Statistical Foundations of Data Analytics

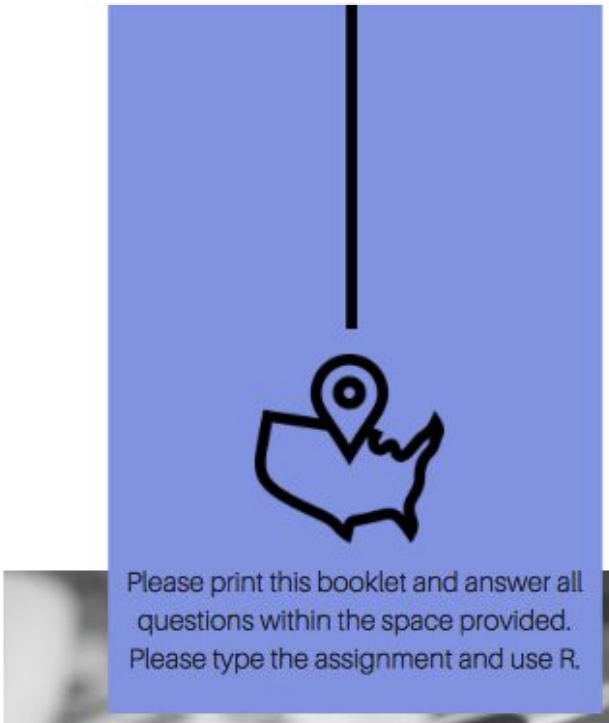
LAB 7: Tree-based methods

Last name: Greene
First name: Samuel
Student ID: 260722742

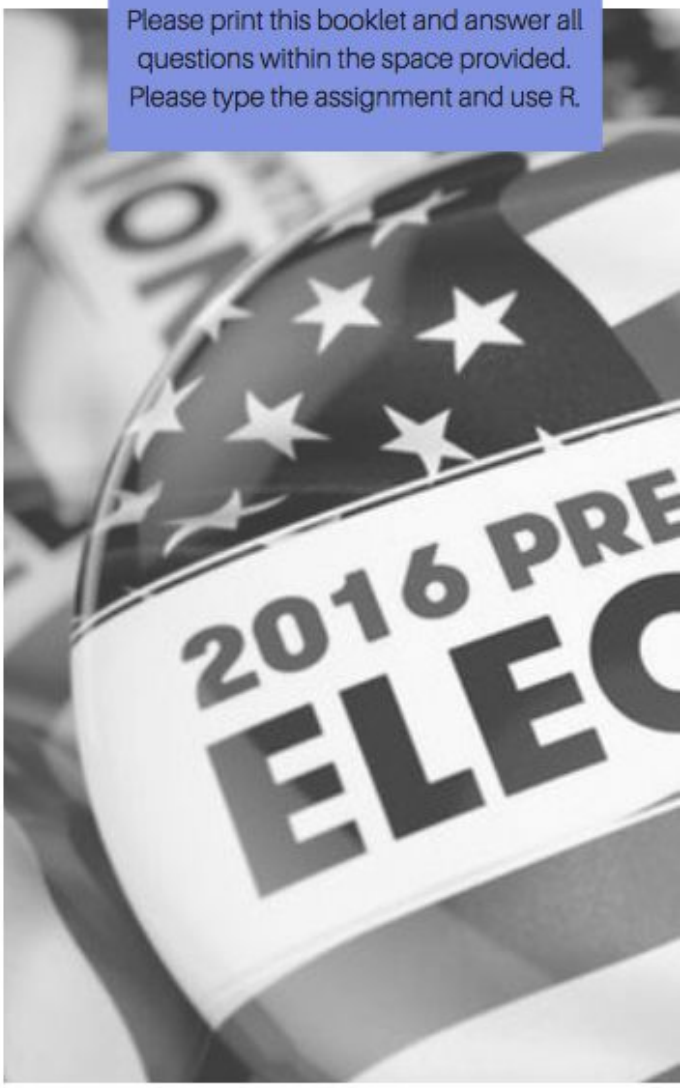
Last name: Tian
First name: Yuan
Student ID: 260727992

Last name: Park
First name: Jamie
Student ID: 260376390

Last name: Toronga
First name: Nicholas
Student ID: 260715831



Please print this booklet and answer all questions within the space provided. Please type the assignment and use R.



U.S. Elections: Who voted for Trump? A classification-trees approach

In 2016, the U.S. had a general presidential election between two main candidates: (i) Donald Trump (from the Republican party); and (ii) Hillary Clinton (from the Democratic party).

Hillary Clinton is a moderate candidate. Donald Trump, in contrast, is a right-wing candidate with drastically more conservative views. Ultimately, Trump won by a small margin, attracting candidates with specific characteristics. I have collected data on all U.S. county districts, with the following information:

Response Variable

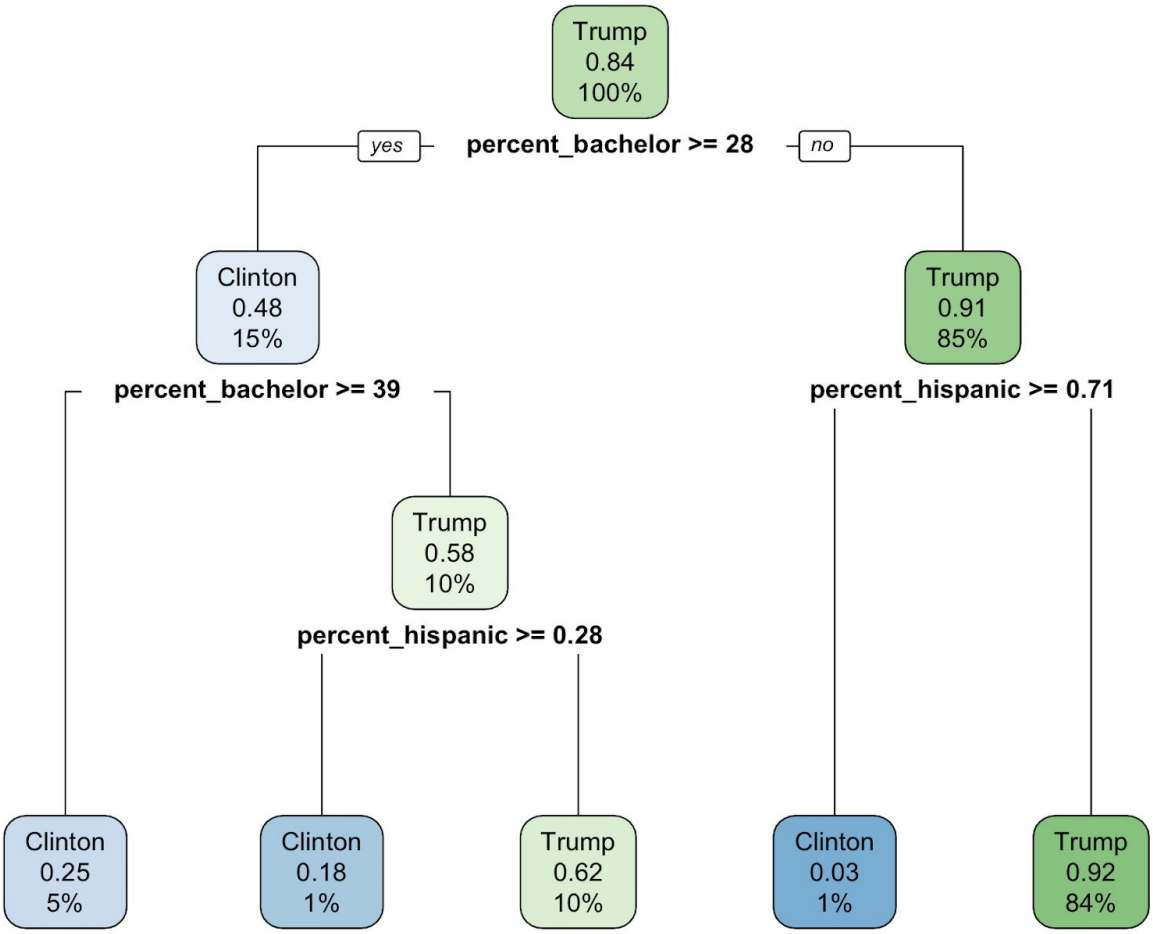
winner: Candidate who obtained more votes in a given county {Clinton, Trump}

