FINAL PRESENTATION CODE

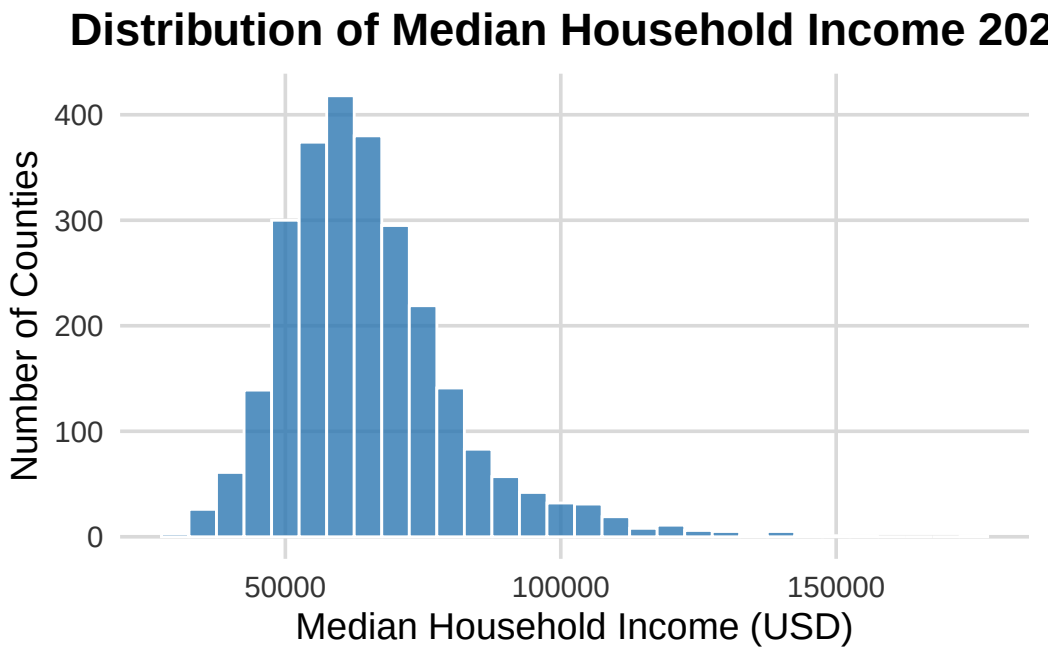
```
pacman::p_load(ggplot2, dplyr, maps, rpart, randomForest, tinytex)
```

Reading in Data; Encoding Factor Variables

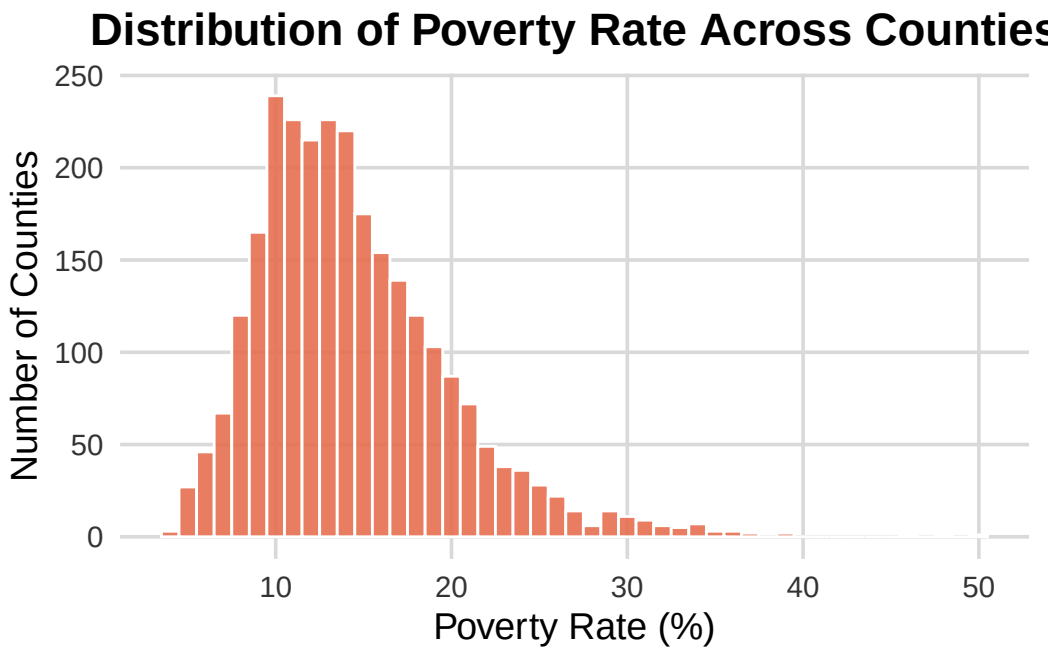
```
county_data <- read.csv("county_data.csv")
county_data$Voted_Democrat <- as.factor(county_data$Voted_Democrat)
county_data$State <- as.factor(county_data$State)
```

