Election

Project done by Chase Humiston,Judiel Salandanan, Jingyu Fang, Rick Chen, Yizhuang Kang

Step 1. Data merging and cleaning.

```
##
## Perl XLSX support libraries successfully installed.
```

Step 5. Final Report

1. Introduction

The goal of this project was to analyze past election results data and attempt to draw meaningful conclusions and correlations from exploring the aforementioned data.

Our data came from six distinct sources with four sources accounting for each of the vote results for a given election year (2004, 2008, 2012, 2016), 2010 census data for all counties, and latitude and longitude values for every county as well. After extracting the data from all the sources, they were merged into a larger dataframe using R. This activity presented several challenges such as the absence of data for some years. For example, the 2004 election year lacked data for the state of Virginia, among others. Further, the merging of the data occured on (state, countyname) pairs which presented challenges that involved standardizing disparate uses names between different counties.

Exploratory data analysis was performed on the data in order to find interesting relationships across years and based on the counties 2010 census attributes. The plots and exact relationships uncovered are elaborted at length below. In particular, an informative map of the recent election results is included.

In an attempt to use the gathered data in a predictive sense, our team created two predictors using two separate methods: recursive partitioning and k-nearest neighbors. The recursive partitioning predictor was used for the 2016 election results and the k-nn predictor was trained on the 2012 results and tested on the 2016 election results. A more elaborate comparison of where the particulars did well and how they compared to each other follows.

Note: Since we used “installXLSXsupport()” to read xlsx file. Your computer need to have java and perl to fully see our data frame

2. Data Description

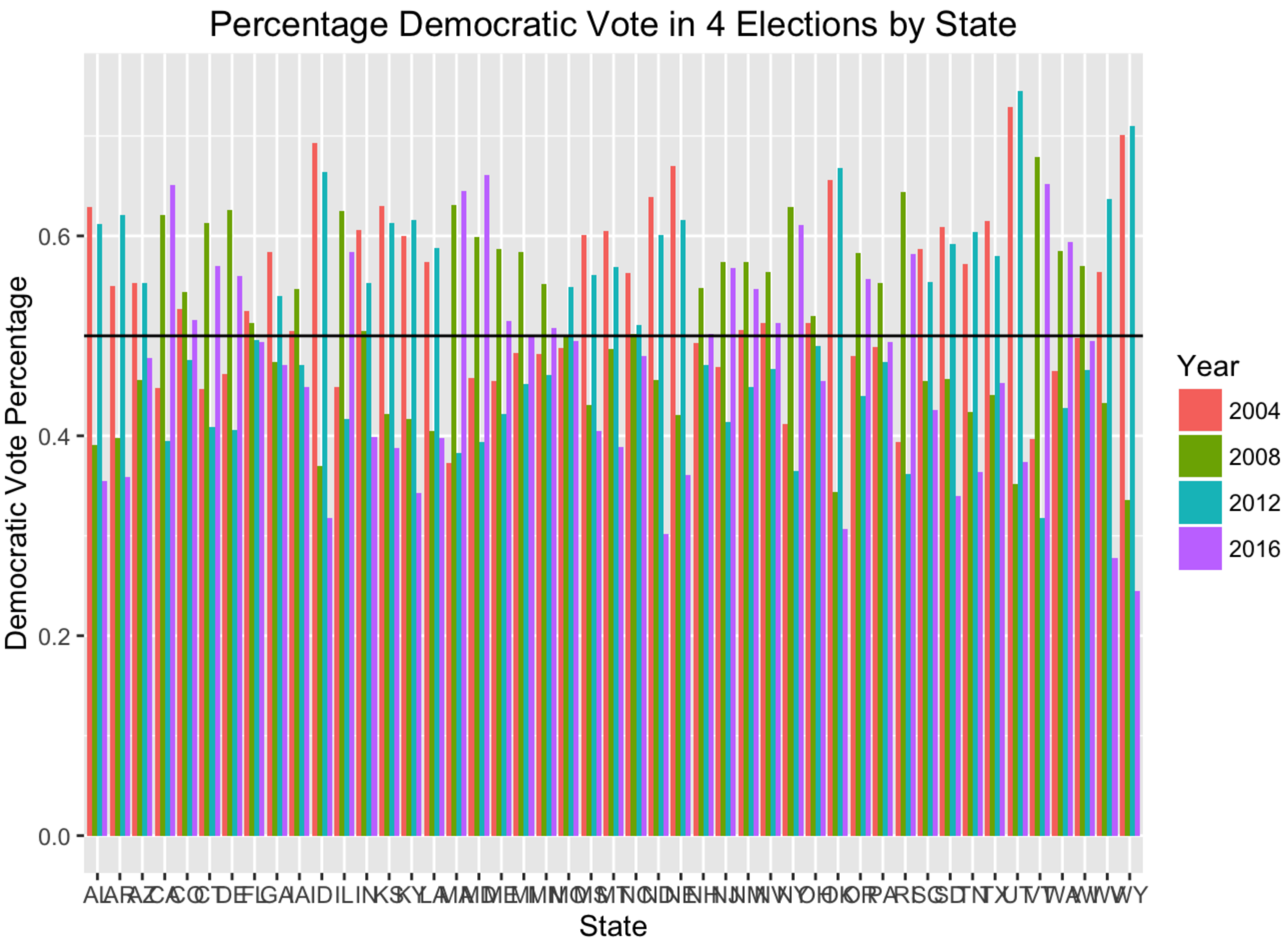
The first plot that was created to explore the final set of data was a plot of the percentage of Democratic votes for each state for each given election year. From the plot, we can assume that the party had the majority votes(more than 50%) won the state. In 2004, Democrat wins 27 states, in 2008 and 2012 Democrat won 22 states, and in 2016, Democrat won 16 states. Based on history, the 2004 and 2016 elected president was both Republican. However, Democrat won the 2004 votes state-wise but lost in 2016. Therefore, our assumption is not correct all the time.

This emphasizes the importance of winning the necessary amount of electoral votes through winning the appropriate counties. The reason our team found this data and finding interesting in terms of the elections of the past 4 years is because we believe it highlights a rhetoric often discussed about the electoral system in America. That is, does the electoral system in America underrepresent the governing bodies of the state in favor of winning incredibly populous counties. The concern is the diminishment of the value of less populous state opinions, which are often mostly cohesive. After all, it can be seen from the plot that the presidency can be won without a majority of the states such as in 2004, 2008, and 2012.

Note that Virginia, DC, and Hawaii were removed in the creation of this plot because of a lack of 2004 election data.

```
require(ggplot2)

# democrate votes change in 4 elections
plot1dataframe= bigDF[,seq(1:10)]
names(plot1dataframe)=c("A","B", "C", "D","E","F","G","H","I","J")

total.de.2016 = aggregate(C~A, plot1dataframe, sum)
total.re.2016 = aggregate(D~A, plot1dataframe, sum)
total.de.2012 = aggregate(E~A, plot1dataframe, sum)
total.re.2012 = aggregate(F~A, plot1dataframe, sum)
total.de.2008 = aggregate(G~A, plot1dataframe, sum)
total.re.2008 = aggregate(H~A, plot1dataframe, sum)
total.de.2004 = aggregate(I~A, plot1dataframe, sum)
total.re.2004 = aggregate(J~A, plot1dataframe, sum)



total.de.2016 = total.de.2016[!total.de.2016$A %in% c("DC", "HI", "VA"),]
total.re.2016 = total.re.2016[!total.re.2016$A %in% c("DC", "HI", "VA"),]
total.de.2012 = total.de.2012[!total.de.2012$A %in% c("DC", "HI", "VA"),]
total.re.2012 = total.re.2012[!total.re.2012$A %in% c("DC", "HI", "VA"),]
total.de.2008 = total.de.2008[!total.de.2008$A %in% c("DC", "HI", "VA"),]
total.re.2008 = total.re.2008[!total.re.2008$A %in% c("DC", "HI", "VA"),]
total.de.2004 = total.de.2004[!total.de.2004$A %in% c("DC", "HI", "VA"),]
total.re.2004 = total.re.2004[!total.re.2004$A %in% c("DC", "HI", "VA"),]

countsplot1= data.frame(total.de.2016,total.de.2012,total.de.2008,total.de.2004,total.re.2016,
total.re.2012,total.re.2008,total.re.2004)
countsplot1=countsplot1[-seq(from=3, to=16,by=2)]
names(countsplot1)=c("State", "2016 dem Votes", "2012 dem Votes", "2008 dem Votes", "2004 dem
Votes","2016 rep Votes", "2012 rep Votes", "2008 rep Votes", "2004 rep Votes")
countsplot1$'2016 total' = countsplot1$`2016 dem Votes` + countsplot1$`2016 rep Votes`
countsplot1$'2012 total' = countsplot1$`2012 dem Votes` + countsplot1$`2012 rep Votes`
countsplot1$'2008 total' = countsplot1$`2008 dem Votes` + countsplot1$`2008 rep Votes`
countsplot1$'2004 total' = countsplot1$`2004 dem Votes` + countsplot1$`2004 rep Votes`

step1 = cbind(countsplot1[,c(1,2,10)],data.frame(rep(2016, nrow(countsplot1))))
step2 = cbind(countsplot1[,c(1,3,11)],data.frame(rep(2012, nrow(countsplot1))))
step3 = cbind(countsplot1[,c(1,4,12)],data.frame(rep(2008, nrow(countsplot1))))
step4 = cbind(countsplot1[,c(1,5,13)],data.frame(rep(2004, nrow(countsplot1))))
demTotal1 = cbind(step1[,1], data.frame(step1[,2]/step1[,3]),step1[,4])
demTotal2 = cbind(step2[,1], data.frame(step2[,2]/step2[,3]),step2[,4])
demTotal3 = cbind(step3[,1], data.frame(step3[,2]/step3[,3]),step3[,4])
demTotal4 = cbind(step4[,1], data.frame(step4[,2]/step4[,3]),step4[,4])
names(demTotal1) = c("State", "PerDem", "Year")
names(demTotal2) = c("State", "PerDem", "Year")
names(demTotal3) = c("State", "PerDem", "Year")
names(demTotal4) = c("State", "PerDem", "Year")
newDF = rbind(demTotal1,demTotal2,demTotal3,demTotal4)
plot1=ggplot(data=newDF, aes(x=State, y = PerDem, fill=factor(Year))) + geom_bar(stat="identit
y", position="dodge")+geom_hline(aes(yintercept=0.5)) + guides(fill=guide_legend(title="Year")
) + labs(title="Percentage Democratic Vote in 4 Elections by State", y = "Democratic Vote Perc
```

```
entage")
plot1
```



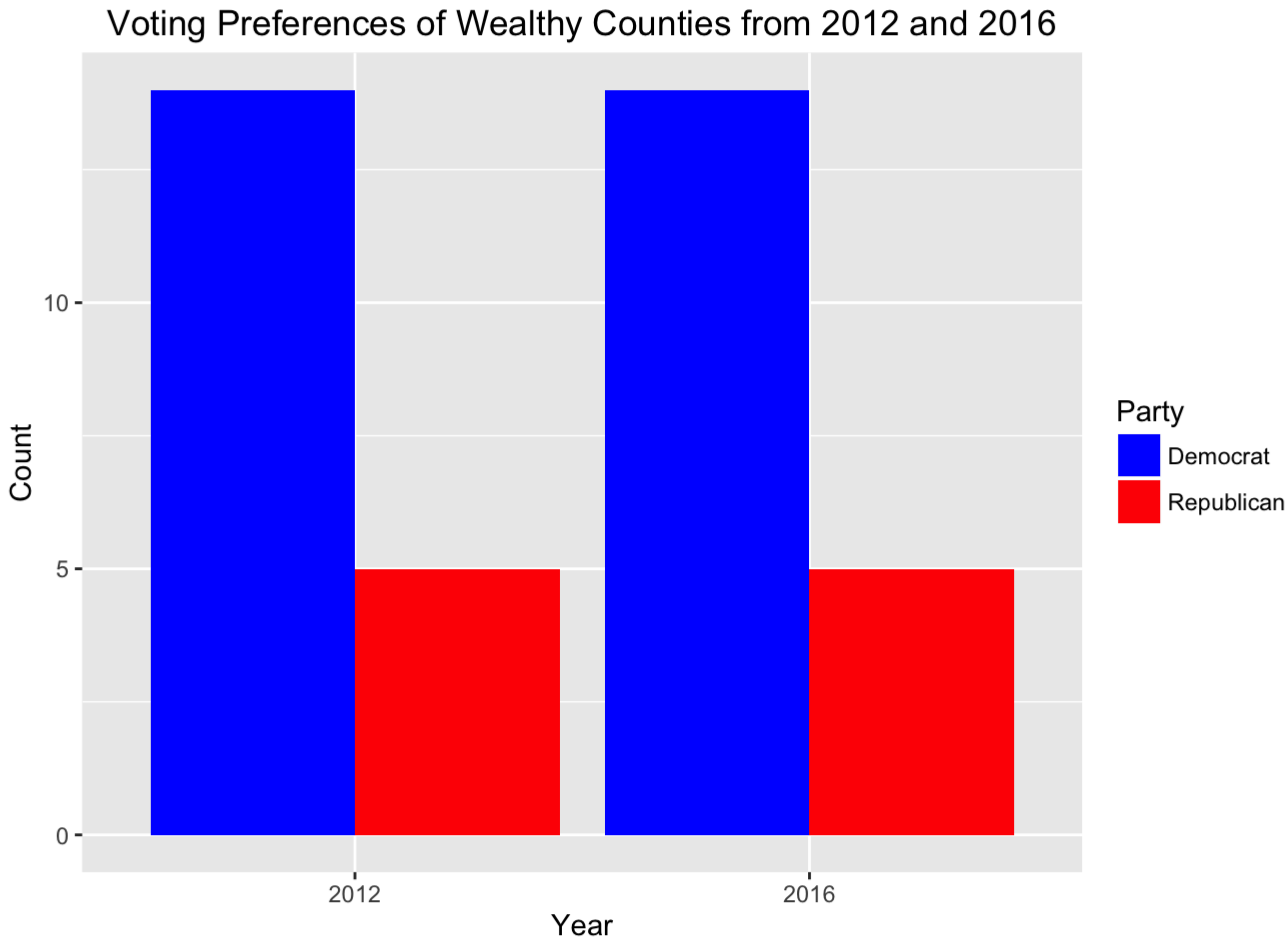
Our group wanted to see how wealthy is going to affect voting preferences. So we examed the top 20 counties wealthiest counties by our metric. To calculate the wealth level for a given county, we leveraged the 2010 census data. The 2010 census data includes population counts for people in various income brackets. The wealth number for a county was calculated taking the total percentage of residents occupying the lowest income bracket on the census (less than $10,000) subtracted by the total percantage of residents occupying the highest income bracket on the census (greater than $200,000). If the number for a county was negative, it indicated a predominantly wealthy county. The converse of the prior statement is also true.