Relevant predictors

- **Population:** population, 2014 estimate
- **County_population_2010:** The county's population in the 2010 census
- **County_population_2014:** The county's population in the 2014 census
- **Population_change:** population change in percentage terms between 2010 and 2015
- **percent_under_5:** % of county population under 5 years of age.
- **percent_under_18:** % of county population under 18 years of age.
- **percent_over_65:** % of county population over 65 years of age.
- **percent_female:** % of county female population.
- **percent_foreign:** % of county foreign population.
- **Percent_black: :** % of county black population.
- **Percent_hispanic:** % of county Hispanic population.
- **Median_household_income:** median income (in USD) in a household.
- **Poverty_percent:** % of population under the poverty line
- **Population_density:** population density (habitants per square mile)
- **per_capita_income:** average income per person in the county.
- **percent_bachelor_degree:** % of population (aged over 25) holding a bachelor degree.

1. Interpreting a classification tree (5 points)

Consider the following tree:



A) In this tree, which are the internal and terminal nodes?

Internal nodes:

Percent_bachelor<=28

Percent_hispanic>=0.71

Percent_bachelor>=39

Percent_hispanic>=0.28

Terminal nodes: Do these need to include the %?

Clinton 0.25, 5%

Clinton 0.18, 1%

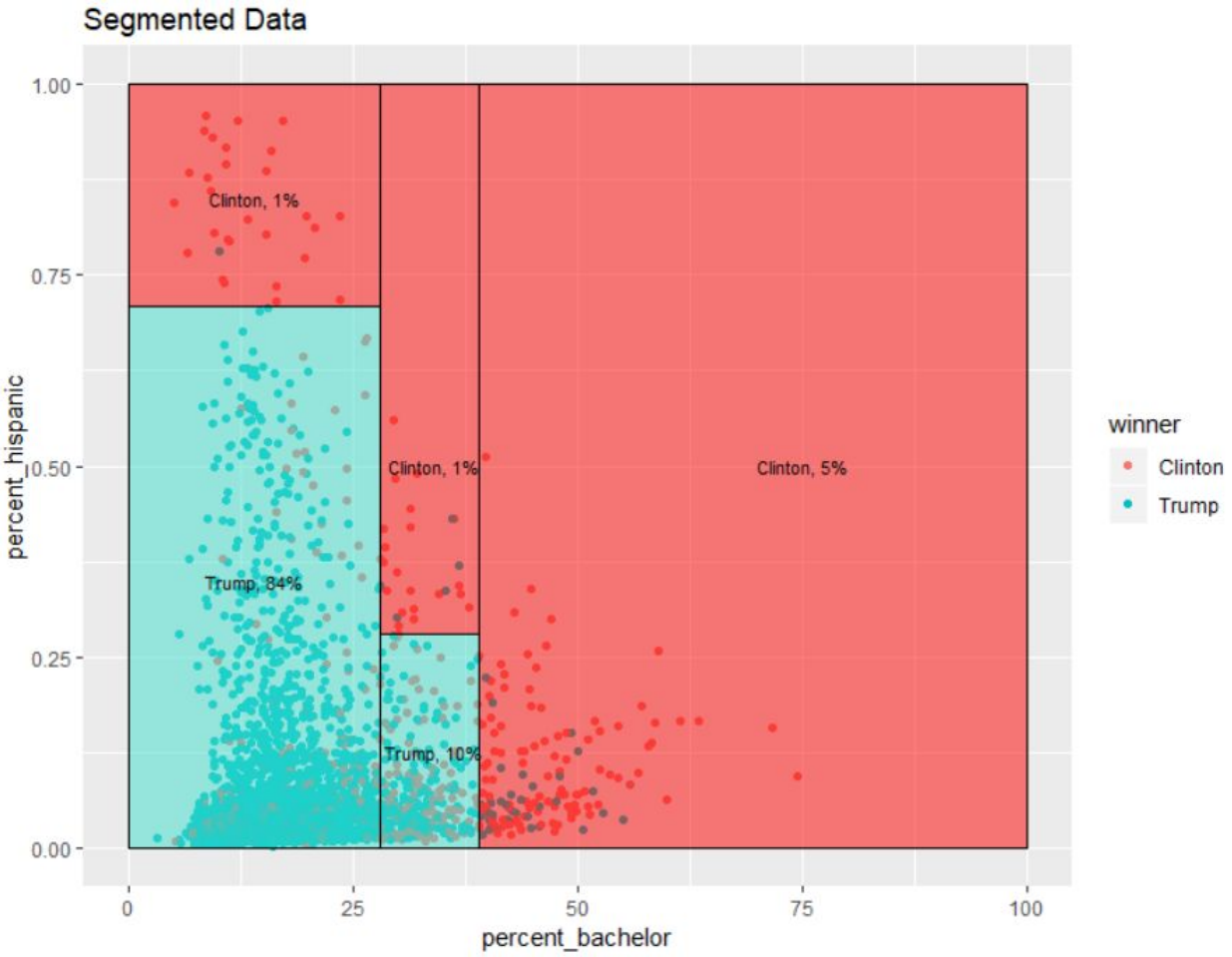
Clinton 0.03, 1%

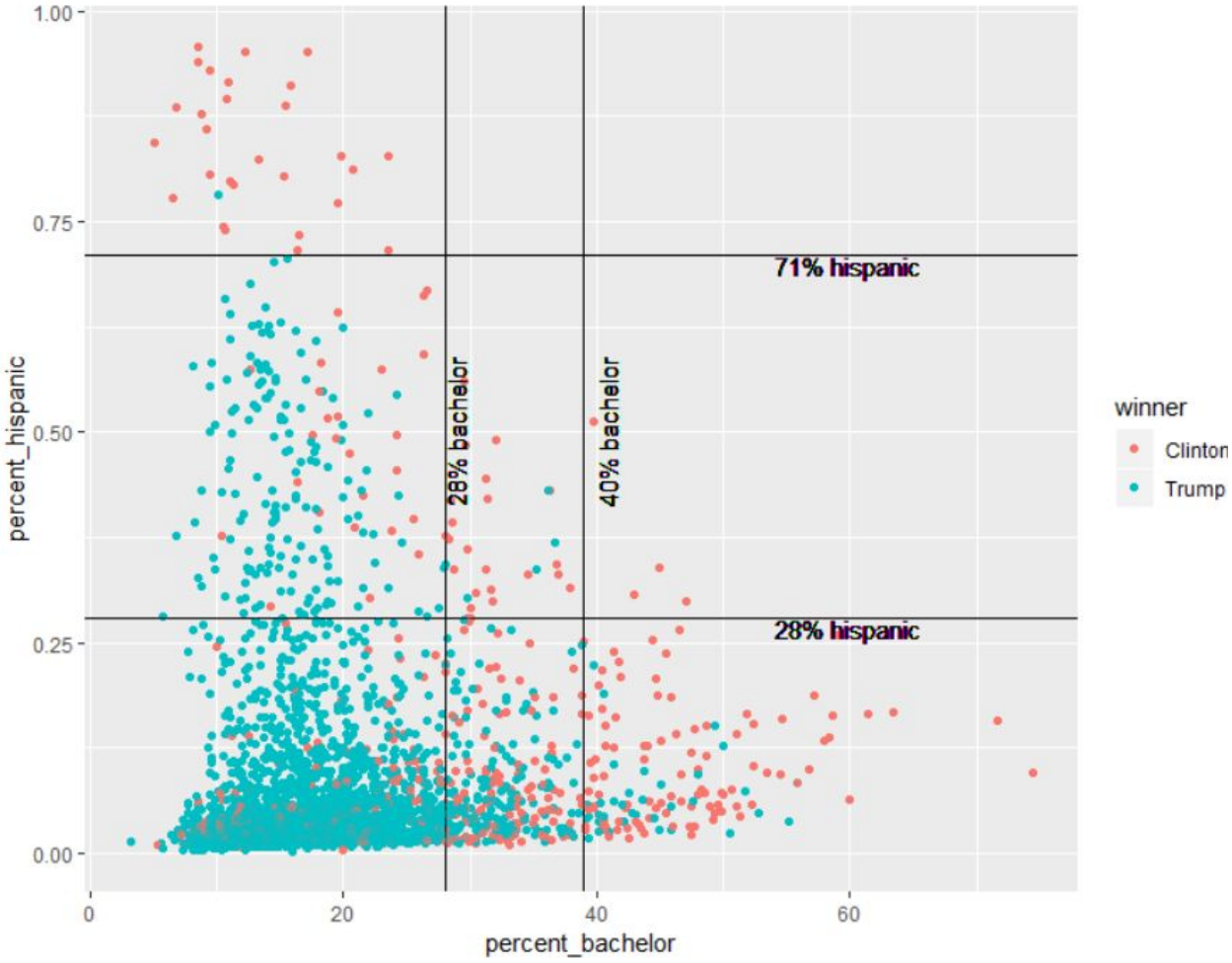
Trump 0.62, 10%

Trump 0.92, 84%

B) Create a scatter plot of percent_bachelor vs. per_hispanic. Display how the regions were split in the tree above.[1]

[1] I am looking to a plot similar to the one I created at the beginning of the slides of Lecture 9 (duration vs. year). You should create it using R-studio

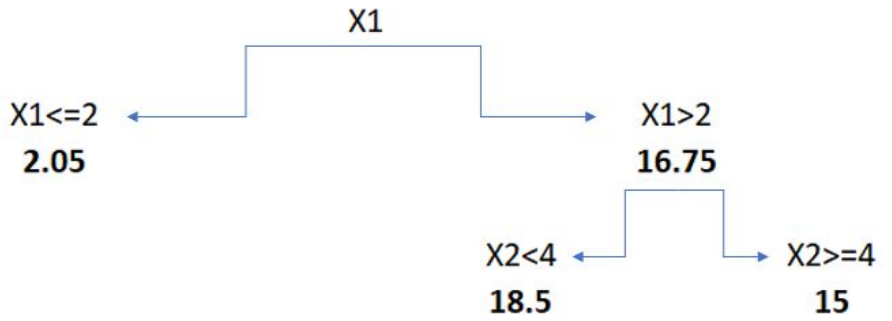

Segmented Data
1.00
0.75
0.50
0.25
0.00
percent_hispanic
Clinton, 1%
Trump, 84%
Clinton, 1%
Trump, 10%
Clinton, 5%
0
25
50
75
100
percent_bachelor
winner
Clinton
Trump



C) Suppose you have the following training data with four observations:

- Obs 1: Y=2.1, X_1=2, X_2=4
- Obs 2: Y=18.5, X_1=7, X_2=3
- Obs 3: Y=2, X_1=1, X_2=2
- Obs 4: Y=15, X_1=8, X_2=6

Find the regression tree with **exactly** three terminal nodes that has the lowest RSS in the training data. Hint: there may be more than one optimal tree. You just need to give me one.



