median : 9.500
##  Mean   :12.64   Mean   :15.93   Mean   :79.52   Mean   : 9.899
##  3rd Qu.:14.90   3rd Qu.:19.60   3rd Qu.:84.00   3rd Qu.:11.300
##  Max.   :36.40   Max.   :48.70   Max.   :97.20   Max.   :29.300
##     Transit             Walk          OtherTransp       WorkAtHome
##  Min.   : 0.0000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.1000   1st Qu.: 1.400   1st Qu.: 0.900   1st Qu.: 2.900
##  Median : 0.3000   Median : 2.300   Median : 1.300   Median : 4.100
##  Mean   : 0.9368   Mean   : 3.235   Mean   : 1.603   Mean   : 4.803
##  3rd Qu.: 0.8000   3rd Qu.: 3.800   3rd Qu.: 1.900   3rd Qu.: 5.800
##  Max.   :61.8000   Max.   :59.200   Max.   :43.200   Max.   :33.000
##   MeanCommute       Employed         PrivateWork       PublicWork
```

```
##  Min.   : 5.10   Min.   :      39   Min.   :31.10   Min.   : 4.40
##  1st Qu.:19.60   1st Qu.:    4550   1st Qu.:71.70   1st Qu.:12.70
##  Median :23.10   Median :  10695   Median :76.30   Median :15.70
##  Mean   :23.35   Mean   :  47931   Mean   :75.07   Mean   :16.89
##  3rd Qu.:26.90   3rd Qu.:  29515   3rd Qu.:80.30   3rd Qu.:19.50
##  Max.   :45.10   Max.   :4805817   Max.   :88.80   Max.   :64.80
##   SelfEmployed      FamilyWork      Unemployment
##  Min.   : 0.000   Min.   :0.0000   Min.   : 0.000
##  1st Qu.: 5.200   1st Qu.:0.1000   1st Qu.: 4.400
##  Median : 6.800   Median :0.2000   Median : 6.100
##  Mean   : 7.758   Mean   :0.2824   Mean   : 6.364
##  3rd Qu.: 9.175   3rd Qu.:0.3000   3rd Qu.: 7.800
##  Max.   :38.000   Max.   :8.0000   Max.   :28.800
```

```
sum(is.na(census))
```

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

There is no missing value in the data set.

Compute the total number of distinct values in State in census to verify that the data contains all states and a federal district.

```
length(unique(census$State))
```

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

There are 51 unique values in State column thus the data contains all states and a federal district.

2. Report the dimension of education.

```
dim(education)
```

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

How many distinct counties contain missing values in the data set?

```
rows_na = education[rowSums(is.na(education)) > 0, ]
length(unique(rows_na$County))
```

```
## [1] 18
```

18 distinct counties contain missing values in the data set.

Compute the total number of distinct values in County in education.

```
length(unique(education$County))
```

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

1877 distinct values in County in education

Compare the values of total number of distinct county in education with that in census.

```r
length(unique(census$County))
```

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

Comment on your findings

The total number of distinct county in education in census and education are the same.

3. Remove all NA values in education, if there is any.

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

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

4. 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.

```r
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 sc~ `High school diplo~ `Some college or ass~
##   <chr> <chr>                    <dbl>               <dbl>                 <dbl>
## 1 AL    Autauga~                  4291               12551                 10596
## 2 AL    Baldwin~                 13893               41797                 47274
## 3 AL    Barbour~                  4812                6396                  4676
## 4 AL    Bibb Co~                  3386                7256                  3848
## 5 AL    Blount ~                  7763               13299                 13519
## 6 AL    Bullock~                  1798                2860                  1587
## # ... with 2 more variables: Bachelor's degree or higher, 2015-19 <dbl>,
## #   Total_Population <dbl>
```

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

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

head(education.state)
```

```
## # A tibble: 6 x 5
##   State ‘Less than a high~ ‘High school dip~ ‘Some college or~ ‘Bachelor’s degr~
##   <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
```

6. 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.

```
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~ ‘High school dip~ ‘Some college or~ ‘Bachelor’s degr~
##   <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 1 more variable:
## #   name of the education degree level with the largest population <chr>
```

```
states <- map_data("state")
head(states)
```

```
##        long      lat group order  region subregion
## 1 -87.46201 30.38968     1     1 alabama      <NA>
## 2 -87.48493 30.37249     1     2 alabama      <NA>
## 3 -87.52503 30.37249     1     3 alabama      <NA>
## 4 -87.53076 30.33239     1     4 alabama      <NA>
## 5 -87.57087 30.32665     1     5 alabama      <NA>
## 6 -87.58806 30.32665     1     6 alabama      <NA>
```

7. Now color the map (on the state level) by the education level with highest population for each state. Show the plot legend.