After finding the top 20 wealthiest counties by this metric, we analyzed the vote numbers for these counties for the election years 2012 and 2016 to see how they voted. Our results were that 15 of the counties voted Democrat and 5 voted Republican. Still, the results are mostly obersvational and merely describe a correlation. It could be the fact that most of the states of the counties represent often vote in such a direction. That is, it is unsurprising that the California counties included in the results voted Democrat or that the Texas counties voted Republican, regardless of their score on our team’s wealth metric.

In fact, perhaps the most interesting result stems from observing the dataframe itself. When analyzing the top 20 wealthy counties that cast a vote separate from the way their state voted in the elections, we observe that more counties cast a vote for the Republican candidate, particularly in the counties of New Jersey. This may indicate that despite the results of the plot below, wealthy counties may actually correlate towards voting Republican even in predominantly Democratic states.

```
# Voting Preferences if Wealthy Counties from 2012 and 2016
```

```
incomeDF2012= bigDF[,c("State","County","HC01_VC74","HC01_VC75","HC01_VC84.y","2016 Trump Vote
s","2016 Clinton Votes","2012 Republican Votes", "2012 Democrat Votes")]
incomeDF2012=incomeDF2012[c(-2878,-2875,-2876),]
names(incomeDF2012)=c("State","County","total","less","more","rep2016","dem2016", "rep2012", "
dem2012")
incomeDF2012$less= incomeDF2012$less/ incomeDF2012$total
incomeDF2012$more= incomeDF2012$more/ incomeDF2012$total
incomeDF2012$diff= incomeDF2012$less - incomeDF2012$more
incomeDF2012 = incomeDF2012[c(order(incomeDF2012$diff)),]
refinedDF2012= head(incomeDF2012,n=28)

refinedDF2012 = refinedDF2012[refinedDF2012$County!='fairfax',]
refinedDF2012 = refinedDF2012[c(-1),]

refinedDF2012$voted2012 = refinedDF2012$rep2012 > refinedDF2012$dem2012
refinedDF2012$voted2012[refinedDF2012$voted2012 == FALSE] = 'Democrat'
refinedDF2012$voted2012[refinedDF2012$voted2012 == TRUE] = 'Republican'
refinedDF2012$voted2016 = refinedDF2012$rep2016 > refinedDF2012$dem2016
refinedDF2012$voted2016[refinedDF2012$voted2016 == FALSE] = 'Democrat'
refinedDF2012$voted2016[refinedDF2012$voted2016 == TRUE] = 'Republican'


plot2DF = data.frame(as.factor(c(2016,2016,2012,2012)), c(length(grep("Democrat",refinedDF2012
$voted2016)),length(grep("Republican",refinedDF2012$voted2016)),length(grep("Democrat",refined
DF2012$voted2012)),length(grep("Republican",refinedDF2012$voted2012))),c('Democrat','Republica
n','Democrat','Republican'))
names(plot2DF) = c("Year", "Count", "Party")
plot2=ggplot(data=plot2DF, aes(x=Year, y=Count, fill=Party)) + geom_bar(stat="identity", posit
ion="dodge")+ggtitle("Voting Preferences of Wealthy Counties from 2012 and 2016")+scale_fill_m
anual(values = c(Democrat="blue",Republican="red"))
plot2
```



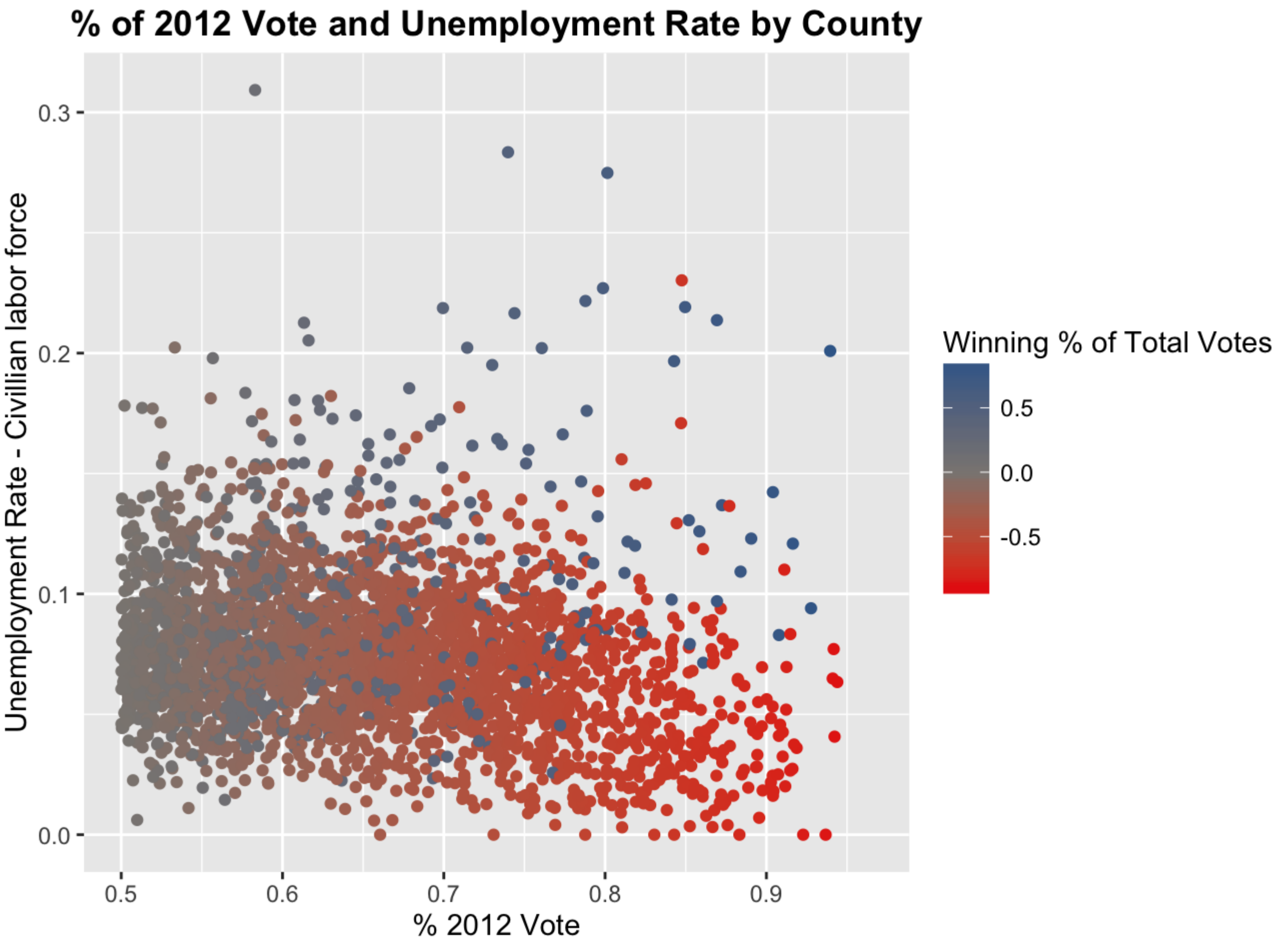
In plot 3, our team tried to find the relationship between percentage of 2012 votes and percentage of unemployment rate by county. We first make a data frame “employmentDF” by using our primary dataframe bigDF by picking up employment status “HC01_VC06, HC01_VC07, HC01_VC08”, “2016 Trump Votes”,“2016 Clinton Votes”,“2012 Republican Votes” and “2012 Democrat Votes” to implement our data analysis.

Right next, our team added three elements to the data frame “unem.rate.2012, dem.per.2012 and rep.per.2012”, which stand for unemployment rate(use formula unemployed population/ total population), percentage of democrats voters and percentage of republican voters for year 2016 and 2012. “winner.per.2012” element in the dataframe stands for the winner of election year 2012. Also, we basically follow the same step for election year 2016. We decided to make a scatter plots because each jitter point with different color will better represent votes of each county of interest. We decide to compare scatter plots of 2016 and 2012 to draw a conclusion. In each graph, every county is a point, x is percentage of democrats or republican votes of a specific county, y is unemployment rate of the county, grouping is color by “total.per.2012”, which is the percentage of democrats votes – percentage of republican votes.