RSS 0.005 = (2.1-2.05)^2+(2-2.05)^2+(18.5-18.5)^2+(15-15)^2

X1 <= 2
(2.1+2)/2

X1 > 2
(18.5+15)/2

X2 < 4
18.5

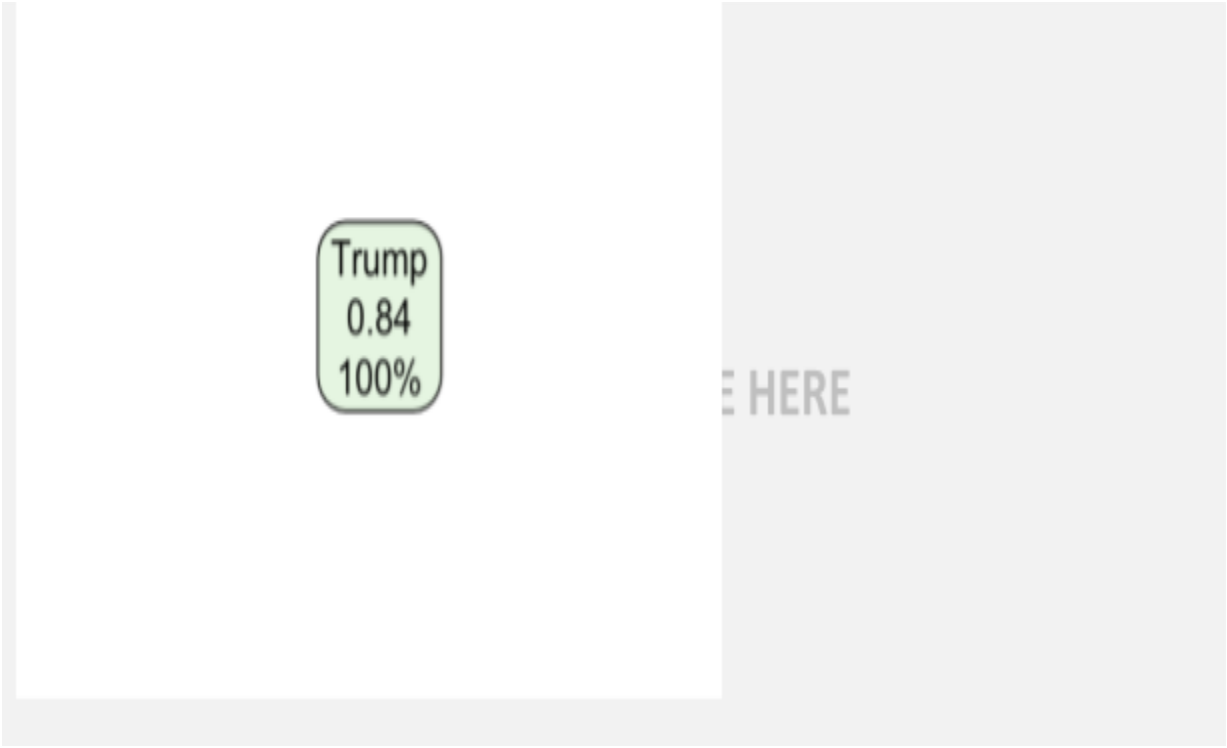
X2 >= 4
15

2. Growing a tree (10 points)

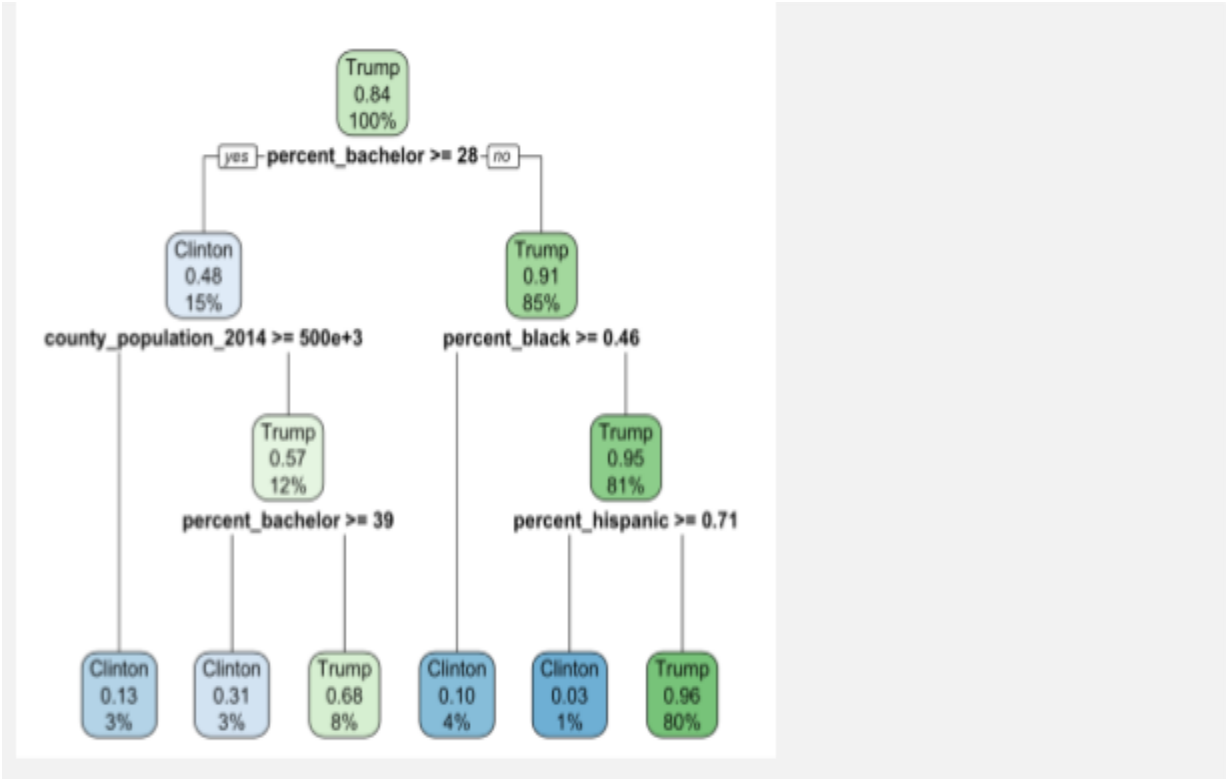
Imagine Trump and Clinton are competing against each other in the next election. We want to predict the county winner across each county. To study this probability, we will look at the following relationship:

Winner=f(county_population_2014, population_change, percent_under_5, percent_over_65, percent_female, percent_black, percent_hispanic, median_household_income, percent_bachelor_degree

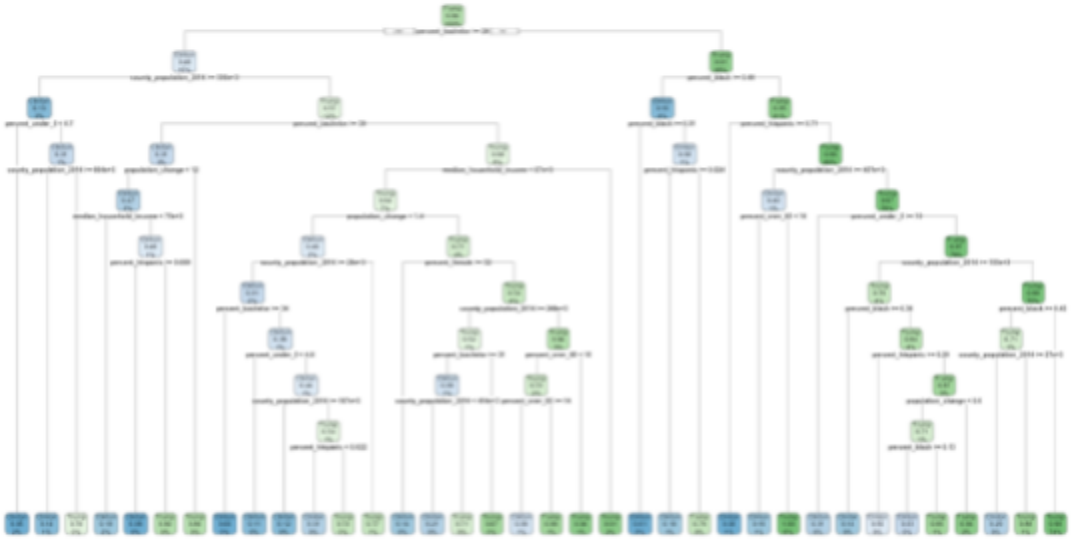
A) Grow a tree using cp=0.2. Paste the tree diagram:



B) Grow a tree using cp=0.05. Paste the tree



C) Grow a tree using cp=0.001. Paste the tree



D) What is the meaning of the “cp”? What are the advantages of having a small value of cp? What are the problems of having a low value of cp?