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

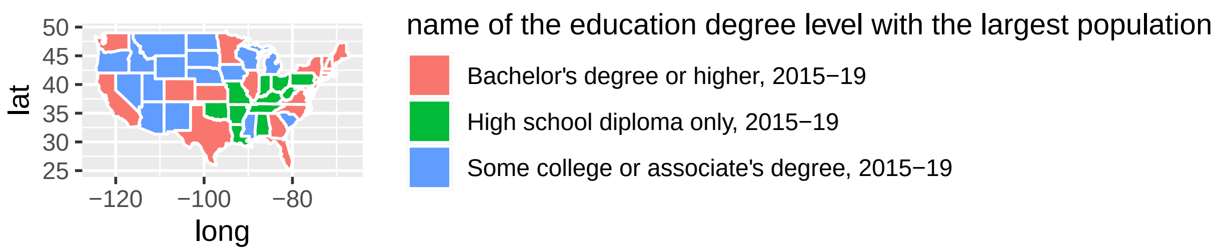
```
##        long      lat group order region subregion
```

```
## 1 -87.46201 30.38968     1     1     AL      <NA>
## 2 -87.48493 30.37249     1     2     AL      <NA>
## 3 -87.52503 30.37249     1     3     AL      <NA>
## 4 -87.53076 30.33239     1     4     AL      <NA>
## 5 -87.57087 30.32665     1     5     AL      <NA>
## 6 -87.58806 30.32665     1     6     AL      <NA>
```

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

```
##        long      lat group order region subregion
## 1 -87.46201 30.38968     1     1     AL      <NA>
## 2 -87.48493 30.37249     1     2     AL      <NA>
## 3 -87.52503 30.37249     1     3     AL      <NA>
## 4 -87.53076 30.33239     1     4     AL      <NA>
## 5 -87.57087 30.32665     1     5     AL      <NA>
## 6 -87.58806 30.32665     1     6     AL      <NA>
##   Less than a high school diploma, 2015-19 High school diploma only, 2015-19
## 1                                   458922                           1022839
## 2                                   458922                           1022839
## 3                                   458922                           1022839
## 4                                   458922                           1022839
## 5                                   458922                           1022839
## 6                                   458922                           1022839
##   Some college or associate's degree, 2015-19
## 1                                      993344
## 2                                      993344
## 3                                      993344
## 4                                      993344
## 5                                      993344
## 6                                      993344
##   Bachelor's degree or higher, 2015-19
## 1                               845772
## 2                               845772
## 3                               845772
## 4                               845772
## 5                               845772
## 6                               845772
##   name of the education degree level with the largest population
## 1                              High school diploma only, 2015-19
## 2                              High school diploma only, 2015-19
## 3                              High school diploma only, 2015-19
## 4                              High school diploma only, 2015-19
## 5                              High school diploma only, 2015-19
## 6                              High school diploma only, 2015-19
```

```r
ggplot(data = left_join_data) +
  geom_polygon(aes(x = long, y = lat, fill = `name of the education degree level with the largest popula
```



8. (Open-ended) Create a visualization of your choice using census data.