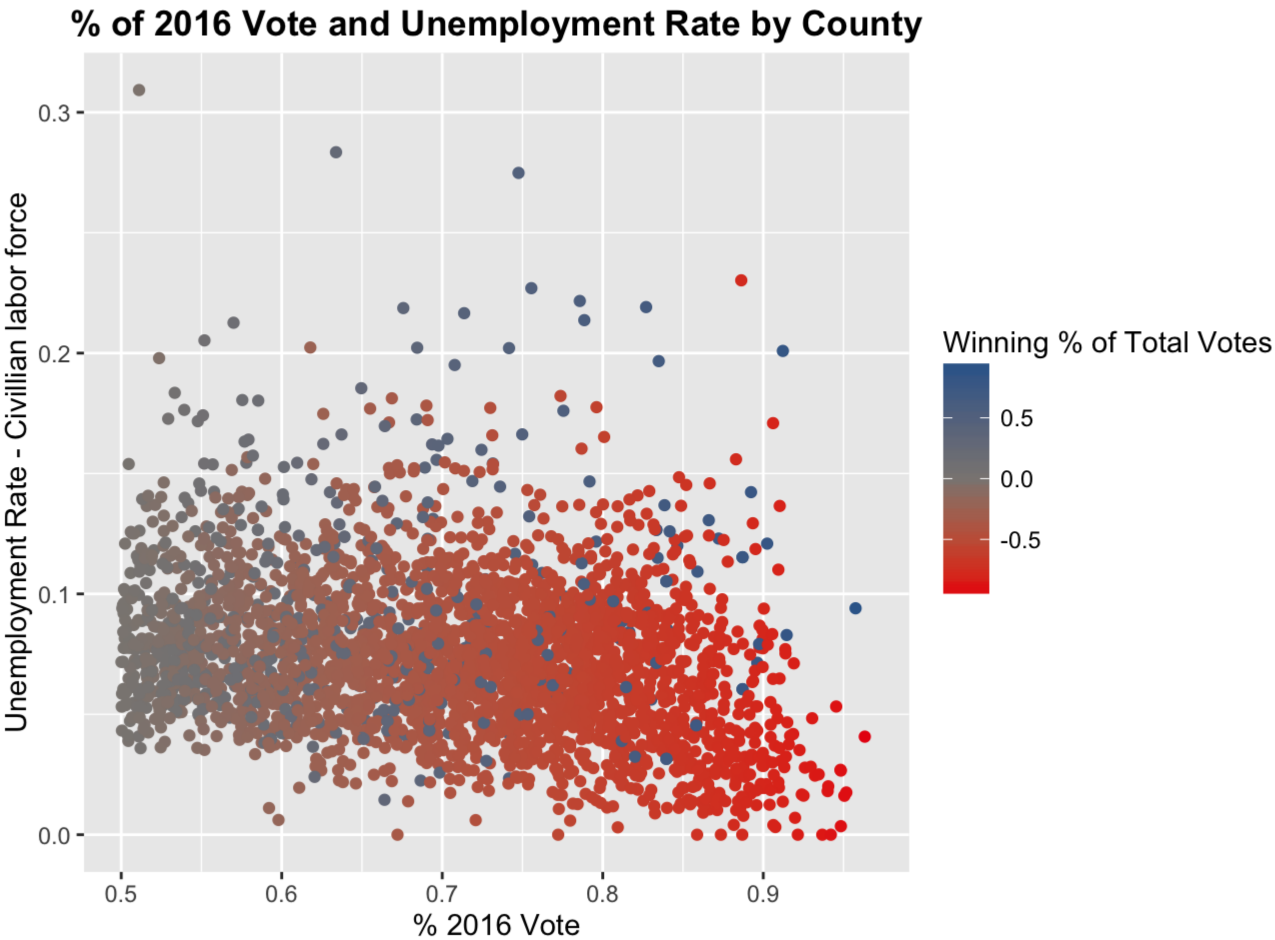
2 plots displayed similar distribution of the colored dot. Red dot, representing county where republic wins over democrats, are concentrated on middle and right region of graph. Blue color dot, representing county where democrats wins , are concentrated on the left region of graph.

```
# unemplo vs employed
# HC01_VC06,Estimate; EMPLOYMENT STATUS - In labor force - Civilian labor force
# HC01_VC07,Estimate; EMPLOYMENT STATUS - In labor force - Civilian labor force – Employed
# HC01_VC08,Estimate; EMPLOYMENT STATUS - In labor force - Civilian labor force – Unemployed
employmentDF = bigDF[,c("State","County","HC01_VC06","HC01_VC07","HC01_VC08","2016 Trump Votes
","2016 Clinton Votes","2012 Republican Votes", "2012 Democrat Votes")]
names(employmentDF) = c("State","County", "Total","Employed","Unemployed","2016 rep","2016 dem
","2012 rep","2012 dem")
employmentDF$unem.rate = employmentDF$Unemployed / employmentDF$Total
```

```
employmentDF$dem.per.2012 = employmentDF$`2012 dem`/(employmentDF$`2012 dem` + employmentDF$`2
012 rep`)
employmentDF$rep.per.2012 = employmentDF$`2012 rep`/(employmentDF$`2012 dem` + employmentDF$`2
012 rep`)
# positive value indicates democratic vote, negative value indicates republican vote
# magnitude is the severity of the victory
employmentDF$total.per.2012 = employmentDF$dem.per.2012 - employmentDF$rep.per.2012
# has the percentage value of the winner
employmentDF$winner.per.2012 = pmax(employmentDF$dem.per.2012, employmentDF$rep.per.2012)
employmentDF$dem.per.2016 = employmentDF$`2016 dem`/(employmentDF$`2016 dem` + employmentDF$`2
016 rep`)
employmentDF$rep.per.2016 = employmentDF$`2016 rep`/(employmentDF$`2016 dem` + employmentDF$`2
016 rep`)
# positive value indicates democratic vote, negative value indicates republican vote
# magnitude is the severity of the victory
employmentDF$total.per.2016 = employmentDF$dem.per.2016 - employmentDF$rep.per.2016
# has the percentage value of the winner
employmentDF$winner.per.2016 = pmax(employmentDF$dem.per.2016, employmentDF$rep.per.2016)


# scatterplot, every county is a point, x is % dem/rep vote, y is unemployment rate, grouping
is color by dem/rep vote?
#2012
ggplot(data=employmentDF) + geom_point(mapping = aes(x = winner.per.2012, y = unem.rate, color
 = total.per.2012)) + xlab("% 2012 Vote") + ylab("Unemployment Rate - Civillian labor force")
+ ggtitle("% of 2012 Vote and Unemployment Rate by County") + theme(plot.title = element_text(
face="bold")) + scale_colour_gradient2(name="Winning % of Total Votes ",low="#E91D0E", mid="se
ashell4",  high="#336699", midpoint = 0)
```



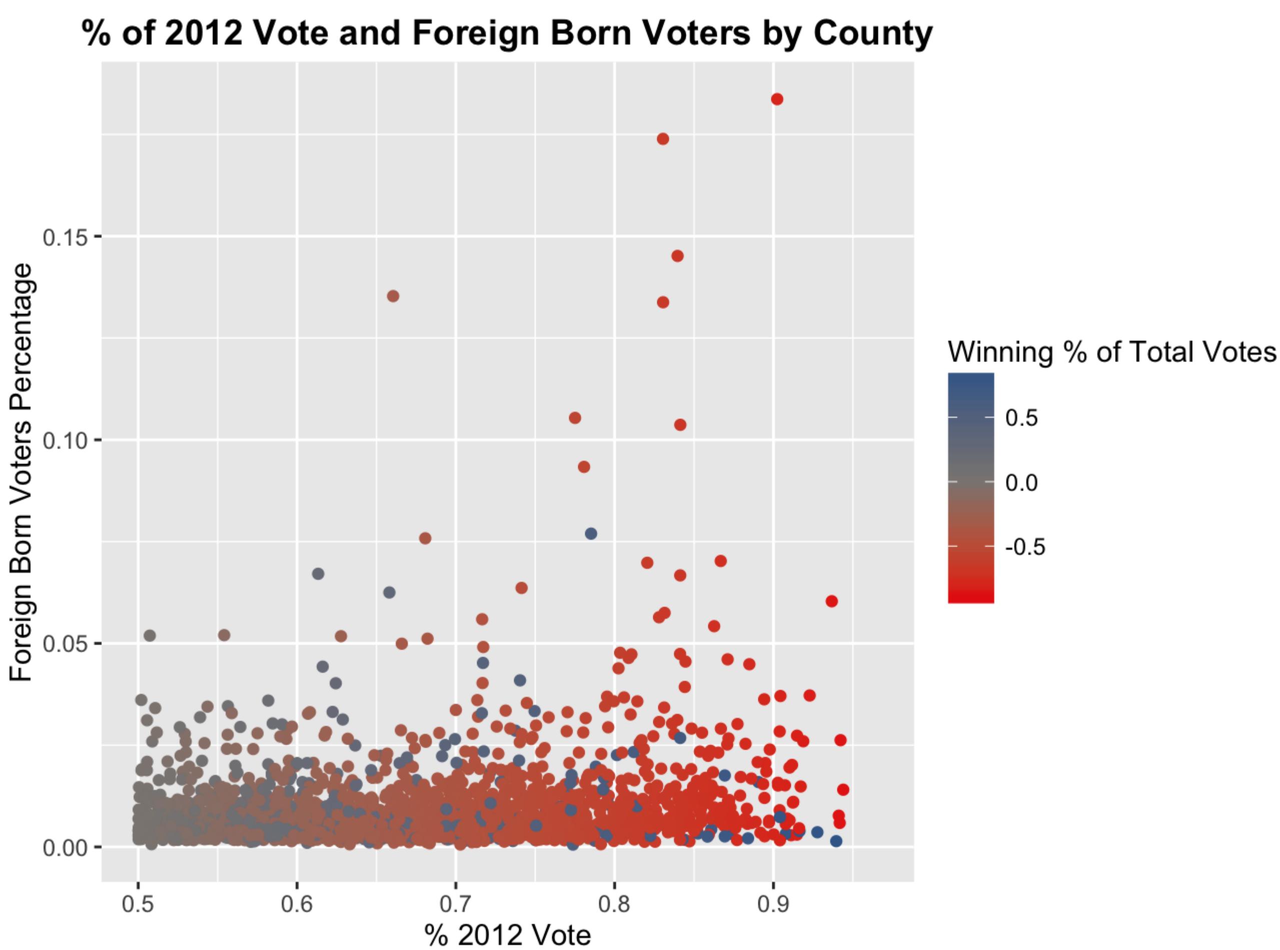

```
#2016
ggplot(data=employmentDF) + geom_point(mapping = aes(x = winner.per.2016, y = unem.rate, color
 = total.per.2016)) + xlab("% 2016 Vote") + ylab("Unemployment Rate - Civillian labor force")
+ ggtitle("% of 2016 Vote and Unemployment Rate by County") + theme(plot.title = element_text(
face="bold")) + scale_colour_gradient2(name="Winning % of Total Votes ",low="#E91D0E", mid="se
ashell4",  high="#336699", midpoint = 0)
```



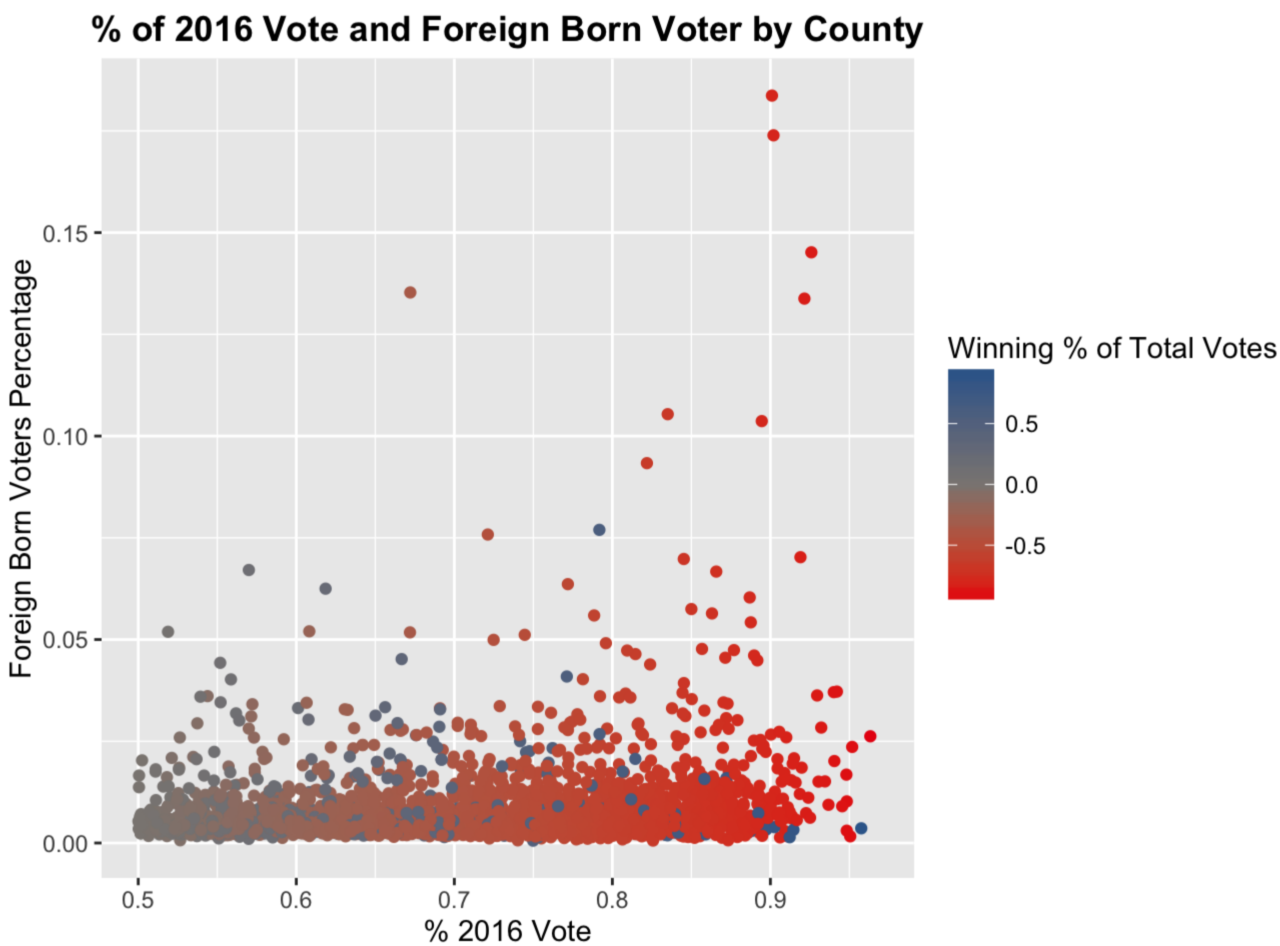
In plot 4, our team tried to find difference between native voters and non-native voters by county. And we compare this in two election years, 2016 and 2012. And we compare this relationship in two election years, 2016 and 2012. We first make a dataframe “birthDF” by using our primary dataframe “bigDF” and picking up residency and birth status “HC01_VC128, HC01_VC129 HC02_VC134”, “2016 Trump Votes”,“2016 Clinton Votes”,“2012 Republican Votes” and “2012 Democrat Votes” to implement our data analysis. Each year, we first calculate percentage of votes of each party by county. We sort winner party by using “pmax” function on the comparison of percentage of democrats votes and republican votes. And in the graph, republican-win county is represented by red dot and democrats-win county are represented by blue dot. We basically follow the same step for 2 election tears. We decided to make a scatter plots because each jitter point with different color will better represent votes of each county of interest. Apparently it we can tell by how much more votes approximately a party wins over the other party in each election within county. We decide to compare scatter plots of 2016 and 2012 to draw a conclusion. In each graph, every county is a point, x is by how many percent of votes a party win over the others, y foreign born voters percentage of county, grouping is color by “total.per.2012 and total.per.2016”, which is the percentage of democrats votes minors percentage of republican votes in 2 election years. 2 plots displayed similar distribution of the colored dot. Red dot, representing county where republic wins over democrats, are concentrated on middle and right region of graph. Blue color dot, representing county where democrats wins, are concentrated on the left region of graph. In 2 election years, scatters are both staying majorly at the bottom. It could be because of the fact the non-native voter’s population percentage are fairly low in each county. Or maybe voting right are primarily reserved to native American voters. Interestingly, outliers with higher non-native vote rate are largely republican-win county.

```
# native vs non-native
# HC01_VC128,Estimate; PLACE OF BIRTH - Total population
# HC01_VC129,Estimate; PLACE OF BIRTH – Native
# HC01_VC130,Estimate; PLACE OF BIRTH - Native - Born in United States
```

```
birthDF = bigDF[,c("State","County","HC01_VC128","HC01_VC129","HC02_VC134","2016 Trump Votes",
"2016 Clinton Votes","2012 Republican Votes", "2012 Democrat Votes")]
names(birthDF) = c("State","County", "Total","Native","Fborn","2016 rep","2016 dem","2012 rep"
,"2012 dem")
birthDF$fborn.rate= birthDF$Fborn/ birthDF$Total

birthDF$dem.per.2012 = birthDF$`2012 dem`/ (birthDF$`2012 dem`+ birthDF$`2012 rep`)
birthDF$rep.per.2012 = birthDF$`2012 rep`/ (birthDF$`2012 dem`+ birthDF$`2012 rep`)
birthDF$dem.per.2016 = birthDF$`2016 dem`/ (birthDF$`2016 dem`+ birthDF$`2016 rep`)
birthDF$rep.per.2016 = birthDF$`2016 rep`/ (birthDF$`2016 dem`+ birthDF$`2016 rep`)

birthDF$total.per.2012 = birthDF$dem.per.2012 - birthDF$rep.per.2012

birthDF$winner.per.2012 = pmax(birthDF$dem.per.2012, birthDF$rep.per.2012)

birthDF$total.per.2016 = birthDF$dem.per.2016 - birthDF$rep.per.2016
# has the percentage value of the winner
birthDF$winner.per.2016 = pmax(birthDF$dem.per.2016, birthDF$rep.per.2016)
#2012

ggplot(data=birthDF) + geom_point(mapping = aes(x = winner.per.2012, y = fborn.rate, color = t
otal.per.2012)) + xlab("% 2012 Vote") + ylab("Foreign Born Voters Percentage") + ggtitle("% of
 2012 Vote and Foreign Born Voters by County") + theme(plot.title = element_text(face="bold"))
 + scale_colour_gradient2(name="Winning % of Total Votes ",low="#E91D0E", mid="seashell4",  hi
gh="#336699", midpoint = 0)
```