Data Visualizations

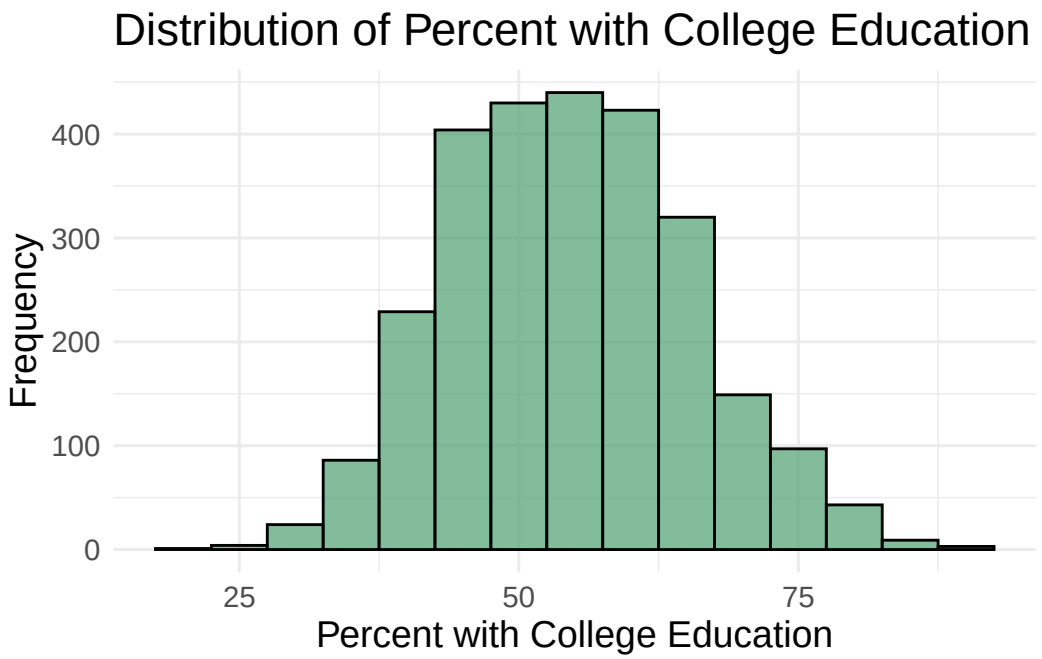
```
#Histogram of Median HH Income

ggplot(county_data, aes(x = MedianHouseholdIncome)) +
  geom_histogram(binwidth = 5000, fill = "#2C77B2", color = "white", alpha = 0.8) +
  labs(
    title = "Distribution of Median Household Income 2023",
    x = "Median Household Income (USD)",
    y = "Number of Counties"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.text = element_text(color = "gray20"),
    panel.grid.major = element_line(color = "gray85"),
    panel.grid.minor = element_blank()
  )
```