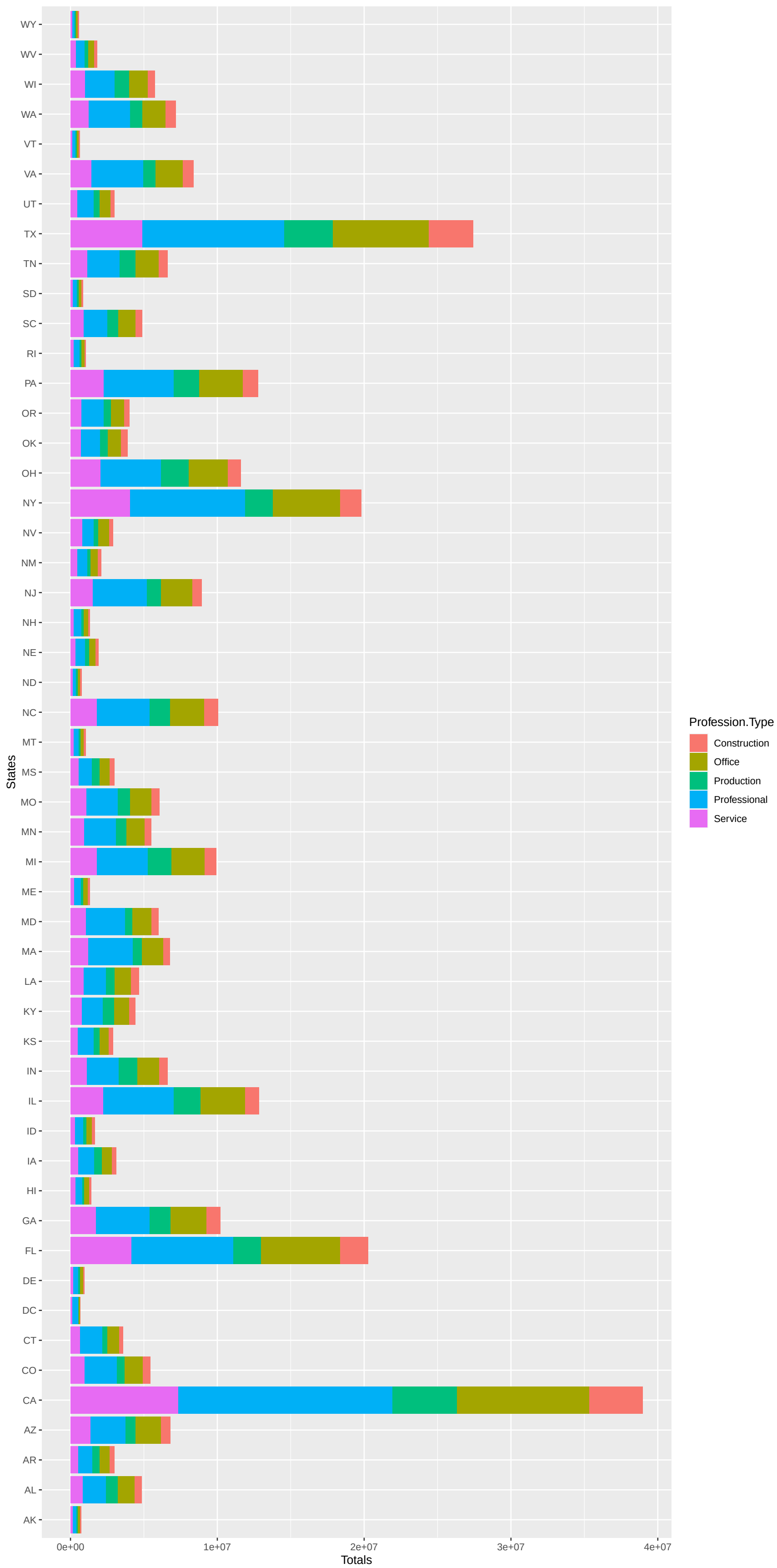
```
profession.percent <- census %>% select(State, TotalPop, Professional, Service, Office, Construction, Pr

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(acro

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

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

#install.packages("reshape")
library(reshape)
```

```
## Warning: package 'reshape' was built under R version 4.0.5

##
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':
```

```
##
##     rename
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
```

```
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
Construction
Office
Production
Professional
Service

9. The census data contains county-level census information. In this problem, we clean and aggregate the information as follows.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}.