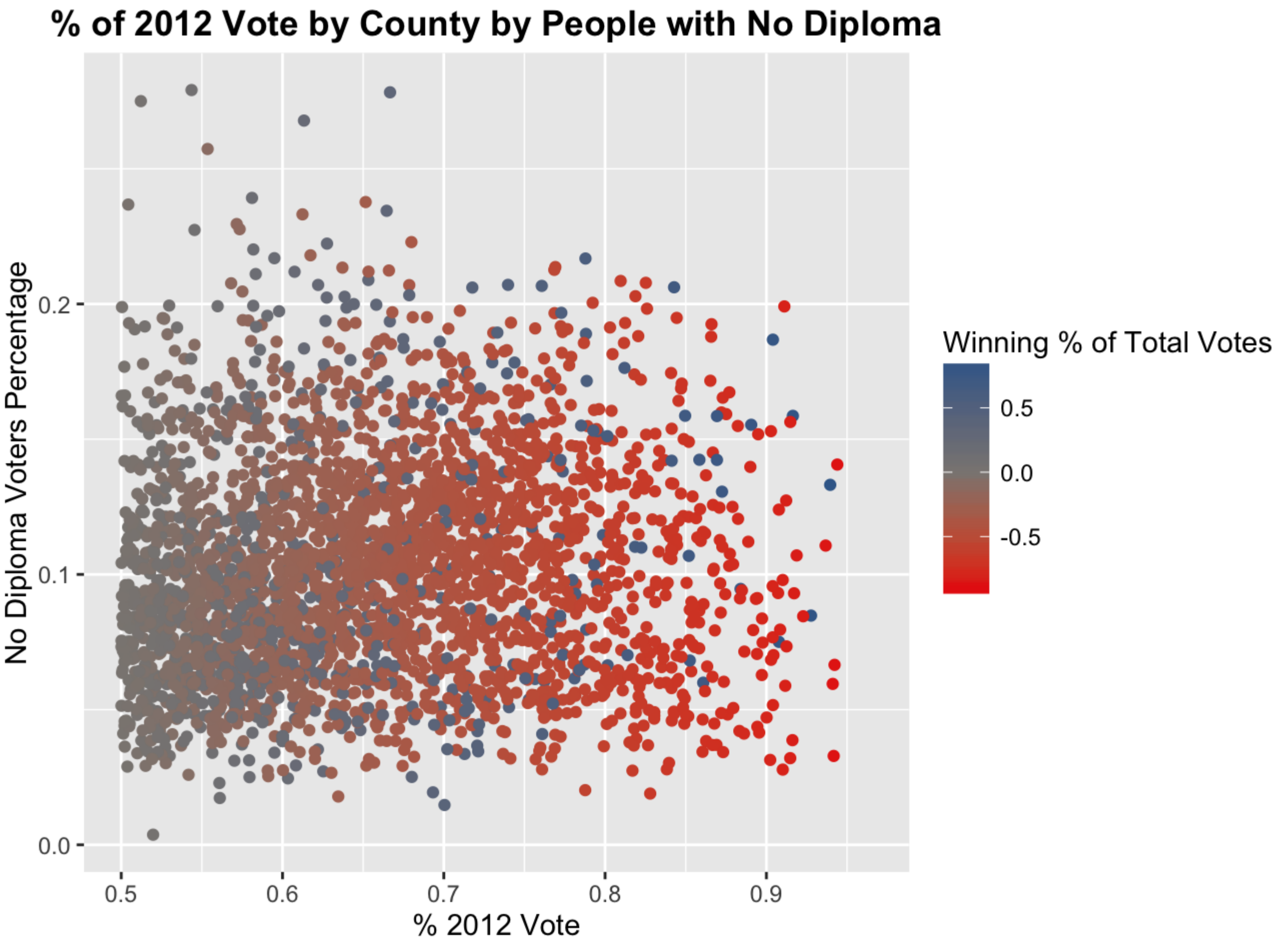


```
#2016
ggplot(data=birthDF) + geom_point(mapping = aes(x = winner.per.2016, y = fborn.rate, color = t
otal.per.2016)) + xlab("% 2016 Vote") + ylab("Foreign Born Voters Percentage") + ggtitle("% of
 2016 Vote and Foreign Born Voter by County") + theme(plot.title = element_text(face="bold"))
+ scale_colour_gradient2(name="Winning % of Total Votes ",low="#E91D0E", mid="seashell4",  hig
h="#336699", midpoint = 0)
```

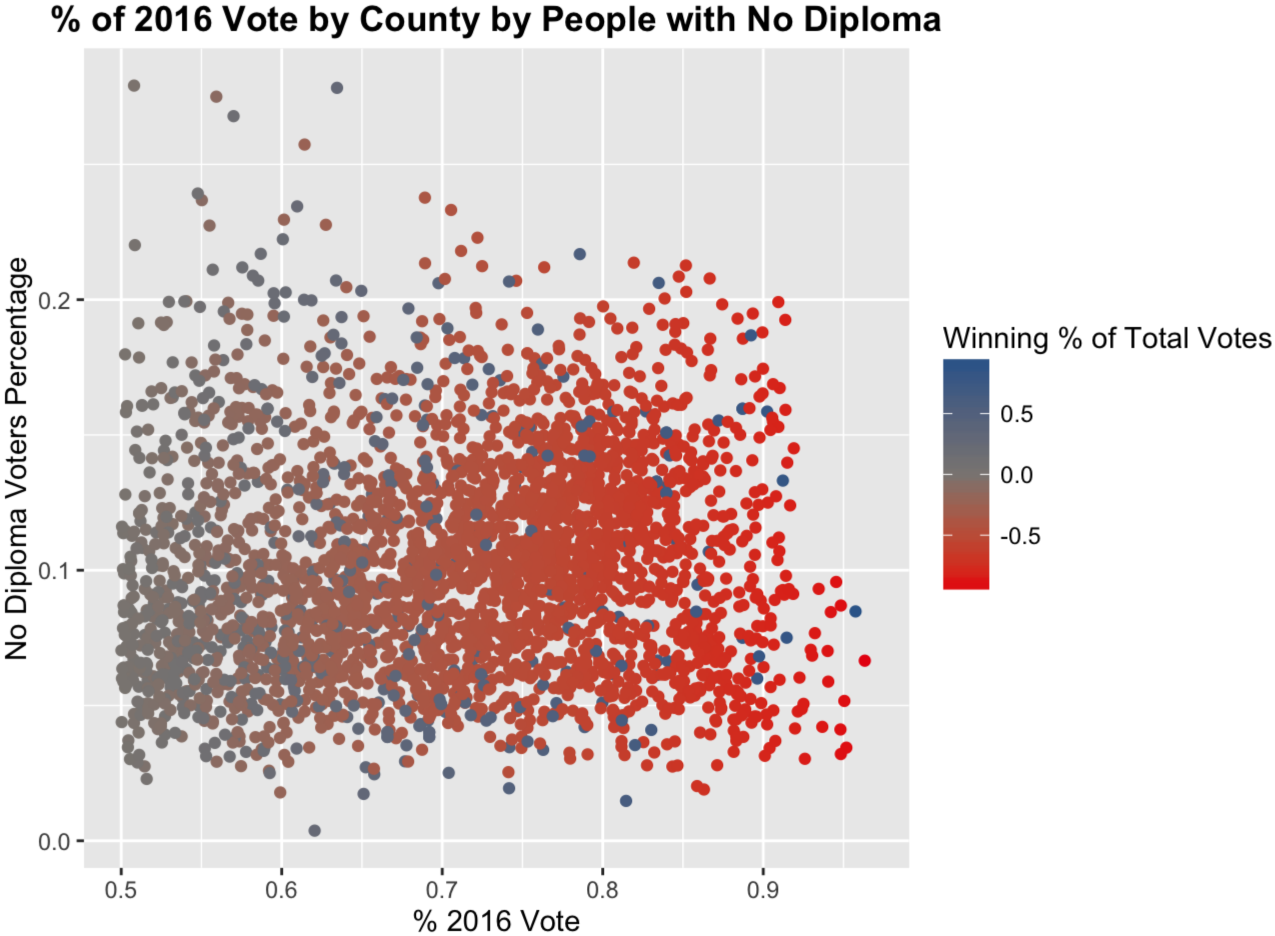


In plot 5, our team tried to find how under-high-school degree voters differ from bachelor degree voters by county. And we compare this in two election years, 2016 and 2012. And we compare this relationship in two election years, 2016 and 2012. We first make a dataframe “educationDF” by using our primary dataframe “bigDF” and picking up “State”,“County”,“Total Under-high-school and Bachelor population”, “Under-high-school population”,“Bachelor population”“2016 Trump Votes”,“2016 Clinton Votes”,“2012 Republican Votes” and “2012 Democrat Votes” to implement our data analysis. In each county, we calculate proportion of under-high-school voters and bachelor degree voters in each county. Each year, we first calculate percentage of votes of each party by county. We sort winner party by using “pmax” function on the comparison of percentage of democrats votes and republican votes. And in the graph, republican-win county is represented by red dot and democrats-win county are represented by blue dot. We basically follow the same step for 2 election tears. Every county is a point, x is by how many percent of votes a party win over the others, y percentage of under-high-school voters or percentage of bachelor-degree voters, grouping is color by “total.per.2012 and total.per.2016”, which is the percentage of democrats votes minors percentage of republican votes in 2 election years.Red dot, representing county where republic wins over democrats, are concentrated on middle and right region of graph. Blue color dot, representing county where democrats wins, are concentrated on the left region of graph. In comparison between percentage of under-high-school voters and bachelor degree voters of 2012, we can’t draw any conclusion for democrats-win county, for scatters of which stay in similar region. For republican-win county, there are more higher scatters on graph with bachelor degree compared to graph with under-high-school. Conclusion can be made that high education voters are more in favor of republican. Whereas at 2016, there’re more proportion of under-high-school in republican-win county.