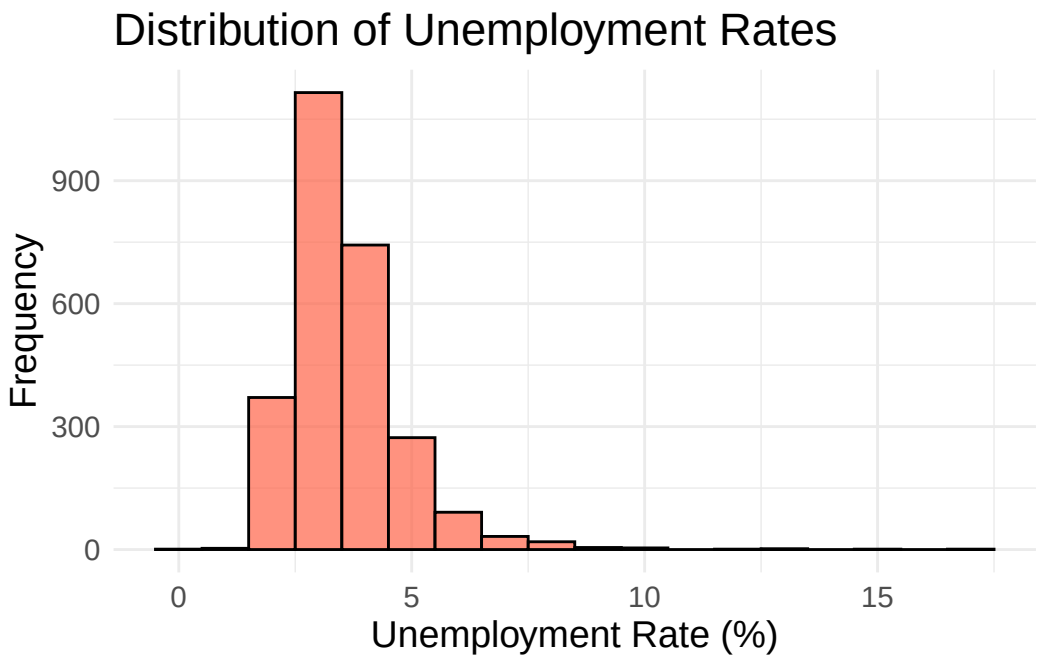

```
# Histogram of Poverty Rate

ggplot(county_data, aes(x = Percent_in_Poverty)) +
  geom_histogram(binwidth = 1, fill = "#E76F51", color = "white", alpha = 0.9) +
  labs(
    title = "Distribution of Poverty Rate Across Counties",
    x = "Poverty Rate (%)",
    y = "Number of Counties"
  ) +
  theme_minimal(base_size = 14) +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.5),
    axis.text = element_text(color = "gray20"),
    panel.grid.major = element_line(color = "gray85"),
    panel.grid.minor = element_blank()
  )
```



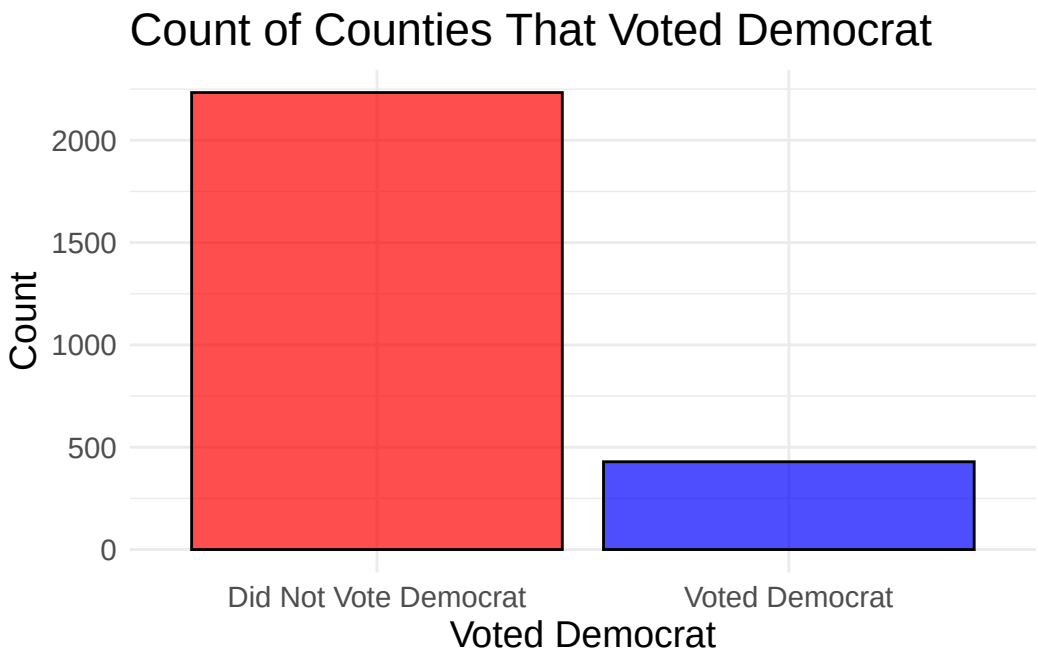

```
#Histogram of Percent College Education

ggplot(county_data, aes(x = Percent_College_Edu)) +
  geom_histogram(binwidth = 5, fill = "#4C9F70", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Percent with College Education",
       x = "Percent with College Education",
       y = "Frequency") +
  theme_minimal() +
  theme(text = element_text(size = 14))
```




```
#Histogram of Unemployment Rate

ggplot(county_data, aes(x = Unemployment_Rate)) +
  geom_histogram(binwidth = 1, fill = "#FF6347", color = "black", alpha = 0.7) +
  labs(title = "Distribution of Unemployment Rates",
       x = "Unemployment Rate (%)",
       y = "Frequency") +
  theme_minimal() +
  theme(text = element_text(size = 14))
```