```
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 TotalPop   Men  Women White VotingAgeCitizen Poverty Professional
##   <chr> <chr>     <dbl> <dbl>  <dbl> <dbl>            <dbl>   <dbl>        <dbl>
## 1 AL    Autau~    55036  48.9  28137  75.4             74.5    13.7         35.3
## 2 AL    Baldw~   203360  48.9 103833  83.1             76.4    11.8         35.7
## 3 AL    Barbo~    26201  53.3  12225  45.7             77.4    27.2         25
## 4 AL    Bibb ~    22580  54.3  10329  74.6             78.2    15.2         24.4
## 5 AL    Bloun~    57667  49.4  29177  87.4             73.7    15.6         28.5
## 6 AL    Bullo~    10478  53.6   4862  21.6             78.4    28.5         19.7
## # ... with 14 more variables: Service <dbl>, Office <dbl>, Production <dbl>,
## #   Drive <dbl>, Carpool <dbl>, Transit <dbl>, OtherTransp <dbl>,
## #   WorkAtHome <dbl>, MeanCommute <dbl>, Employed <dbl>, PrivateWork <dbl>,
## #   SelfEmployed <dbl>, FamilyWork <dbl>, Minority <dbl>
```

(Note that many columns are perfectly collineared, 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 we can know that Women and TotalPop are highly correlated, Minority and White are highly correlated

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

10. Print the first 5 rows of census.clean

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

```
## # A tibble: 5 x 21
##   State County      TotalPop   Men VotingAgeCitizen Poverty Professional Service
##   <chr> <chr>          <dbl> <dbl>            <dbl>   <dbl>        <dbl>   <dbl>
## 1 AL    Autauga Co~    55036  48.9             74.5    13.7         35.3    18
## 2 AL    Baldwin Co~   203360  48.9             76.4    11.8         35.7    18.2
## 3 AL    Barbour Co~    26201  53.3             77.4    27.2         25      16.8
## 4 AL    Bibb County    22580  54.3             78.2    15.2         24.4    17.6
## 5 AL    Blount Cou~    57667  49.4             73.7    15.6         28.5    12.9
## # ... with 13 more variables: Office <dbl>, Production <dbl>, Drive <dbl>,
## #   Carpool <dbl>, Transit <dbl>, OtherTransp <dbl>, WorkAtHome <dbl>,
## #   MeanCommute <dbl>, Employed <dbl>, PrivateWork <dbl>, SelfEmployed <dbl>,
## #   FamilyWork <dbl>, Minority <dbl>
```

11. Run PCA for the cleaned county level census data (with State and County excluded).

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

Save the first two principle components PC1 and PC2 into a two-column data frame, call it pc.county.

```r
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
```

Discuss whether you chose to center and scale the features before running PCA and the reasons for your choice.

We chose to center and scale the features before running PCA because features need to be centered before PCA is performed and features were recorded on different scales; Several groups of features seem to be recorded as percentages of the population like race or commute type.

What are the three features with the largest absolute values of the first principal component?

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

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

WorkAtHome, SelfEmployed, Drive are the three features with the largest absolute values of the first principal component.

Which features have opposite signs and what does that mean about the correlation between these features?

```r
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 with the most significant principle loadings, "WorkAtHome", "SelfEmployed", and "Professional" have positive signs while "Drive" and "PrivateWork" have negative signs. Positive loadings indicate a feature and a principal component 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.