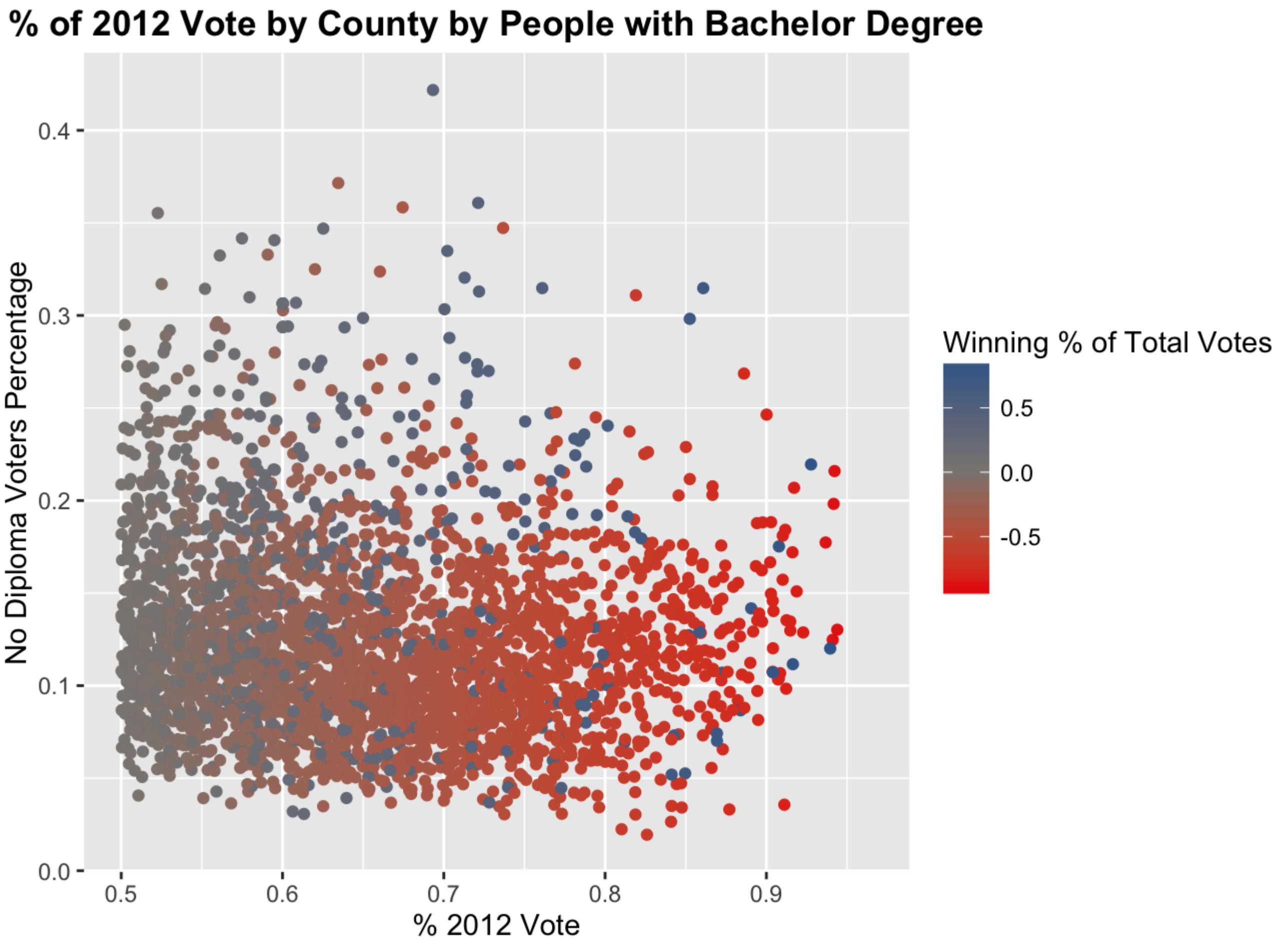
```
# 9-12 vs bachelor dooooone

educationDF = bigDF[,c("State","County","HC01_VC84.x","HC01_VC86","HC01_VC90","2016 Trump Vote
s","2016 Clinton Votes","2012 Republican Votes", "2012 Democrat Votes")]
names(educationDF) = c("State","County", "Total","Underhighschool","Bachelor","2016 rep","2016
 dem","2012 rep","2012 dem")

educationDF$underhighschool.rate= educationDF$Underhighschool/ educationDF$Total
educationDF$bachelor.rate= educationDF$Bachelor/ educationDF$Total

educationDF$dem.per.2012 = educationDF$`2012 dem`/ (educationDF$`2012 dem`+ educationDF$`2012
rep`)
educationDF$rep.per.2012 = educationDF$`2012 rep`/ (educationDF$`2012 dem`+ educationDF$`2012
rep`)
educationDF$dem.per.2016 = educationDF$`2016 dem`/ (educationDF$`2016 dem`+ educationDF$`2016
rep`)
educationDF$rep.per.2016 = educationDF$`2016 rep`/ (educationDF$`2016 dem`+ educationDF$`2016
rep`)

educationDF$total.per.2012 = educationDF$dem.per.2012 - educationDF$rep.per.2012
educationDF$winner.per.2012 = pmax(educationDF$dem.per.2012, educationDF$rep.per.2012)
educationDF$total.per.2016 = educationDF$dem.per.2016 - educationDF$rep.per.2016
# has the percentage value of the winner
educationDF$winner.per.2016 = pmax(educationDF$dem.per.2016, educationDF$rep.per.2016)

#2012 9-12 grade
ggplot(data=educationDF) + geom_point(mapping = aes(x = winner.per.2012, y = underhighschool.r
ate, color = total.per.2012)) + xlab("% 2012 Vote") + ylab("No Diploma Voters Percentage") + g
gtitle("% of 2012 Vote by County by People with No Diploma") + theme(plot.title = element_text
(face="bold")) + scale_colour_gradient2(name="Winning % of Total Votes ",low="#E91D0E", mid="s
eashell4",  high="#336699", midpoint = 0)
```



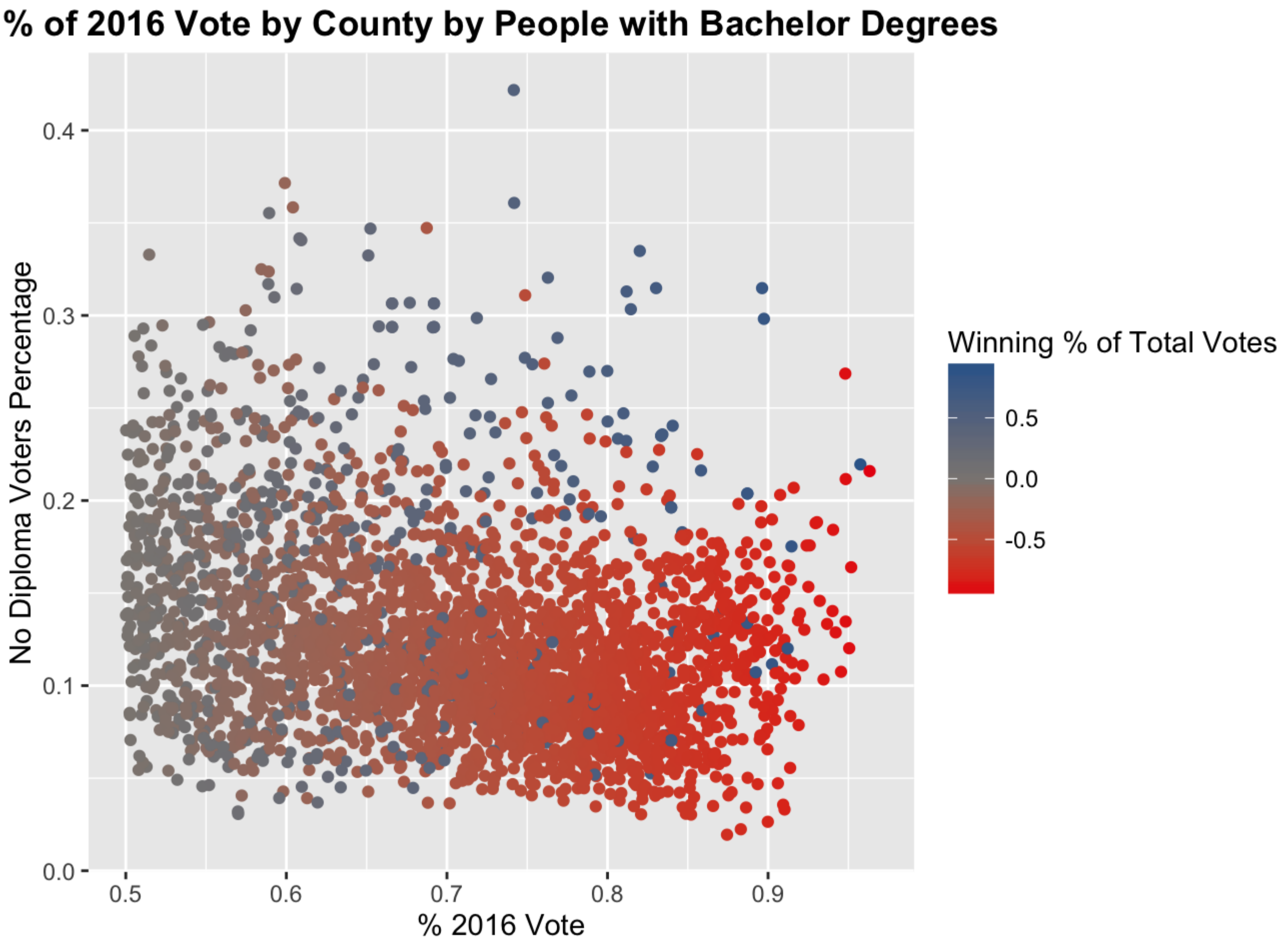

```
#2016 9-12 grade
ggplot(data=educationDF) + geom_point(mapping = aes(x = winner.per.2016, y = underhighschool.r
ate, color = total.per.2016)) + xlab("% 2016 Vote") + ylab("No Diploma Voters Percentage") + g
gtitle("% of 2016 Vote by County by People with No Diploma") + theme(plot.title = element_text
(face="bold")) + scale_colour_gradient2(name="Winning % of Total Votes ",low="#E91D0E", mid="s
eashell4",  high="#336699", midpoint = 0)
```




```
#2012 bachelor degree
ggplot(data=educationDF) + geom_point(mapping = aes(x = winner.per.2012, y = bachelor.rate, co
lor = total.per.2012)) + xlab("% 2012 Vote") + ylab("No Diploma Voters Percentage") + ggtitle(
"% of 2012 Vote by County by People with Bachelor Degree") + theme(plot.title = element_text(f
ace="bold")) + scale_colour_gradient2(name="Winning % of Total Votes ",low="#E91D0E", mid="sea
shell4",  high="#336699", midpoint = 0)
```



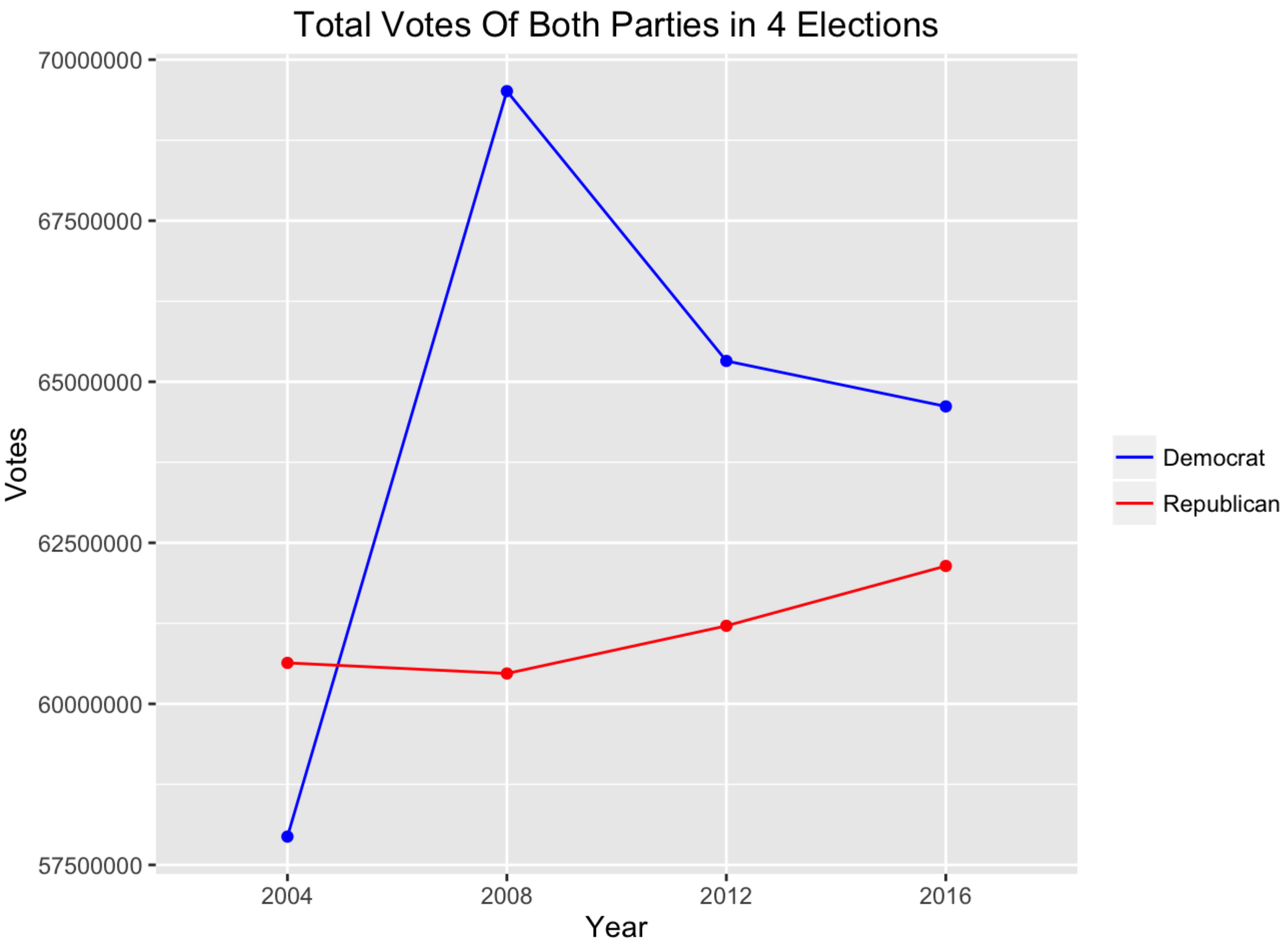

```
#2016 bachelor degree
ggplot(data=educationDF) + geom_point(mapping = aes(x = winner.per.2016, y = bachelor.rate, co
lor = total.per.2016)) + xlab("% 2016 Vote") + ylab("No Diploma Voters Percentage") + ggtitle(
"% of 2016 Vote by County by People with Bachelor Degrees") + theme(plot.title = element_text(
face="bold")) + scale_colour_gradient2(name="Winning % of Total Votes ",low="#E91D0E", mid="se
ashell4",  high="#336699", midpoint = 0)
```