Your answer:

Cp stands for cut point. It is the threshold at which the tree will stop branching out, making the tree more complex.

The advantages of having a small cp are that it will lead to a more complex and sophisticated tree.

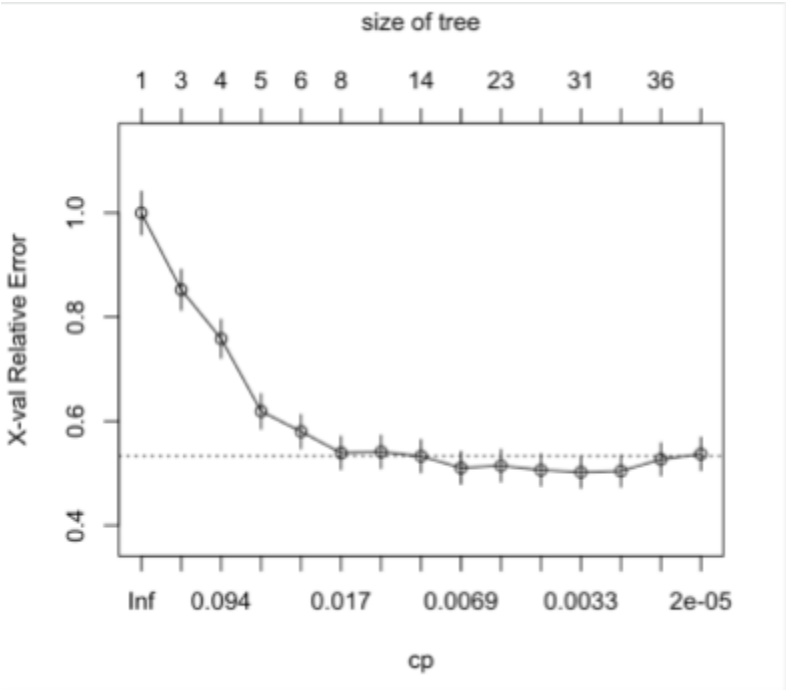
The disadvantages of having a small cp are that it can cause overfitting and poor out of sample performance.

E) Which steps should we follow to find the optimal cp value?

Your answer:

We should start with a very complex tree (one with a low cp, such as 0.00001). We then need to prune (cut down the branches of) this complex tree. In order to prune the tree, we must find the cp value in the complex tree that minimizes error, and gives us the best out of sample performance. We then use that cp value that we found to create a new tree.

F) Plot the out-of-sample error (i.e. the relative x-error) of a tree as a function of the cp value:



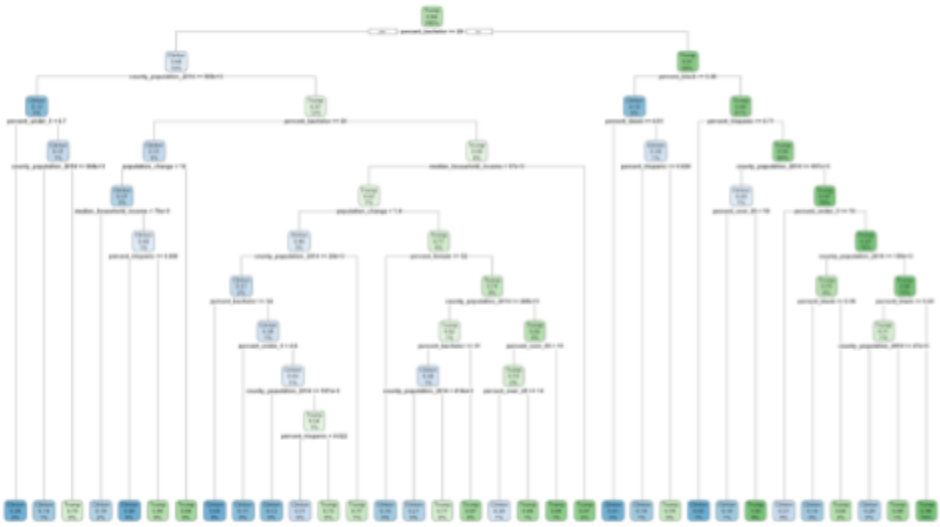
G) Find the cp value with the optimal out-of-sample performance:

```
overfittedtree$cptable[which.min(overfittedtree$cptable[,'xerror']),'CP']
min(overfittedtree$cptable[,'xerror'])
```

```
> overfittedtree$
[1] 0.00307377
```

CP value = 0.00307377

H) Plot the tree diagram with the highest predictive performance.



3. Growing a random forest (10 points)

A) What is a key problem with classification/regression trees?

Your answer:

They suffer from high variance. For example, if we fit two different datasets on a given regression tree, results will vary tremendously.

B) Suppose we have a dataset with five observations (4 predictors):

- **Obs$_1$**=(Y_1,$X_{1,1}$,$X_{2,1}$,$X_{3,1}$, $X_{4,1}$)
- **Obs$_2$**=(Y_2,$X_{1,2}$,$X_{2,2}$,$X_{3,2}$, $X_{4,2}$)
- **Obs$_3$**=(Y_3,$X_{1,3}$,$X_{2,3}$,$X_{3,3}$, $X_{4,3}$)
- **Obs$_4$**=(Y_4,$X_{1,4}$,$X_{2,4}$,$X_{3,4}$, $X_{4,4}$)
- **Obs$_5$**=(Y_5,$X_{1,5}$,$X_{2,5}$,$X_{3,5}$, $X_{4,5}$)

Imagine you are a computer, and I instruct you to give me five bootstrapped samples. Show me an example that illustrates the spirit of bootstrapping:

Bootstrapped sample 1	**Bootstrapped sample 2**	**Bootstrapped sample 3**	**Bootstrapped sample 4**	**Bootstrapped sample 5**

Obs1,1,2,3,4	Obs2,2,4,5,1	Obs3,4,2,1,4	Obs2,5,5,4,3	Obs1,2,3,5,5

C) What is bagging? Roughly speaking, what would bagging do to the above bootstrapped samples? Why is bagging desirable?

Your answer:

Bagging uses the technology called Bootstrapping by selecting multiple sub-samples of size N with replacement. Then, it fits a regression tree on each bootstrapped. Finally, it takes the average of the trees to find the mean response. Bagging is desirable because it will reduce the overfitting and presents a solution to the high variance that regression trees face.

D) What are random forests? How are random forests different from Bagging?