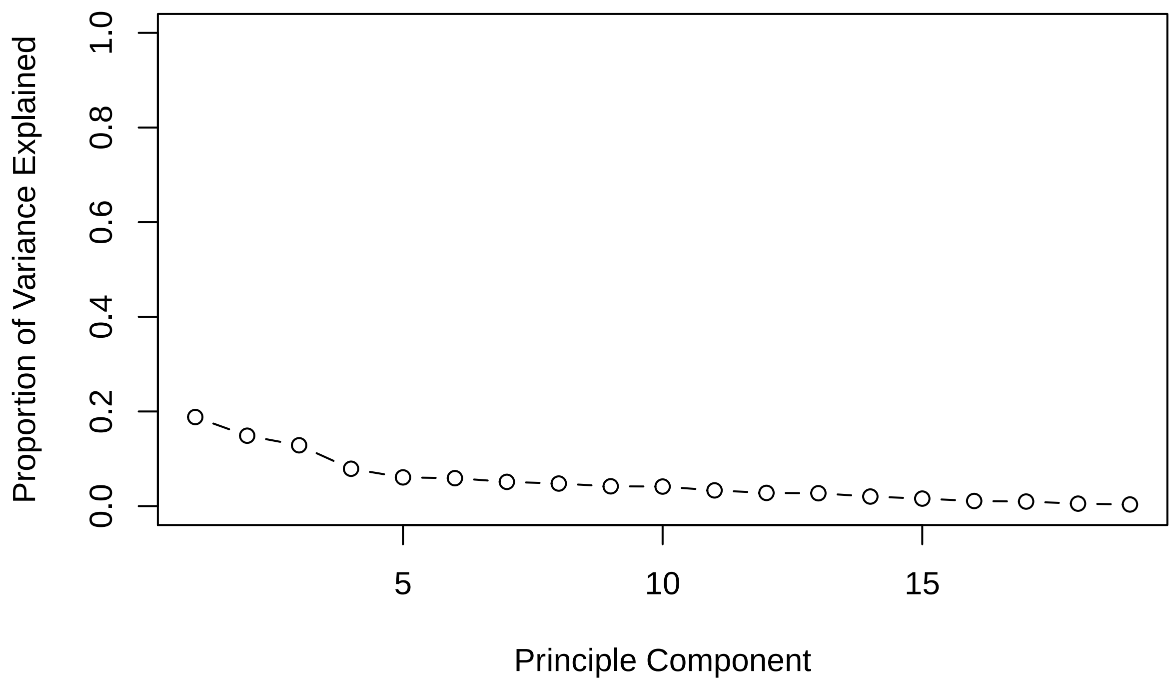
12. Determine the number of minimum number of PCs needed to capture 90% of the variance for the analysis.

```r
pr.var = pr.out$sdev^2
pve = pr.var/sum(pr.var)
min(which(cumsum(pve) > .9))
```

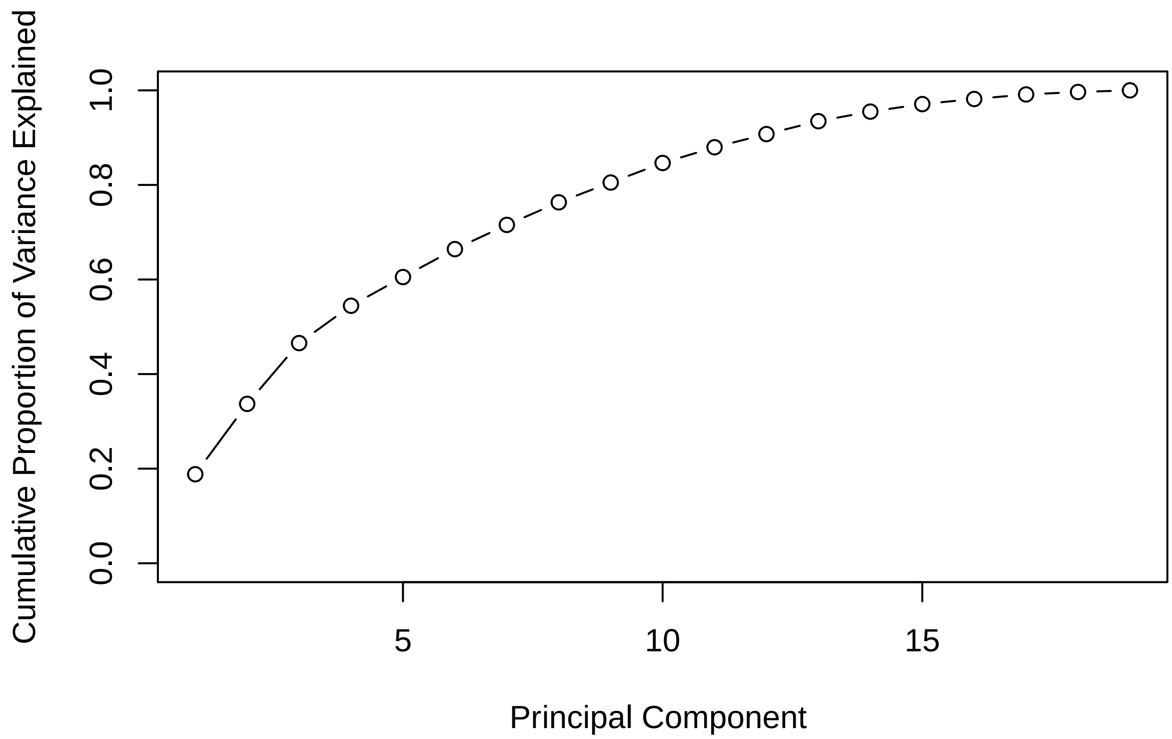
```
## [1] 12
```

Plot proportion of variance explained (PVE) and cumulative PVE.

```r
plot(pve, xlab = "Principle 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')
```