In plot 6, our team wanted to compare total votes in 4 election years. We decide to use line plot because it will directly show the number of votes change in 4 elections for both parties. We first make a dataframe “educationDF” by using our primary dataframe “partyDF” and subset “State”,“County”,“2016 Trump Votes”,“2016 Clinton Votes”,“2012 Republican Votes”, “2012 Democrat Votes”, “2008 McCain Votes”, “2008 Obama Votes”, “2004 Bush Votes”, “2004 Kerry Votes” from our original dataframe “bigDF”. In addition, we add a new dataframe called “newpartyDF” by summing up total votes of democrats and republican in 4 years. And we drop all NA data. Also, we calculate total vote number of each year. We use out newpartyDf to make our data frame. For first line plot, we set x as year of election, y as total number of votes and color as “democrats”. For second line plot, we set x as year of election, y as total number of votes and color as “republican”. We find that the total number of votes of 2008, 2012 and 2016 stays similar because that voting population remain the same in the passing 12 years. Furthermore, by insititution, we assume that whoever has more votes wins the election. In our plot, it states correct in 2004, 2008, 2012. However, in 2016, Democrat won more votes than Republican, and Donald Trump still wins the election, which overturned our assumption. Therefore, we know that more votes doesn’t mean wining the election.

```
# Democrat Vs Republican
partyDF = bigDF[,c("State","County","2016 Trump Votes","2016 Clinton Votes","2012 Republican V
otes", "2012 Democrat Votes", "2008 McCain Votes", "2008 Obama Votes", "2004 Bush Votes", "200
4 Kerry Votes")]
names(partyDF) = c("State","County","2016 rep","2016 dem","2012 rep","2012 dem","2008 rep","20
08 dem","2004 rep","2004 dem")

newpartyDF = data.frame(c(sum(partyDF$`2016 dem`, na.rm=TRUE), sum(partyDF$`2012 dem`, na.rm=T
RUE), sum(partyDF$`2008 dem`, na.rm=TRUE), sum(partyDF$`2004 dem`, na.rm=TRUE)), c(sum(partyDF
$`2016 rep`, na.rm=TRUE), sum(partyDF$`2012 rep`, na.rm=TRUE), sum(partyDF$`2008 rep`, na.rm=T
RUE), sum(partyDF$`2004 rep`, na.rm=TRUE)), factor(c(2016,2012,2008,2004)))
```

```
newpartyDF$Total = newpartyDF[,1] + newpartyDF[,2]
names(newpartyDF) = c("Total Democrat Votes", "Total Republican Votes", "Year", "Total")

ggplot(data=newpartyDF,aes(group=1))+ geom_line(aes( x = newpartyDF$Year, y =newpartyDF$`Total
 Democrat Votes`,colour='Democrat')) +geom_line(aes( x = newpartyDF$Year, y =newpartyDF$`Total
 Republican Votes`,colour='Republican')) +labs(title= "Total Votes Of Both Parties in 4 Electi
ons", x= "Year", y= "Votes")+scale_colour_manual("", breaks = c("Democrat", "Republican"),valu
es = c("blue", "red"))+geom_point(aes(x=newpartyDF$Year, y=newpartyDF$`Total Democrat Votes`),
 color='blue')+geom_point(aes(x=newpartyDF$Year, y=newpartyDF$`Total Republican Votes`), color
='red')
```

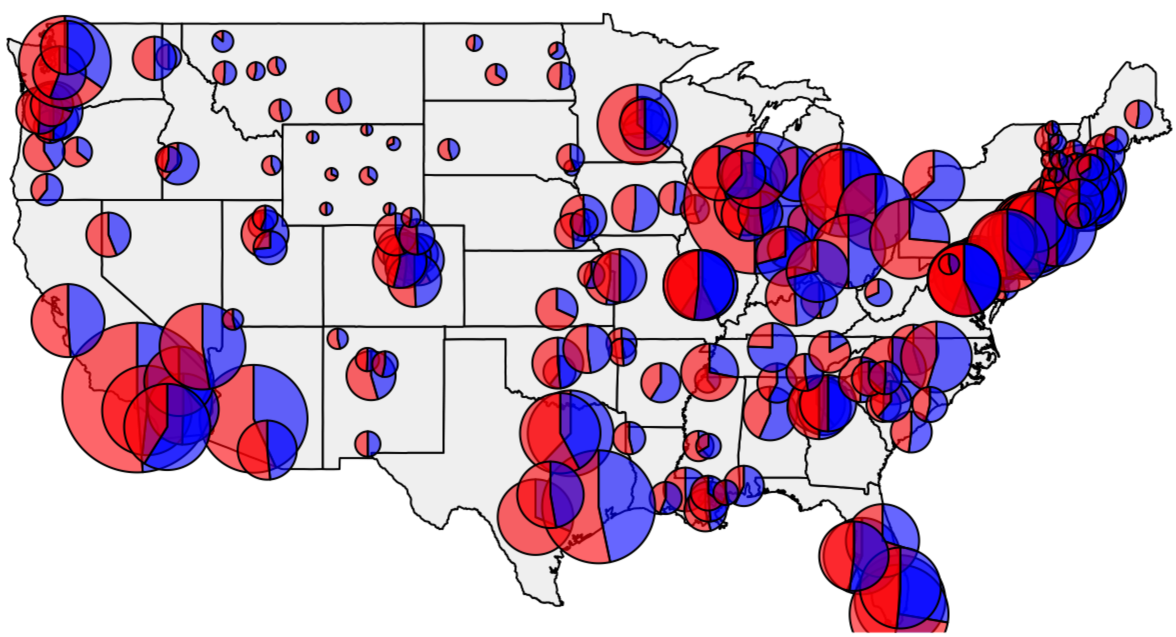


3. Map

For the election map, our group is trying to draw the proportion of voting in different county. By using base R graphing,we use the longitude and latitude from data frame to identify each county location. We extract state names, county names, longitude, latitude and 2016 votes for Clinton and Trump. We omit all NA in the data and calculate the total vote from each county. At the beginning, we draw the pie graphs for each county that have portion of Clinton’s vote and trump’s votes, and the radius of each pie graph is depending on the number of total vote for each county. However, some counties have so little vote that radius tends to be zero, and become black little dot. Hence, we want to delete those counties votes if they have too little portion in their states. We use dglyr library to group all the vote by state. Since the votes are distributing according to the portion in each state, we group the vote by different state, and calculate the portion of vote from each county in the state. Since there are so many small counties, we only take the voting data from counties that have more than 5 percent total number of votes of their states.

```
library(maps)
library(mapdata)
```

```
library(mapplots)
library(dplyr)
library(scales)
myDF=bigDF[,c(1,2,3,4,84,85)]
myDF=na.omit(myDF)
myDF$votes=myDF[,3]+myDF[,4]
vote= group_by (myDF,State)
vote=summarise(vote, totalvote=sum(votes))
vote1=left_join (myDF,vote,by="State")
vote1$percentVote=vote1$votes/vote1$totalvote
votefinal=vote1[vote1$percentVote>=0.05,]
ClinVote=as.numeric(votefinal[,3])
TrumVote=as.numeric(votefinal[,4])
z=array(c(abs(ClinVote/(ClinVote+TrumVote)),abs(TrumVote/(ClinVote+TrumVote))))
z=matrix(z,ncol=2,byrow=TRUE)
map("state", col="gray95", fill=TRUE)
points(votefinal$Longitude, votefinal$Latitude)
text(x= votefinal$Longitude, votefinal$Latitude,labels = votefinal$County, pos=4)
draw.pie(z=z,x=as.numeric(votefinal$Longitude)/1000000,y=as.numeric(votefinal$Latitude)/100000
0,radius=sqrt(ClinVote+TrumVote)/500, col=c(alpha("blue",0.6), alpha("red",0.6)))
```



4. predicting the 2016 result (Made by Chase Humiston) Method Use building regression line predict 2016 election results.

Our group recombined the data frame with all different group of people’s votes, such as the people with high level of education, native people, and so on. We use those specific group of people so that the prediction of data will be more accurate. We set up the seed to generate random number, and put them in matrix as folds. To have less error in the predictor, we tried different cp values. By taking all columns except for the actual vote in column three, we use for loop and rpart function to generate and predict the data for each cp value. We apply the function on each prediction data set and find out the the ones that have sum more than 0.5.