Your answer:

Random Forests provide an improvement over bagged trees. In traditional bagging problems, we would always utilize four predictors to build each tree. In random forests, we use only two predictors in each bootstrapped sample. By randomly using a subset of predictors in each model, random forests eliminate possible biases due to multicollinearity.

E) Suppose I perform a random forest algorithm on the sample from part (B). How many observations and predictors would be selected in each iteration?

Your answer:

There are five observations and two predictors selected in each iteration.

$\sqrt{4}=2$

F) What is the Out-of-bag performance of a random forest? In your samples from part (B), which would be the out-of-bag observations?

OOB observations - sample 1	OOB observations - sample 2	OOB observations - sample 3	OOB observations - sample 4	OOB observations - sample 5
Obs5	Obs3	Obs5	Obs1	Obs4

Perform a random forest algorithm on the following model (grow 10,000 trees):

Winner=f(county_population_2014, population_change, percent_under_5, percent_over_65, percent_female, percent_black, percent_hispanic, median_household_income, percent_bachelor_degree

G) In this model, how many observations did the random forest pick when growing each tree?

Your answer:

The random forest will pick 3112*(⅔)=2075 observations when growing each tree. Because each bootstrapped subsample uses ⅔ rd of the dataset's observations.

H) In the above model, trace the out-of-bag error rate of the model and paste the code below:

Your answer:

thewinner=randomForest(winner ~ county_population_2014+population_change+percent_under_5+percent_over_65+percent_female+percent_black+percent_hispanic+median_household_income+percent_bachelor,ntree=10000,do.trace=50,importance=TRUE)

I) What was the error rate achieved in this model? What does this mean?

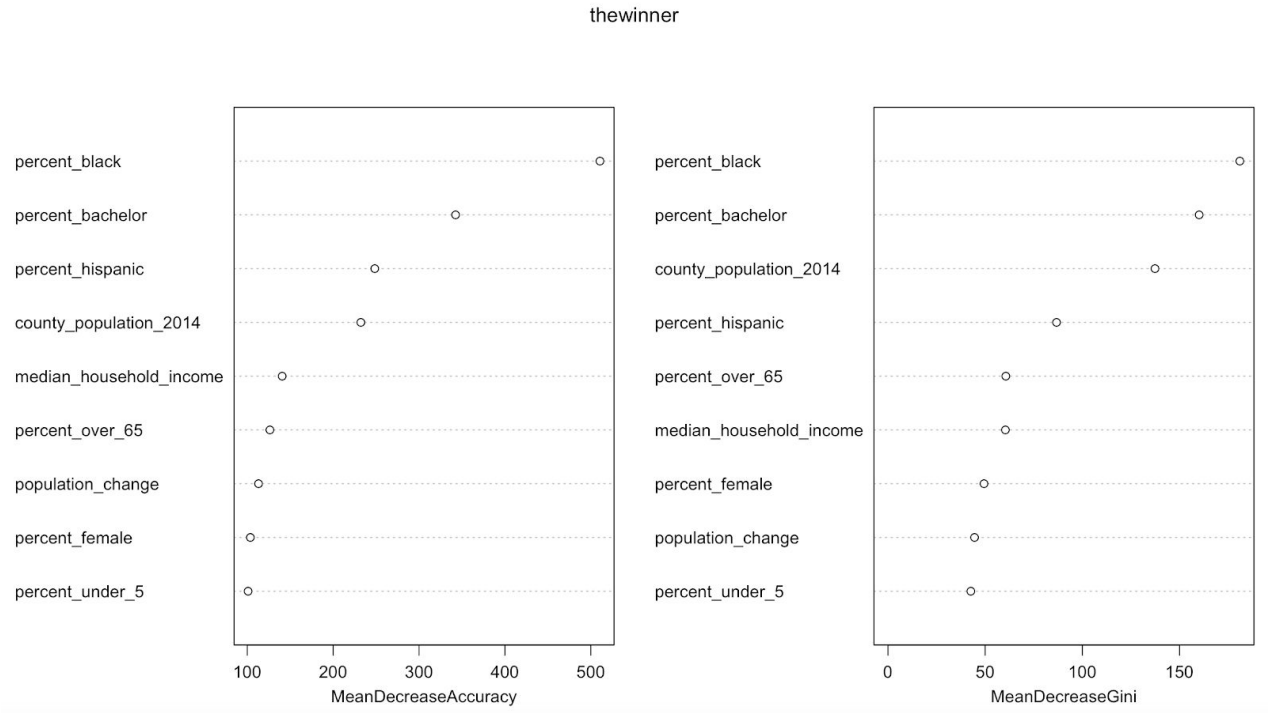
Your answer:

```
Call:
 randomForest(formula = winner ~ county_population_2014 + population_change +
percent_under_5 + percent_over_65 + percent_female + percent_black +      percent
_hispanic + median_household_income + percent_bachelor,      ntree = 10000, impor
tance = TRUE)
               Type of random forest: classification
                     Number of trees: 10000
No. of variables tried at each split: 3

        OOB estimate of  error rate: 6.59%
Confusion matrix:
        Clinton Trump class.error
Clinton     353   135  0.27663934
Trump        70  2554  0.02667683
```

The error rate achieved in this model is from 6.52% to 6.72% from the test of out-of-bag error rate. This means around 6.59% of the classifications are predicted wrong.

J) Create a variable importance plot and paste it below:

Your answer:



K) Suppose that a political strategist hires you to give her an brief summary about the counties where the Democrats (Hillary Clinton's Party) are expected to do better in the next election (assuming that the demographics don't change. Provide her with a quick summary of your findings. **No jargon.**

Your answer:

For democrats, the test shows the percentage of black people is the most important factor in determining the result of the election. In an order, the percentage of people who get bachelor degree, the county population in 2014 and the percentage of hispanic are among the most significant factors. The higher these percentages are, the higher chance Clinton will win in the next election. Afterwards, the percentages of citizens under five years and over 25 years old are also very significant. However, the population change during these years and the median household income shows little significance in determining the result of next election.

4. Representing a random forest visually (5 points)

Random forests have immense predictive power. But they have a practical drawback: They can't be easily represented visually, like a simple tree, or a regression line. This means that you need to find a way to represent your predictions.

In this case, the natural way to represent the findings of your classification trees is to draw a map that illustrates your predictions. In this question, we will learn how to geo-represent your random forests predictions and, additionally, learn how to map geographic information using R.