13. With census.clean (with State and County excluded), 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)
```

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
```

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
```

Compare the results and comment on your observations. For both approaches investigate the cluster that contains Santa Barbara County.

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

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

```
clus2[index]
```

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

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

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

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

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

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

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

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

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

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

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.

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 does not contain meaningful analytical value. The second approach organizes Counties into more evenly distributed clusters.

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

14. 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.

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

```
## # A tibble: 6 x 23
##   TotalPop   Men VotingAgeCitizen Poverty Professional Service Office Production
##      <dbl> <dbl>            <dbl> <fct>          <dbl>   <dbl>  <dbl>      <dbl>
## 1    55036  48.9             74.5 0               35.3    18     23.2       15.4
## 2   203360  48.9             76.4 0               35.7    18.2   25.6       10.8
## 3    26201  53.3             77.4 1               25      16.8   22.6       24.1
## 4    22580  54.3             78.2 0               24.4    17.6   19.7       22.4
## 5    57667  49.4             73.7 0               28.5    12.9   23.3       19.5
## 6    10478  53.6             78.4 1               19.7    17.1   18.6       30.6
## # ... with 15 more variables: Drive <dbl>, Carpool <dbl>, 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>
```

Partition the dataset into 80% training and 20% test data. Make sure to set.seed before the partition.

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

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))
```

Using the following error rate function. And the object records is used to record the classification performance of each method in the subsequent problems.

```
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

15. Decision tree: train a decision tree by cv.tree().

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 4.0.5
```

```r
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.0.5

## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

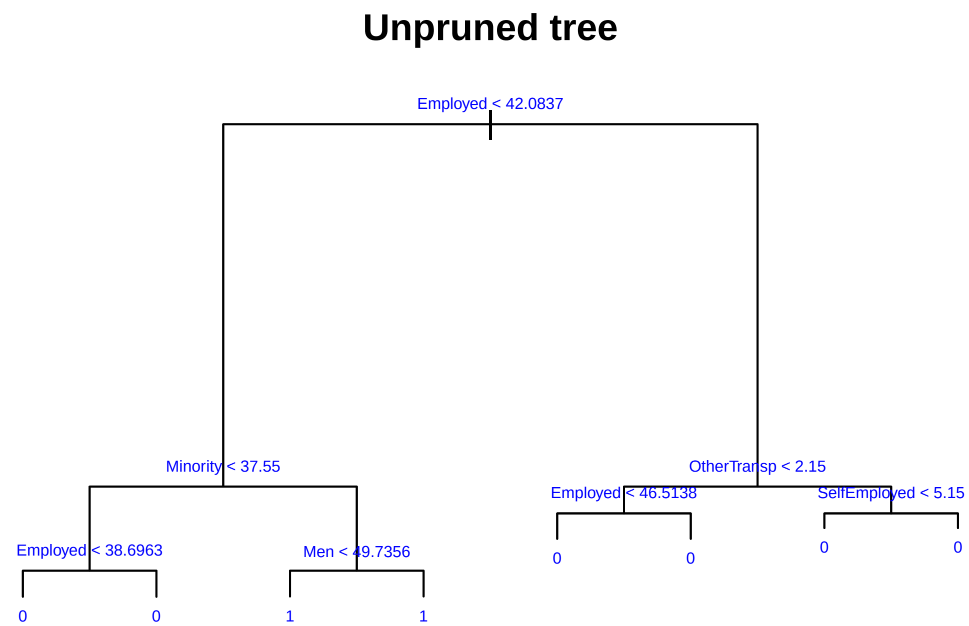
```r
library(maptree)
```