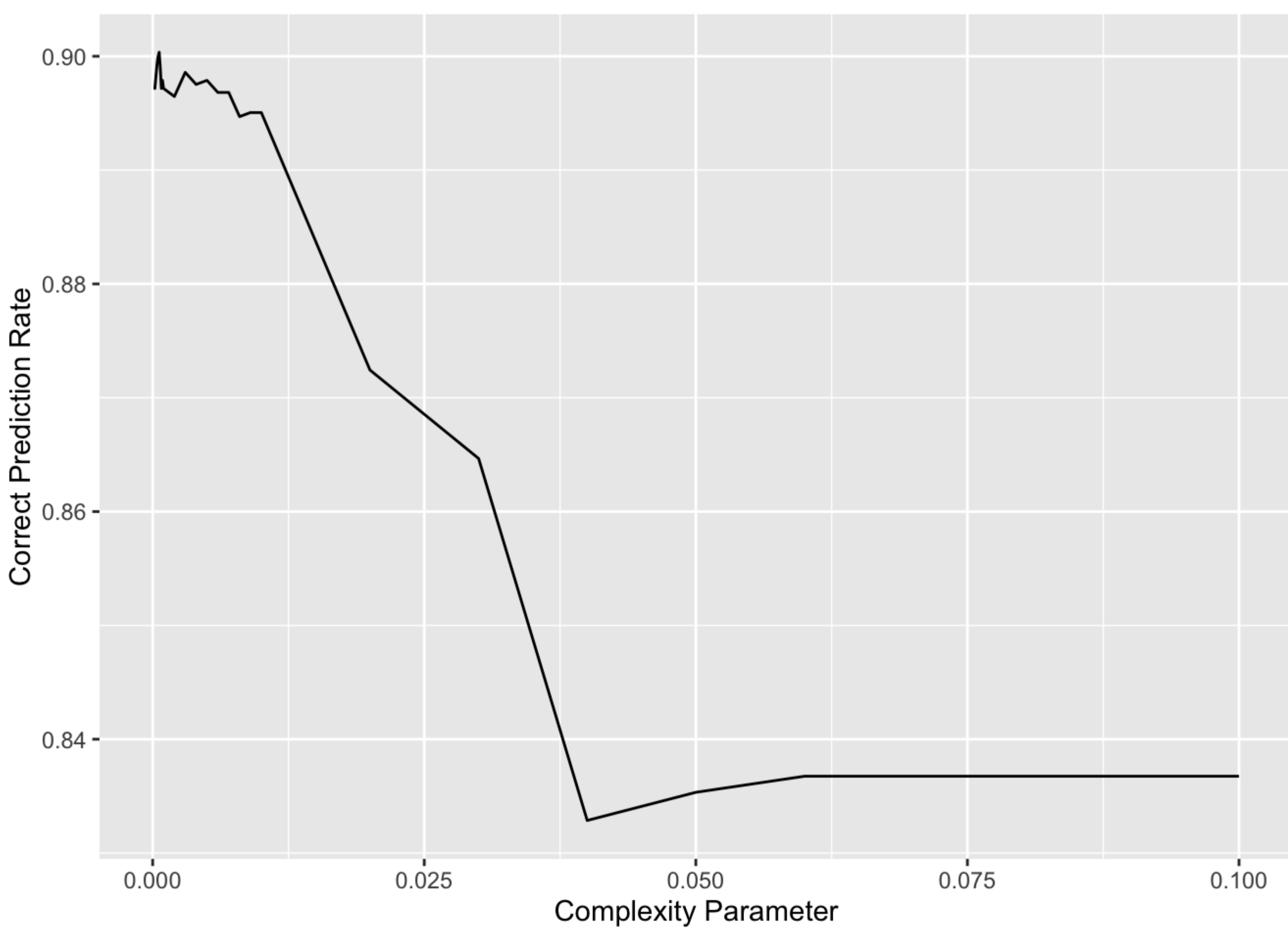
Then, we select the maximum cpr value to use for the actual prediction to be more precise. We draw a plot with cps and cpr values to figure our the relations between them. Using exactly the same steps, we redo the previous steps with different ranges of cp values. Then, we draw the plot again for relations between cps and cpr. Looking at the plot, we find out when cp value is 0.0008, we have maximum of correct prediction rate. By looking at both the plot and the table, we see that when cp value is 0.0008, the error percentage is 0.17177 which means we predict 0.82823 correct.

```
# Building a regression tree
meddf = bigDF[,c("State","County","2016 Clinton Votes","2016 Trump Votes","HC03_VC04","HC03_VC06","HC03_VC07","HC03_VC08","HC03_VC09","HC03_VC10","HC03_VC11","HC03_VC12","HC03_VC13.x","HC03_VC14","HC03_VC15","HC03_VC17","HC03_VC18","HC03_VC85","HC03_VC86","HC03_VC87","HC03_VC88","HC03_VC89","HC03_VC90","HC03_VC91","HC03_VC129","HC03_VC130","HC03_VC131","HC03_VC132","HC03_VC133","HC03_VC134","HC03_VC05","HC03_VC13.y","HC03_VC75","HC03_VC76","HC03_VC77","HC03_VC78","HC03_VC79","HC03_VC80","HC03_VC81","HC03_VC82","HC03_VC83","HC03_VC84","HC01_VC85.y","HC03_VC156","Longitude","Latitude")]
meddf[,-(1:2)] = sapply(meddf[,-(1:2)],as.numeric)
meddf = meddf[complete.cases(meddf),]
meddf$State = as.factor(meddf$State)
pred2016 = data.frame("repvotes" = meddf[,4]/(meddf[,3]+meddf[,4]), meddf[,-(3:4)])


#separate predictor build and test data
set.seed(98070320)
npred = nrow(pred2016)
splitset = sample(npred, size = round(npred * .9, -1), replace = FALSE)
testset = pred2016[-splitset, ]
buildset = pred2016[splitset, ]

#create folds matrix
nbuild = nrow(buildset)
permuteIndices = sample(nbuild)
folds = matrix(permuteIndices, ncol = 10)

#cp value vector
cps = c(seq(0.0001, 0.001, by = 0.0001),
       seq(0.001, 0.01, by = 0.001),
       seq(0.01, 0.1, by = 0.01))

#prediction matrix
preds = matrix(nrow = nbuild, ncol = length(cps))
tree = list()

require(rpart)

#build tree
for (i in 1:10) {
  buildfold = folds[, -i]
  testfold = folds[, i]
```

```
  for (j in 1:length(cps)) {
    tree[[j]] = rpart( repvotes ~ .,
            data = buildset[buildfold, -(3)],
            method = "anova",
            control = rpart.control(cp = cps[j]))
    preds[testfold, j] =
      predict(tree[[j]],
              newdata = buildset[testfold, -c(1,3)],
              type = "vector")
  }
}

#use correct prediction rates to select cp value for predicor
cpr = apply(preds, 2, function(oneSet) {
  return(sum(c(oneSet>.5)==c(buildset[,1]>.5))/nbuild)
})

which.max(cpr)
```

```
## [1] 6
```

```
cprdf = data.frame(cps, cpr)
ggplot(data = cprdf, aes(x = cps, y = cpr)) +
  geom_line() +
  labs(x = "Complexity Parameter", y = "Correct Prediction Rate")
```



```
for(i in 1:10){
  printcp(tree[[i]])
}
```

```
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC08
##  [6] HC03_VC09   HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129
## [11] HC03_VC13.x HC03_VC13.y HC03_VC130  HC03_VC131  HC03_VC132
## [16] HC03_VC133  HC03_VC134  HC03_VC14   HC03_VC15   HC03_VC156
## [21] HC03_VC17   HC03_VC18   HC03_VC75   HC03_VC76   HC03_VC77
## [26] HC03_VC78   HC03_VC79   HC03_VC80   HC03_VC81   HC03_VC82
## [31] HC03_VC83   HC03_VC84   HC03_VC85   HC03_VC86   HC03_VC87
## [36] HC03_VC88   HC03_VC89   HC03_VC90   HC03_VC91   Latitude
## [41] Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##             CP nsplit rel error  xerror     xstd
## 1   0.26894188      0   1.00000 1.00060 0.031334
## 2   0.15300146      1   0.73106 0.73643 0.023451
## 3   0.04905506      2   0.57806 0.58958 0.020901
## 4   0.03483358      3   0.52900 0.53984 0.019920
## 5   0.02575390      4   0.49417 0.51093 0.019307
## 6   0.02255952      5   0.46841 0.49562 0.018471
## 7   0.02195363      6   0.44585 0.49419 0.018285
## 8   0.01812404      7   0.42390 0.47646 0.017220
## 9   0.01460201      8   0.40578 0.46132 0.017048
## 10  0.01373019      9   0.39117 0.44618 0.016689
## 11  0.01318438     10   0.37744 0.43350 0.016484
## 12  0.01113299     11   0.36426 0.42325 0.016196
## 13  0.01044189     12   0.35313 0.41953 0.016186
## 14  0.00876037     13   0.34269 0.41599 0.016108
## 15  0.00771485     14   0.33393 0.40633 0.015347
## 16  0.00737522     15   0.32621 0.40716 0.015450
## 17  0.00651898     16   0.31884 0.40303 0.015399
## 18  0.00609098     17   0.31232 0.40237 0.015363
## 19  0.00582690     18   0.30623 0.40261 0.015740
## 20  0.00523612     19   0.30040 0.39500 0.015544
## 21  0.00482466     20   0.29516 0.39270 0.015241
## 22  0.00479629     21   0.29034 0.39517 0.015487
## 23  0.00455377     22   0.28554 0.39640 0.015521
## 24  0.00406023     23   0.28099 0.39004 0.015212
## 25  0.00390075     24   0.27693 0.38808 0.015236
## 26  0.00386309     25   0.27303 0.38666 0.015304
## 27  0.00373690     26   0.26916 0.38525 0.015297
```

```
## 28  0.00323021     27   0.26543 0.38076 0.015068
## 29  0.00287222     28   0.26220 0.37850 0.014835
## 30  0.00276227     29   0.25932 0.37629 0.014708
## 31  0.00273625     30   0.25656 0.37533 0.014709
## 32  0.00269698     31   0.25383 0.37514 0.014702
## 33  0.00265792     32   0.25113 0.37438 0.014689
## 34  0.00261475     33   0.24847 0.37416 0.014686
## 35  0.00259402     34   0.24586 0.37330 0.014664
## 36  0.00242374     35   0.24326 0.37332 0.014779
## 37  0.00234022     36   0.24084 0.37024 0.014713
## 38  0.00226835     37   0.23850 0.37107 0.014736
## 39  0.00217154     39   0.23396 0.37087 0.014747
## 40  0.00203158     40   0.23179 0.37154 0.015028
## 41  0.00200903     41   0.22976 0.36905 0.014958
## 42  0.00197756     43   0.22574 0.36816 0.014948
## 43  0.00192518     44   0.22376 0.36702 0.014934
## 44  0.00180554     45   0.22184 0.36682 0.015000
## 45  0.00174488     47   0.21823 0.36976 0.015152
## 46  0.00171649     48   0.21648 0.36939 0.015192
## 47  0.00171513     49   0.21476 0.36973 0.015184
## 48  0.00163250     50   0.21305 0.36946 0.015207
## 49  0.00160365     51   0.21142 0.36864 0.015182
## 50  0.00154749     52   0.20981 0.36985 0.015204
## 51  0.00145486     53   0.20827 0.36981 0.015213
## 52  0.00145420     54   0.20681 0.37015 0.015223
## 53  0.00142485     55   0.20536 0.36934 0.015167
## 54  0.00140643     56   0.20393 0.36917 0.015170
## 55  0.00139441     58   0.20112 0.36960 0.015170
## 56  0.00136615     59   0.19972 0.36947 0.015178
## 57  0.00134432     62   0.19563 0.36940 0.015172
## 58  0.00132904     63   0.19428 0.36923 0.015173
## 59  0.00132240     64   0.19295 0.36983 0.015204
## 60  0.00123928     65   0.19163 0.36856 0.015183
## 61  0.00123217     66   0.19039 0.36812 0.015177
## 62  0.00116417     67   0.18916 0.36738 0.015148
## 63  0.00115728     68   0.18799 0.36803 0.015220
## 64  0.00111347     70   0.18568 0.36825 0.015222
## 65  0.00107149     71   0.18457 0.36852 0.015303
## 66  0.00106199     72   0.18350 0.36702 0.015268
## 67  0.00099296     73   0.18243 0.36725 0.015345
## 68  0.00097298     74   0.18144 0.36948 0.015488
## 69  0.00096571     75   0.18047 0.36941 0.015501
## 70  0.00094299     76   0.17950 0.36998 0.015519
## 71  0.00092604     77   0.17856 0.37055 0.015541
## 72  0.00086791     78   0.17763 0.37156 0.015585
## 73  0.00086082     80   0.17590 0.37109 0.015605
## 74  0.00083882     81   0.17504 0.37031 0.015558
## 75  0.00082185     82   0.17420 0.37032 0.015563
## 76  0.00081999     83   0.17338 0.37073 0.015619
## 77  0.00081510     84   0.17256 0.37041 0.015619
## 78  0.00080718     85   0.17174 0.37112 0.015655
## 79  0.00080253     86   0.17093 0.37087 0.015652
## 80  0.00080214     87   0.17013 0.37065 0.015652
## 81  0.00079999     88   0.16933 0.37036 0.015647
```

```
## 82  0.00077675     89   0.16853 0.37153 0.015728
## 83  0.00076692     90   0.16775 0.37091 0.015715
## 84  0.00075579     91   0.16698 0.37126 0.015720
## 85  0.00074858     92   0.16623 0.37181 0.015736
## 86  0.00070380     94   0.16473 0.37057 0.015729
## 87  0.00069992     95   0.16403 0.37065 0.015701
## 88  0.00069223     96   0.16333 0.37231 0.015737
## 89  0.00068902     97   0.16264 0.37143 0.015663
## 90  0.00068832     98   0.16195 0.37143 0.015663
## 91  0.00066915     99   0.16126 0.37178 0.015659
## 92  0.00066580    100   0.16059 0.37261 0.015680
## 93  0.00063716    103   0.15859 0.37354 0.015665
## 94  0.00063355    104   0.15795 0.37339 0.015649
## 95  0.00061813    105   0.15732 0.37451 0.015673
## 96  0.00061596    107   0.15609 0.37494 0.015713
## 97  0.00061410    108   0.15547 0.37492 0.015715
## 98  0.00060509    109   0.15486 0.37473 0.015707
## 99  0.00059366    110   0.15425 0.37504 0.015709
## 100 0.00057957    111   0.15366 0.37536 0.015537
## 101 0.00057526    112   0.15308 0.37499 0.015539
## 102 0.00055524    114   0.15193 0.37495 0.015537
## 103 0.00055083    115   0.15137 0.37485 0.015539
## 104 0.00054831    116   0.15082 0.37452 0.015542
## 105 0.00054719    117   0.15027 0.37469 0.015543
## 106 0.00049059    118   0.14972 0.37552 0.015582
## 107 0.00048324    119   0.14923 0.37483 0.015544
## 108 0.00048039    120   0.14875 0.37537 0.015631
## 109 0.00046824    121   0.14827 0.37549 0.015575
## 110 0.00046728    122   0.14780 0.37506 0.015546
## 111 0.00046207    123   0.14733 0.37506 0.015546
## 112 0.00044998    124   0.14687 0.37446 0.015509
## 113 0.00043664    125   0.14642 0.37418 0.015501
## 114 0.00043135    126   0.14599 0.37427 0.015512
## 115 0.00042858    127   0.14555 0.37443 0.015512
## 116 0.00041923    128   0.14513 0.37377 0.015491
## 117 0.00041219    129   0.14471 0.37363 0.015448
## 118 0.00040882    130   0.14429 0.37380 0.015456
## 119 0.00040711    131   0.14389 0.37412 0.015491
## 120 0.00040614    132   0.14348 0.37412 0.015491
## 121 0.00039895    133   0.14307 0.37434 0.015494
## 122 0.00038747    134   0.14267 0.37353 0.015486
## 123 0.00038480    135   0.14229 0.37329 0.015486
## 124 0.00038424    136   0.14190 0.37344 0.015486
## 125 0.00037809    137   0.14152 0.37354 0.015484
## 126 0.00037584    138   0.14114 0.37336 0.015480
## 127 0.00036870    140   0.14039 0.37356 0.015485
## 128 0.00034819    141   0.14002 0.37350 0.015472
## 129 0.00034302    142   0.13967 0.37379 0.015472
## 130 0.00033810    143   0.13933 0.37387 0.015472
## 131 0.00033694    144   0.13899 0.37442 0.015486
## 132 0.00033661    145   0.13865 0.37425 0.015469
## 133 0.00033647    146   0.13832 0.37425 0.015469
## 134 0.00032613    147   0.13798 0.37449 0.015445
## 135 0.00031906    148   0.13765 0.37482 0.015451
```