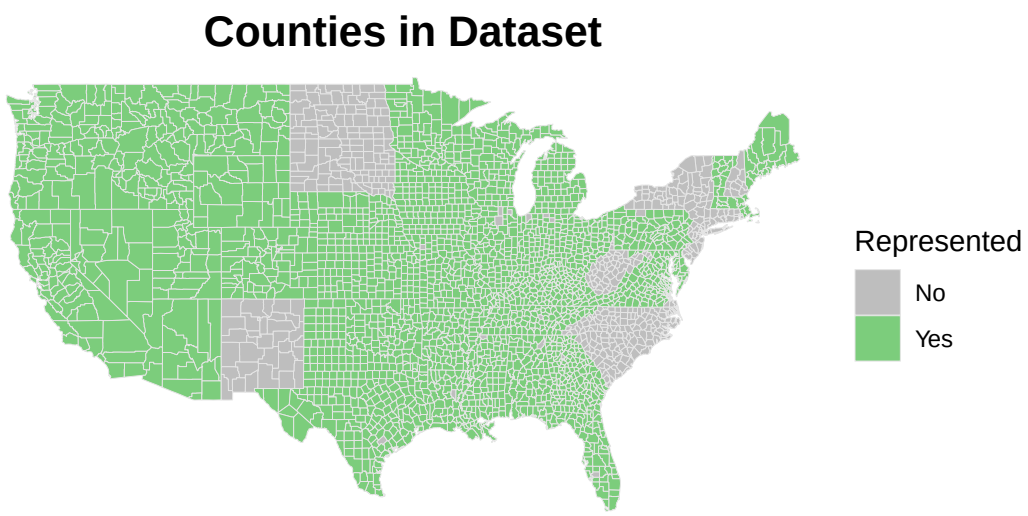

```
#Barplot of County Vote Outcome

ggplot(county_data, aes(x = Voted_Democrat)) +
  geom_bar(fill = c("red", "blue"), color = "black", alpha = 0.7) +
  labs(title = "Count of Counties That Voted Democrat",
       x = "Voted Democrat",
       y = "Count") +
  scale_x_discrete(labels = c("FALSE" = "Did Not Vote Democrat", "TRUE" = "Voted Democrat")) +
  theme_minimal() +
  theme(text = element_text(size = 14))
```




```
#Map of States included in Data Set (doesn't show AL or HI :( )


# map data is full state name, our data is abbreviations.
#So doing some data joining/cleaning to get map to show states properly.

state_lookup <- data.frame(
  State = state.abb,
  FullName = tolower(state.name)
)

county_data_clean <- county_data %>%
  left_join(state_lookup, by = c("State")) %>%
  mutate(region = FullName,
         subregion = tolower(County)) %>%
  mutate(
    subregion = gsub(" county| parish| borough", "", subregion),
    subregion = gsub("[.']", "", subregion),
    subregion = gsub("saint", "st", subregion),
    subregion = trimws(subregion)
  )
county_map <- map_data("county")
county_map_joined <- county_map %>%
```

```r
  left_join(county_data_clean, by = c("region", "subregion"))

# Finally plotting map
ggplot() +
  geom_polygon(data = county_map_joined,
               aes(x = long, y = lat, group = group,
                   fill = !is.na(County)),
               color = "gray90", size = 0.1) +
  scale_fill_manual(values = c("grey", "palegreen3"),
                    name = "Represented",
                    labels = c("No", "Yes")) +
  coord_fixed(1.3) +
  labs(title = "Counties in Dataset") +
  theme_void() +
  theme(legend.position = "right",
        plot.title = element_text(hjust = 0.5, size = 16, face = "bold"))
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



MULTIPLE LINEAR REGRESSION MODEL

LOGISTIC REGRESSION MODEL

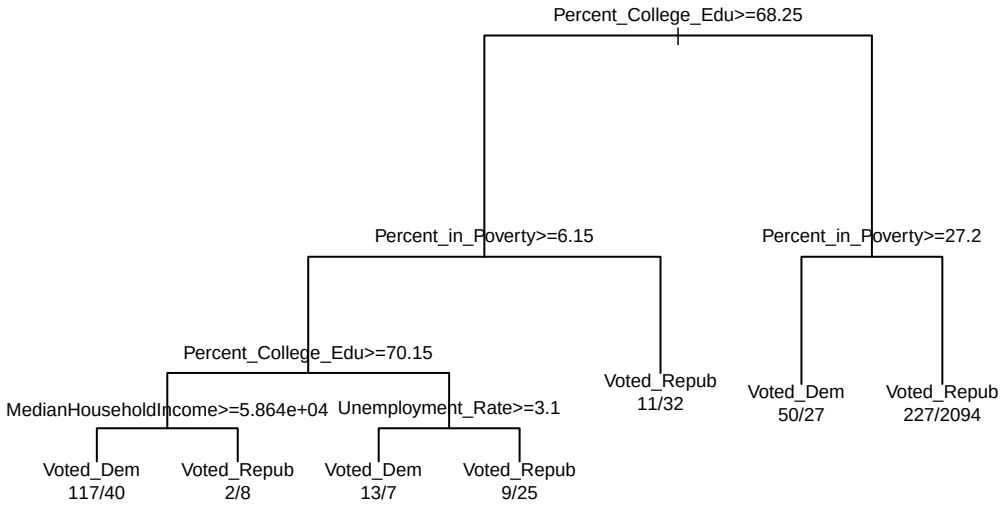
RANDOM FOREST MODEL

RF Model Without State as a Possible Predictor