```
## Warning: package 'maptree' was built under R version 4.0.5

## Loading required package: cluster

## Loading required package: rpart
```

```r
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
```

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



Prune tree to minimize misclassification error. Be sure to use the folds from above for cross-validation.

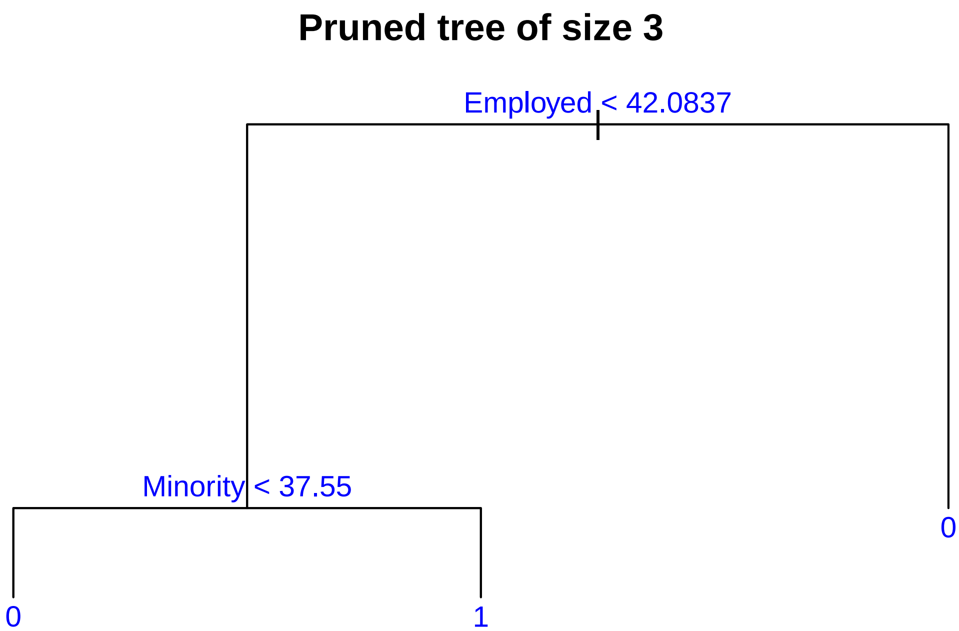
```
set.seed(1)

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 with that minimum rate:", best_size))
```

```
## [1] "Smallest tree size with that minimum rate: 3"
```

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

Visualize the trees before and after pruning.

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



Save training and test errors to 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
```

Interpret and discuss the results of the decision tree analysis.

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.

Use this plot to tell a story about Poverty.

A population that is less employed has less income and insufficient income is indicative of poverty. This decision tree indicates that states with larger minority populations as well as less employment are more likely to be in poverty; this may be a sign that states with larger minority populations, in less employed states, are more likely to have populations with insufficient income and therefore, be in poverty.

16. Run a logistic regression to predict Poverty in each county.

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

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

Save training and test errors to records variable.

```
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)
```

What are the significant variables?

```
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
```

Are they consistent with what you saw in decision tree analysis?

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, we find that the significant logistic regression variables are fairly consistent with the significant decision tree analysis variables.

Interpret the meaning of a couple of the significant coefficients in terms of a unit change in the variables.

The variable Men has a coefficient -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. The variable Employed has a coefficient -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. The variable Minority has a coefficient 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.

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.

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 λ.

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.0.5
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:reshape':
##
##     expand
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-1
```

```
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 =
```

What is the optimal value of λ in cross validation?

```
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"
```

What are 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"
##                               1
## (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
```

How do they compare to the unpenalized logistic regression?

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.

Comment on the comparison.

The similarities in coefficients may explain their same training errors.