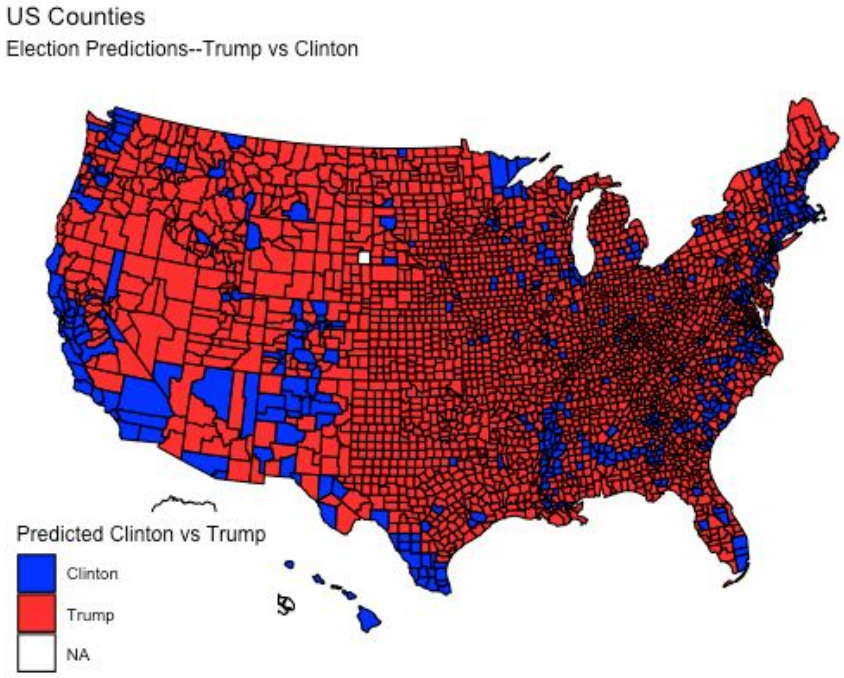
So, suppose you're working in a political agency, or in a newspaper. Your boss just told you: "I need you to give me a map that visualizes your predictions." To do this, you will need to follow the instructions below:

A) Create a variable called "Predicted_winner" in the dataset election_2016 that includes the predicted winners, based on your 10,000-trees random forest estimation. Paste your code below to create this variable:

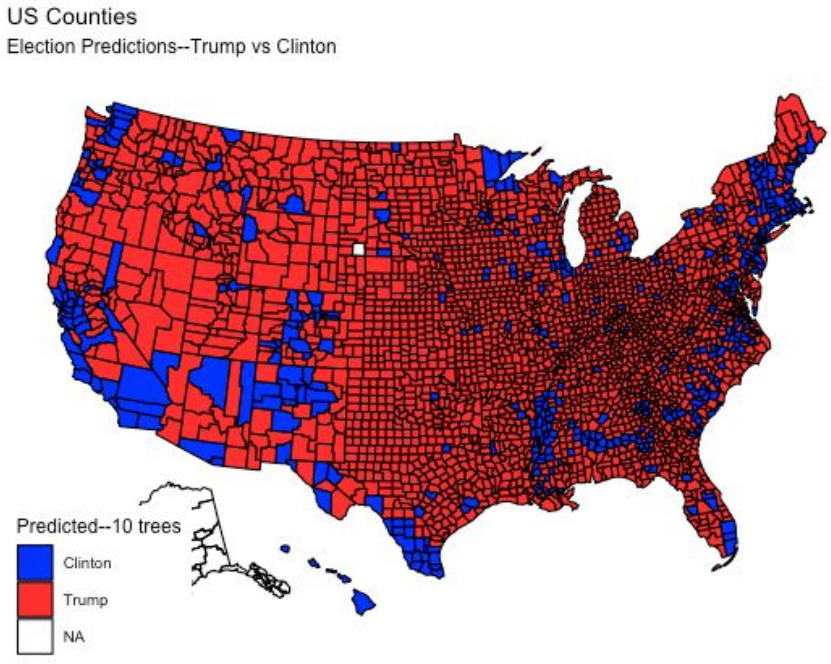
```
el = read.csv("election_2016.csv")
#removing null values for RandomForest to work
election = na.omit(el)
sum(is.na(election))
attach(election)
rf_model = randomForest(winner~county_population_2010 +county_population_2014
                        + population_change + percent_under_5 + percent_under_18
                        + percent_over_65 + percent_female + percent_black+ percent_hispanic
                        + median_household_income + poverty_percentage + percent_bachelor,
                        ntree = 10000, data = election, importance = TRUE)
election$prediction = predict(rf_model, election)
election$prediction = as.factor(election$prediction)
detach(election)
attach(election)
table(prediction)

install.packages("ggplot2")
```

B) Install the packages "ggplot2" and "usmap." After installing them, create a map, at the county level, that shows the predicted winner in each county. If Clinton is predicted to wins, the county should be blue shaded; if trump is predicted to win, the county should be red-shaded. You'll need to figure out on your own how to create this map (it shouldn't be that hard). To make it easier for you, you'll need to follow this guide:

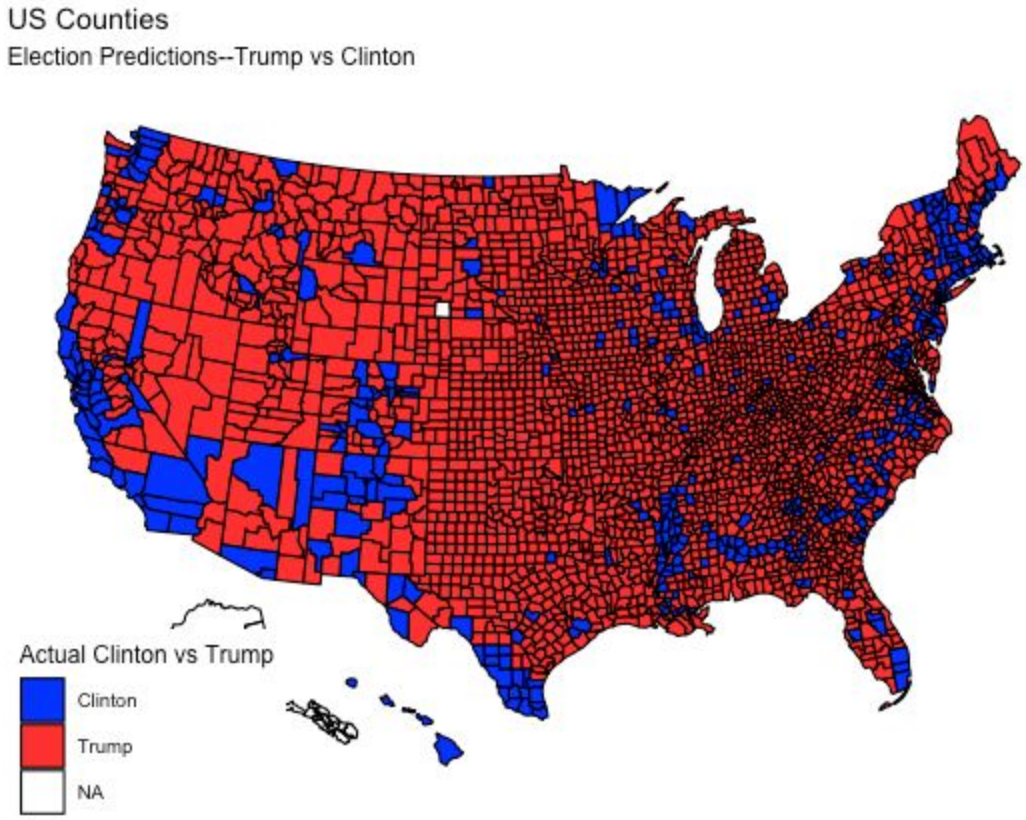
https://cran.r-project.org/web/packages/usmap/vignettes/mapping.html



C) Re-run your random forest estimation, but this time using a 10-tree random forest to predict the winners.