```
#renaming levels in response variable for clarity
county_data$Voted_Democrat2 <- factor(county_data$Voted_Democrat,
levels = c("TRUE","FALSE"), labels=c("Voted_Dem", "Voted_Repub"))
table(county_data$Voted_Democrat2)
```

```
  Voted_Dem Voted_Repub
        429        2233
```

```
#creating a new dataframe with just the variables I want for this model
crop_county_data <- county_data %>%
  select(MedianHouseholdIncome,
         Percent_in_Poverty,
         Unemployment_Rate,
         Percent_College_Edu,
         Voted_Democrat2)

#setting seed for reproducability
set.seed(1992)


# plotting the rpart tree
t1 <- rpart(Voted_Democrat2~Percent_in_Poverty +
              Percent_College_Edu + Unemployment_Rate
            + MedianHouseholdIncome, data=crop_county_data)
par(cex=0.6, xpd=NA)
plot(t1)
text(t1, use.n=T)
```




```
#fitting the random forest
rf1 <- randomForest(Voted_Democrat2~ .,   data=crop_county_data)

#getting model predictions
rf_preds <- predict(rf1, type = "response")

# make confusion matrix
tb <- table(actual = county_data$Voted_Democrat2, predicted = rf_preds)
addmargins(tb)
```

```
             predicted
actual        Voted_Dem Voted_Repub  Sum
  Voted_Dem         204         225  429
  Voted_Repub        97        2136 2233
  Sum               301        2361 2662
```

```
# Accuracy (percent correctly classified)
(204+2136)/2662
```

```
[1] 0.8790383
```

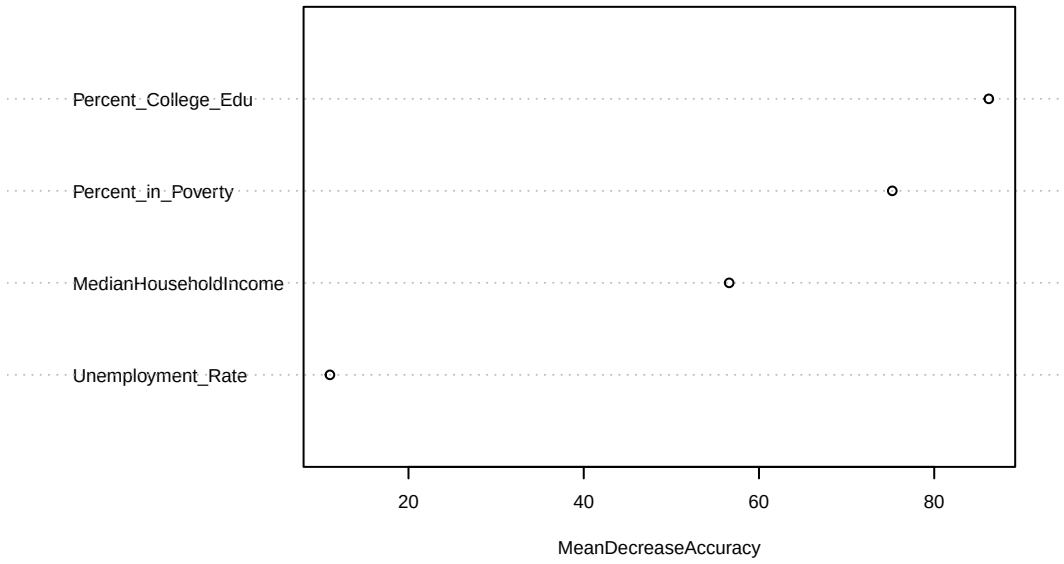
```
# Sensitivity (percent of Voted Dems correctly classified)
204/429
```

```
[1] 0.4755245
```

```
# Specificity (percent of Didn't Vote Dem correctly classified)
2136/2233
```

```
[1] 0.9565607
```

```
##VARIABLE IMPORTANCE
rf1.2 <- randomForest(Voted_Democrat2~ ., data=crop_county_data, importance = TRUE)
varImpPlot(rf1.2, type = 1, n.var = 4, main = "")
```




```
imp <- importance(rf1.2)
print(imp)
```

```
                     Voted_Dem Voted_Repub MeanDecreaseAccuracy
MedianHouseholdIncome 32.878746   41.945521             56.60037
Percent_in_Poverty    12.164016   68.925769             75.21239
```

```
Unemployment_Rate      7.276604    7.680806             11.03636
Percent_College_Edu   47.050964   67.664885             86.22928
                     MeanDecreaseGini
MedianHouseholdIncome         188.3299
Percent_in_Poverty            188.7656
Unemployment_Rate             106.7595
Percent_College_Edu           234.4018
```

```
############

#TRYING TO PENALIZE THE SYSTEM
#FOR MISCLASSIFYING COUNTIES THAT VOTED DEMOCRAT;
#does not improve, actually makes worse.