Save training and test errors to the records variable.

```
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
```

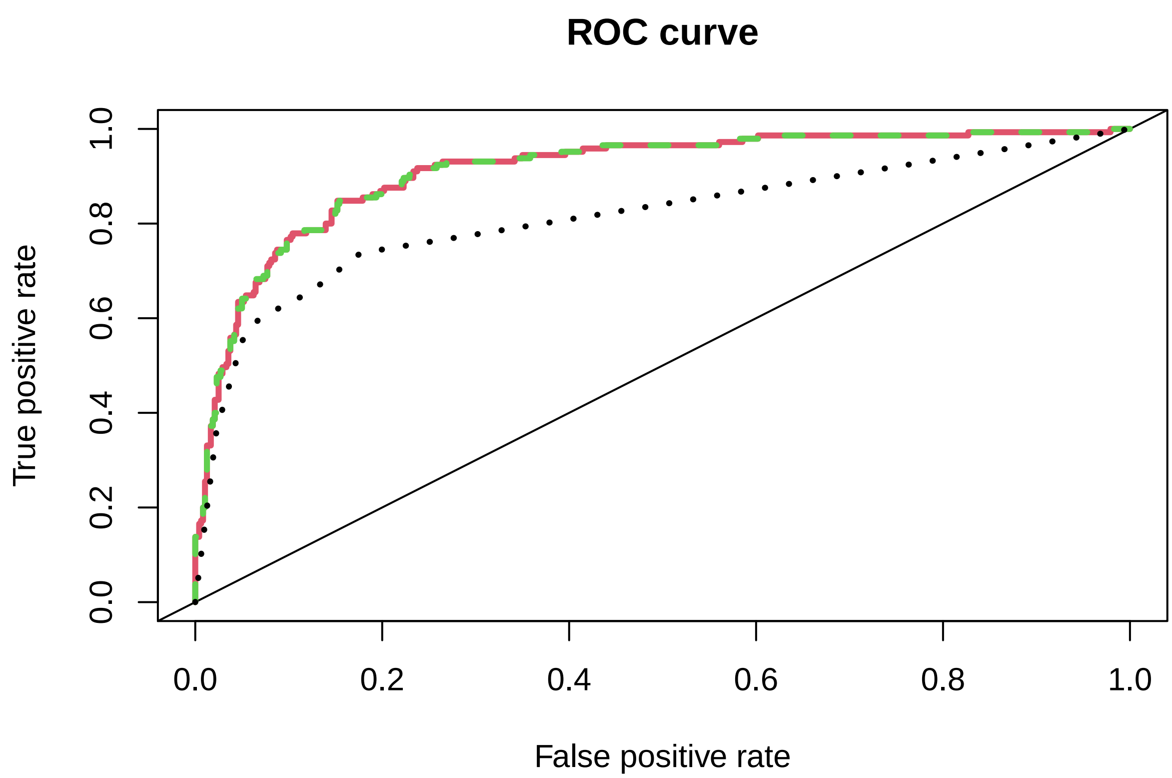
```
##          train.error test.error
## tree       0.1553865     0.1680
## logistic   0.1233480     0.1248
## lasso      0.1233480     0.1232
```

18. 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.

```
library("ROCR")
```

```
## Warning: package 'ROCR' was built under R version 4.0.5
```

```
#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
library(rpart)
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)
```



Based on your classification results, discuss the pros and cons of the various methods.

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.

Are the different classifiers more appropriate for answering different kinds of questions about Poverty?

Yes; 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. However, 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.

19. 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).

```
library("FNN")
```

```
## Warning: package 'FNN' was built under R version 4.0.5
```

```r
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
```

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

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

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

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

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

```r
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"
```

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.0.5
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
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
```

How do these compare to the tree method, logistic regression, and the lasso logistic regression?

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.

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

Consider a regression problem! Use regression models to predict the actual value of Poverty (before we transformed Poverty to a binary variable) by county. Compare and contrast these results with the classification models. Which do you prefer and why? How might they complement one another?

```
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)
```

```r
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")
```

```
## Warning in predict.lm(regression, newdata = all.num.te, type = "response"):
## prediction from a rank-deficient fit may be misleading
```

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

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

We prefer the regression method because poverty rate is a much more flexible and useful indicator than simply "poverty or not." We 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.

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).

All methods indicated Men, Employment, and Minority to be significant predictors of Poverty in a state; 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, Men are more likely to make more money, and Minorities are given less opportunities and subject to discrimination which may be a cause for less income and therefore 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 states having large minority and unemployment populations. Additional data in states with high poverty rates could be gathered regarding unemployment and Minority populations in order to predict when those populations will be significant predictors of poverty.