D) Create a new map, this time having the actual winner of the election (i.e., using the variable winner)



E) What differences do you notice in the two random forests predictions (compared to the actual winner map). What can you say about random forests after looking at the above three maps?

Your answer:

Random Forest is a very powerful predictive technique. The model with 10,000 trees is way more powerful than the model with only 10 trees. The model with many trees accurately predicted the winning candidate; however, the model with 10 trees incorrectly classified some states as Democrats when they were Republicans and vice-versa. In numbers, predicted using the model with 10 trees had 8 less democratic states than the actual statistics showed. The actual and the predicted with 10,000 trees model is pretty much similar. The 10 trees model has some incorrect classifications.

Note: This question is designed to get you to independently learn new tasks and packages in R, which will be vital during your jobs as data analysts.

Code

#1

```
attach(election_2016)

library(ggplot2)

ggplot(election_2016,
aes(percent_bachelor,percent_hispanic,colour=winner))+geom_point()+geom_vline(xintercept
=39, show.legend = "bachelor")+
  geom_vline(xintercept=28)+geom_hline(yintercept=0.28)+geom_hline(yintercept=0.71)+
  geom_point(alpha=0.1)+
  geom_text(aes(x=28, label="28% bachelor", y=0.5), colour="black", angle=90, vjust = 1)+
  geom_text(aes(x=40, label="40% bachelor", y=0.5), colour="black", angle=90, vjust = 1)+
  geom_text(aes(y=0.71, label="71% hispanic", x=60), colour="black", angle=0, vjust = 1)+
  geom_text(aes(y=0.28, label="28% hispanic", x=60), colour="black", angle=0, vjust = 1)


ggplot(election_2016, aes(percent_bachelor,percent_hispanic,colour=winner))+geom_point()+
  geom_point(alpha=0.1)+
  annotate("rect", xmin=39, xmax=100, ymin=0, ymax= 1,
                                          fill="red", colour="black", alpha = 0.5) +
  annotate("text", x=75, y=0.5, label="Clinton, 5%", size=3)+
  annotate("rect", xmin=28, xmax=39, ymin=0.28, ymax= 1,
                                          fill="red", colour="black", alpha = 0.5) +
  annotate("text", x=34, y=0.5, label="Clinton, 1%", size=3)+
  annotate("rect", xmin=0, xmax=28, ymin=0.71, ymax= 1,
                                          fill="red", colour="black", alpha = 0.5) +
  annotate("text", x=14, y=0.85, label="Clinton, 1%", size=3)+
  annotate("rect", xmin=0, xmax=28, ymin=0, ymax= 0.71,
                                          fill="turquoise", colour="black", alpha = 0.5) +
  annotate("text", x=14, y=0.35, label="Trump, 84%", size=3)+
  annotate("rect", xmin=28, xmax=39, ymin=0, ymax= 0.28,
```

```
                                        fill="turquoise", colour="black", alpha = 0.5) +
  annotate("text", x=34, y=0.125, label="Trump, 10%", size=3)+ggtitle("Segmented Data")
observation = c(2.1,18.5,2,15)
predictor_1 = c(2,7, 1, 8)
predictor_2 = c(4, 3, 2, 6)
q1c = data.frame(observation, predictor_1, predictor_2)
View(q1c)
attach(q1c)
tree = rpart(observation~predictor_1+predictor_2, control =
rpart.control(cp=0.0000000000001))
rpart.plot(tree)
summary(tree)
tree$cptable[which.min(tree$cptable[,"xerror"]),"CP"]
newtree = rpart(observation~predictor_1+predictor_2, control = rpart.control(cp=1e-13))
rpart.plot(newtree)
thewinner=randomForest(winner ~
county_population_2014+population_change+percent_under_5+percent_over_65+percent_fe
male+percent_black+percent_hispanic+median_household_income+percent_bachelor,ntree=1
0000,importance=TRUE,do.trace=50)
varImpPlot(thewinner)
importance(thewinner)
Question 4
el = read.csv("election_2016.csv")
#removing null values for RandomForest to work
election = na.omit(el)
sum(is.na(election))
attach(election)

#first model using 10000 trees
rf_model = randomForest(winner~county_population_2010 +county_population_2014
```

```
                + population_change + percent_under_5 + percent_under_18
                + percent_over_65 + percent_female + percent_black+ percent_hispanic
                + median_household_income + poverty_percentage + percent_bachelor,
                ntree = 10000, data = election, importance = TRUE)

#second model using 10 trees

new_model = randomForest(winner~county_population_2010 +county_population_2014
                 + population_change + percent_under_5 + percent_under_18
                 + percent_over_65 + percent_female + percent_black+ percent_hispanic
                 + median_household_income + poverty_percentage + percent_bachelor,
                 ntree = 10, data = election, importance = TRUE)
#prediction column using the random forest model with 10 000 trees

election$prediction = predict(rf_model, election)
# prediction using the model with 10 trees

election$new_pred = predict(new_model, election)

election$prediction = as.factor(election$prediction)
election$new_pred = as.factor(election$new_pred)
detach(election)
attach(election)
table(prediction)
table(new_pred)

install.packages("ggplot2")
library(ggplot2)
install.packages("usmap")
```

```
library(ggplot2)

library(usmap)


# scale_colour_manual(values = election$prediction, aesthetics = c("blue", ""))


#value = ifelse(election$prediction == "Trump", "red", "blue")

plot_usmap(data = election, values = "prediction", lines = "black", label_color = "blue" ) +
guides(fill=guide_legend(title="Predicted Clinton vs Trump"))+  labs(title = "US Counties",
subtitle = "Election Predictions--Trump vs Clinton") + scale_fill_manual(values =
c("blue","firebrick1"))

?plot_usmap()



plot_usmap(data = election, values = "new_pred", lines = "black", label_color = "blue" ) +
guides(fill=guide_legend(title="Predicted--10 trees"))+  labs(title = "US Counties", subtitle =
"Election Predictions--Trump vs Clinton") + scale_fill_manual(values = c("blue","firebrick1"))


plot_usmap(data = election, values = "winner", lines = "black", label_color = "blue" ) +
guides(fill=guide_legend(title="Actual Clinton vs Trump"))+  labs(title = "US Counties", subtitle =
"Election Predictions--Trump vs Clinton") + scale_fill_manual(values = c("blue","firebrick1"))


election$winner = as.factor(election$winner)

table(winner)

table(new_pred)

table(prediction)
```