#########

rf2 <- randomForest(Voted_Democrat2~ ., data=crop_county_data,
                    classwt = c("Voted_Dem" = 20, "Voted_Repub" = 1))


rf_preds2 <- predict(rf2, type = "response")

# make confusion matrix
tb2 <- table(actual = county_data$Voted_Democrat2, predicted = rf_preds2)
addmargins(tb2)
```

```
             predicted
actual        Voted_Dem Voted_Repub  Sum
  Voted_Dem         172         257  429
  Voted_Repub        73        2160 2233
  Sum               245        2417 2662
```

```
# Accuracy (percent correctly classified)
(170+2166)/2662
```

```
[1] 0.8775357
```

```
# Sensitivity (percent of Voted Dems correctly classified)
170/429
```

```
[1] 0.3962704
```

```
# Specificity (percent of Didn't Vote Dem correctly classified)
2166/2233
```

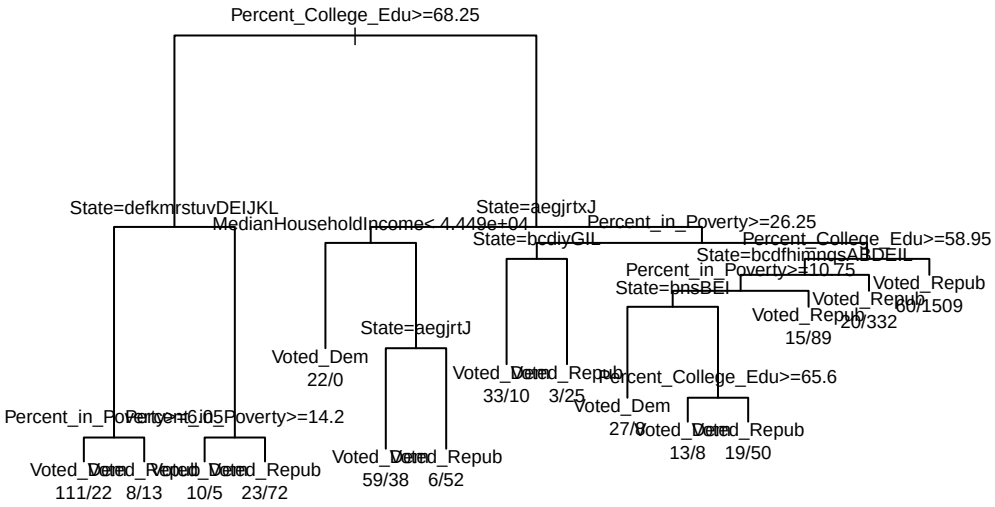
```
[1] 0.9699955
```

RF Model With State as a Possible Predictor

```
#renaming levels in response variable for clarity
county_data$Voted_Democrat2 <- factor(county_data$Voted_Democrat,
levels = c("TRUE","FALSE"), labels=c("Voted_Dem", "Voted_Repub"))
table(county_data$Voted_Democrat2)
```

```
  Voted_Dem Voted_Repub
        429        2233
```

```
#creating a new dataframe with just the variables I want for this model
crop_county <- county_data %>%
  select(MedianHouseholdIncome,
         Percent_in_Poverty,
         Unemployment_Rate,
         Percent_College_Edu,
         Voted_Democrat2,
         State)

#setting seed for reproducability
set.seed(1992)


# plotting the rpart tree
t3 <- rpart(Voted_Democrat2~ State + Percent_in_Poverty +
              Percent_College_Edu + Unemployment_Rate +
              MedianHouseholdIncome,   data=crop_county)
par(cex=0.6, xpd=NA)
plot(t3)
text(t3, use.n=T)
```




```
#fitting the random forest
rf3 <- randomForest(Voted_Democrat2~ ., data=crop_county)

#getting model predictions
rf_preds3 <- predict(rf3, type = "response")

# make confusion matrix
tb3 <- table(actual = crop_county$Voted_Democrat2, predicted = rf_preds3)
addmargins(tb3)
```

```
             predicted
actual        Voted_Dem Voted_Repub  Sum
  Voted_Dem         215         214  429
  Voted_Repub        61        2172 2233
  Sum               276        2386 2662
```

```
# Accuracy (percent correctly classified)
(215+2172)/2662
```

```
[1] 0.8966942
```

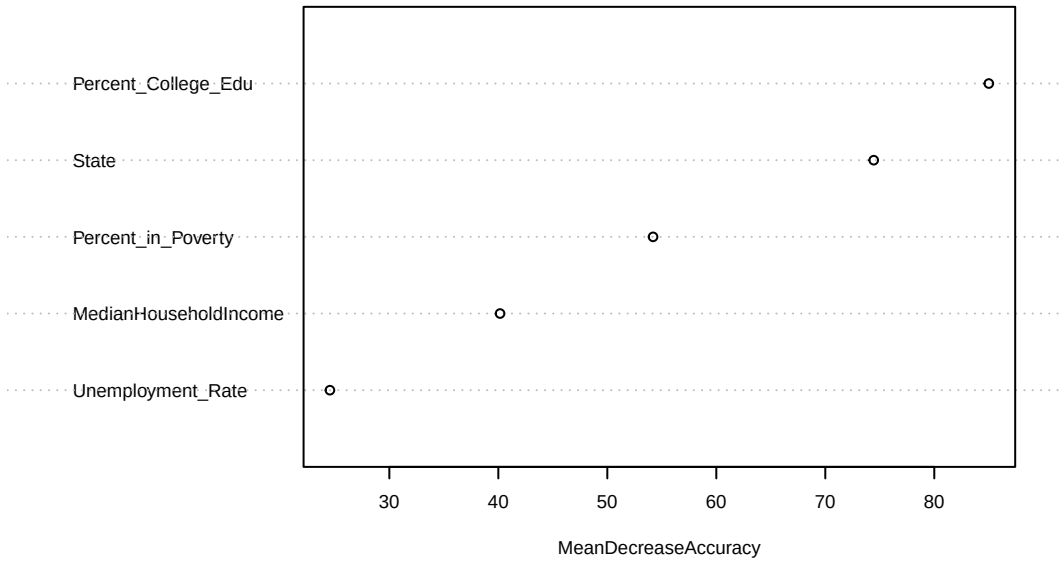
```
# Sensitivity (percent of Voted Dems correctly classified)
215/429
```

```
[1] 0.5011655
```

```
# Specificity (percent of Didn't Vote Dem correctly classified)
2172/2233
```

```
[1] 0.9726825
```

```
##VARIABLE IMPORTANCE
rf3.2 <- randomForest(Voted_Democrat2~ ., data=crop_county, importance = TRUE)
varImpPlot(rf3.2, type = 1, n.var = 5, main = "")
```




```
imp3.2 <- importance(rf3.2)
print(imp3.2)
```

```
                      Voted_Dem Voted_Repub MeanDecreaseAccuracy
MedianHouseholdIncome  29.31293    25.99659             40.15952
Percent_in_Poverty     21.08452    43.48471             54.19663
```

```
Unemployment_Rate      18.04099    16.42222             24.55055
Percent_College_Edu    57.13341    66.84872             85.00303
State                  81.71309    35.49015             74.44364
                     MeanDecreaseGini
MedianHouseholdIncome         122.7324
Percent_in_Poverty            131.3185
Unemployment_Rate              69.1417
Percent_College_Edu           189.4019
State                         206.8535
```

```
############

#TRYING TO PENALIZE THE SYSTEM FOR MISCLASSIFYING COUNTIES
#THAT VOTED DEMOCRAT, in this new model;
#still makes it worse, somehow.

#########

rf4 <- randomForest(Voted_Democrat2~ .,   data=crop_county,
                    classwt = c("Voted_Dem" = 20, "Voted_Repub" = 1))


rf_preds4 <- predict(rf4, type = "response")

# make confusion matrix
tb4 <- table(actual = crop_county$Voted_Democrat2, predicted = rf_preds4)
addmargins(tb4)
```

```
             predicted
actual        Voted_Dem Voted_Repub  Sum
  Voted_Dem         181         248  429
  Voted_Repub        40        2193 2233
  Sum               221        2441 2662
```

```
# Accuracy (percent correctly classified)
(181+2193)/2662
```

```
[1] 0.8918107
```

```
# Sensitivity (percent of Voted Dems correctly classified)
181/429
```

```
[1] 0.4219114
```

```
# Specificity (percent of Didn't Vote Dem correctly classified)
2193/2233
```

```
[1] 0.9820869
```