```
## 136 0.00031893    149   0.13733 0.37529 0.015451
## 137 0.00031861    150   0.13702 0.37529 0.015451
## 138 0.00031509    151   0.13670 0.37538 0.015451
## 139 0.00031365    152   0.13638 0.37564 0.015450
## 140 0.00031258    153   0.13607 0.37564 0.015448
## 141 0.00030788    154   0.13576 0.37534 0.015452
## 142 0.00029802    155   0.13545 0.37502 0.015446
## 143 0.00029594    156   0.13515 0.37554 0.015443
## 144 0.00027781    158   0.13456 0.37574 0.015442
## 145 0.00027672    159   0.13428 0.37562 0.015436
## 146 0.00027287    160   0.13400 0.37534 0.015436
## 147 0.00027246    161   0.13373 0.37583 0.015438
## 148 0.00027010    162   0.13346 0.37583 0.015438
## 149 0.00025144    163   0.13319 0.37617 0.015439
## 150 0.00024464    164   0.13294 0.37599 0.015447
## 151 0.00024056    165   0.13269 0.37561 0.015441
## 152 0.00023660    166   0.13245 0.37567 0.015442
## 153 0.00023198    167   0.13221 0.37581 0.015445
## 154 0.00023124    168   0.13198 0.37582 0.015445
## 155 0.00022249    169   0.13175 0.37621 0.015476
## 156 0.00021870    170   0.13153 0.37658 0.015475
## 157 0.00020670    173   0.13087 0.37678 0.015476
## 158 0.00020481    174   0.13067 0.37716 0.015474
## 159 0.00020284    175   0.13046 0.37697 0.015474
## 160 0.00020023    176   0.13026 0.37705 0.015474
## 161 0.00019943    177   0.13006 0.37705 0.015474
## 162 0.00019887    178   0.12986 0.37713 0.015473
## 163 0.00019599    179   0.12966 0.37696 0.015473
## 164 0.00019341    180   0.12946 0.37696 0.015473
## 165 0.00019139    182   0.12908 0.37694 0.015472
## 166 0.00018900    183   0.12889 0.37703 0.015468
## 167 0.00018848    184   0.12870 0.37696 0.015468
## 168 0.00018555    185   0.12851 0.37696 0.015468
## 169 0.00018454    186   0.12832 0.37660 0.015465
## 170 0.00018155    187   0.12814 0.37663 0.015465
## 171 0.00017521    188   0.12796 0.37713 0.015468
## 172 0.00016557    189   0.12778 0.37697 0.015467
## 173 0.00015708    190   0.12762 0.37744 0.015467
## 174 0.00015606    191   0.12746 0.37737 0.015468
## 175 0.00014319    192   0.12730 0.37794 0.015476
## 176 0.00013881    194   0.12702 0.37833 0.015476
## 177 0.00013062    195   0.12688 0.37871 0.015474
## 178 0.00013040    196   0.12675 0.37903 0.015473
## 179 0.00012943    197   0.12662 0.37898 0.015472
## 180 0.00012822    198   0.12649 0.37890 0.015472
## 181 0.00012368    199   0.12636 0.37899 0.015473
## 182 0.00012341    200   0.12624 0.37903 0.015473
## 183 0.00010871    201   0.12611 0.37919 0.015476
## 184 0.00010297    202   0.12600 0.37954 0.015475
## 185 0.00010000    203   0.12590 0.37978 0.015475
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
```

```
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC08
##  [6] HC03_VC09   HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129
## [11] HC03_VC13.x HC03_VC13.y HC03_VC130  HC03_VC131  HC03_VC132
## [16] HC03_VC133  HC03_VC134  HC03_VC14   HC03_VC15   HC03_VC156
## [21] HC03_VC17   HC03_VC18   HC03_VC75   HC03_VC76   HC03_VC77
## [26] HC03_VC78   HC03_VC79   HC03_VC80   HC03_VC81   HC03_VC82
## [31] HC03_VC83   HC03_VC84   HC03_VC85   HC03_VC86   HC03_VC87
## [36] HC03_VC88   HC03_VC89   HC03_VC90   HC03_VC91   Latitude
## [41] Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##             CP nsplit rel error  xerror     xstd
## 1   0.26894188      0   1.00000 1.00098 0.031338
## 2   0.15300146      1   0.73106 0.73858 0.023654
## 3   0.04905506      2   0.57806 0.59007 0.021072
## 4   0.03483358      3   0.52900 0.54442 0.020147
## 5   0.02575390      4   0.49417 0.51565 0.019528
## 6   0.02255952      5   0.46841 0.51013 0.019315
## 7   0.02195363      6   0.44585 0.49704 0.018760
## 8   0.01812404      7   0.42390 0.49106 0.018329
## 9   0.01460201      8   0.40578 0.47962 0.018020
## 10  0.01373019      9   0.39117 0.47186 0.018194
## 11  0.01318438     10   0.37744 0.46590 0.018004
## 12  0.01113299     11   0.36426 0.45023 0.017765
## 13  0.01044189     12   0.35313 0.44112 0.017537
## 14  0.00876037     13   0.34269 0.43251 0.017122
## 15  0.00771485     14   0.33393 0.42497 0.016794
## 16  0.00737522     15   0.32621 0.41461 0.016156
## 17  0.00651898     16   0.31884 0.41097 0.016121
## 18  0.00609098     17   0.31232 0.41483 0.016740
## 19  0.00582690     18   0.30623 0.41510 0.016719
## 20  0.00523612     19   0.30040 0.41369 0.016688
## 21  0.00482466     20   0.29516 0.41064 0.016571
## 22  0.00479629     21   0.29034 0.40951 0.016741
## 23  0.00455377     22   0.28554 0.40815 0.016688
## 24  0.00406023     23   0.28099 0.40742 0.016828
## 25  0.00390075     24   0.27693 0.40181 0.016641
## 26  0.00386309     25   0.27303 0.40074 0.016614
## 27  0.00373690     26   0.26916 0.39568 0.016166
## 28  0.00323021     27   0.26543 0.39376 0.016105
## 29  0.00287222     28   0.26220 0.39062 0.016089
## 30  0.00276227     29   0.25932 0.39411 0.016199
## 31  0.00273625     30   0.25656 0.39431 0.016198
## 32  0.00269698     31   0.25383 0.39312 0.016164
## 33  0.00265792     32   0.25113 0.39238 0.016130
## 34  0.00261475     33   0.24847 0.39238 0.016130
## 35  0.00259402     34   0.24586 0.39238 0.016130
## 36  0.00242374     35   0.24326 0.39109 0.016082
## 37  0.00234022     36   0.24084 0.38771 0.015916
```

```
## 38  0.00226835     37   0.23850 0.38761 0.016034
## 39  0.00217154     39   0.23396 0.38544 0.015918
## 40  0.00203158     40   0.23179 0.38352 0.015974
## 41  0.00200903     41   0.22976 0.38255 0.016043
## 42  0.00197756     43   0.22574 0.38255 0.016046
## 43  0.00192518     44   0.22376 0.38399 0.016129
## 44  0.00180554     45   0.22184 0.38242 0.015993
## 45  0.00174488     47   0.21823 0.38238 0.015988
## 46  0.00171649     48   0.21648 0.38259 0.015749
## 47  0.00171513     49   0.21476 0.38325 0.015761
## 48  0.00163250     50   0.21305 0.38208 0.015706
## 49  0.00160365     51   0.21142 0.38033 0.015676
## 50  0.00154749     52   0.20981 0.38100 0.015701
## 51  0.00145486     53   0.20827 0.38232 0.015750
## 52  0.00145420     54   0.20681 0.37885 0.015567
## 53  0.00142485     55   0.20536 0.37846 0.015565
## 54  0.00140643     56   0.20393 0.37809 0.015589
## 55  0.00139441     58   0.20112 0.37787 0.015588
## 56  0.00136615     59   0.19972 0.37638 0.015568
## 57  0.00134432     62   0.19563 0.37603 0.015571
## 58  0.00132904     63   0.19428 0.37585 0.015570
## 59  0.00132240     64   0.19295 0.37526 0.015574
## 60  0.00123928     65   0.19163 0.37343 0.015254
## 61  0.00123217     66   0.19039 0.37367 0.015260
## 62  0.00116417     67   0.18916 0.37365 0.015305
## 63  0.00115728     68   0.18799 0.37505 0.015321
## 64  0.00111347     70   0.18568 0.37451 0.015302
## 65  0.00107149     71   0.18457 0.37406 0.015345
## 66  0.00106199     72   0.18350 0.37361 0.015321
## 67  0.00099296     73   0.18243 0.37273 0.015338
## 68  0.00097298     74   0.18144 0.37366 0.015334
## 69  0.00096571     75   0.18047 0.37388 0.015338
## 70  0.00094299     76   0.17950 0.37429 0.015408
## 71  0.00092604     77   0.17856 0.37393 0.015402
## 72  0.00086791     78   0.17763 0.37429 0.015446
## 73  0.00086082     80   0.17590 0.37551 0.015423
## 74  0.00083882     81   0.17504 0.37597 0.015448
## 75  0.00082185     82   0.17420 0.37605 0.015443
## 76  0.00081999     83   0.17338 0.37479 0.015399
## 77  0.00081510     84   0.17256 0.37454 0.015381
## 78  0.00080718     85   0.17174 0.37465 0.015385
## 79  0.00080253     86   0.17093 0.37471 0.015392
## 80  0.00080214     87   0.17013 0.37501 0.015393
## 81  0.00079999     88   0.16933 0.37515 0.015393
## 82  0.00077675     89   0.16853 0.37537 0.015392
## 83  0.00076692     90   0.16775 0.37450 0.015345
## 84  0.00075579     91   0.16698 0.37333 0.015315
## 85  0.00074858     92   0.16623 0.37369 0.015325
## 86  0.00070380     94   0.16473 0.37251 0.015291
## 87  0.00069992     95   0.16403 0.37289 0.015369
## 88  0.00069223     96   0.16333 0.37286 0.015382
## 89  0.00068902     97   0.16264 0.37359 0.015382
## 90  0.00068832     98   0.16195 0.37352 0.015383
## 91  0.00066915     99   0.16126 0.37321 0.015382
```

```
## 92  0.00066580    100   0.16059 0.37289 0.015364
## 93  0.00063716    103   0.15859 0.37277 0.015368
## 94  0.00063355    104   0.15795 0.37276 0.015292
## 95  0.00061813    105   0.15732 0.37396 0.015336
## 96  0.00061596    107   0.15609 0.37507 0.015394
## 97  0.00061410    108   0.15547 0.37542 0.015411
## 98  0.00060509    109   0.15486 0.37538 0.015413
## 99  0.00059366    110   0.15425 0.37507 0.015409
## 100 0.00057957    111   0.15366 0.37528 0.015444
## 101 0.00057526    112   0.15308 0.37555 0.015468
## 102 0.00055524    114   0.15193 0.37660 0.015497
## 103 0.00055083    115   0.15137 0.37684 0.015572
## 104 0.00054831    116   0.15082 0.37718 0.015577
## 105 0.00054719    117   0.15027 0.37729 0.015578
## 106 0.00049059    118   0.14972 0.37568 0.015534
## 107 0.00048324    119   0.14923 0.37599 0.015554
## 108 0.00048039    120   0.14875 0.37628 0.015558
## 109 0.00046824    121   0.14827 0.37707 0.015566
## 110 0.00046728    122   0.14780 0.37745 0.015574
## 111 0.00046207    123   0.14733 0.37783 0.015578
## 112 0.00044998    124   0.14687 0.37828 0.015592
## 113 0.00043664    125   0.14642 0.37793 0.015586
## 114 0.00043135    126   0.14599 0.37792 0.015588
## 115 0.00042858    127   0.14555 0.37818 0.015589
## 116 0.00041923    128   0.14513 0.37763 0.015562
## 117 0.00041219    129   0.14471 0.37765 0.015564
## 118 0.00040882    130   0.14429 0.37858 0.015569
## 119 0.00040711    131   0.14389 0.37859 0.015571
## 120 0.00040614    132   0.14348 0.37859 0.015571
## 121 0.00039895    133   0.14307 0.37867 0.015572
## 122 0.00038747    134   0.14267 0.37857 0.015577
## 123 0.00038480    135   0.14229 0.37832 0.015576
## 124 0.00038424    136   0.14190 0.37832 0.015576
## 125 0.00037809    137   0.14152 0.37833 0.015575
## 126 0.00037584    138   0.14114 0.37838 0.015576
## 127 0.00036870    140   0.14039 0.37875 0.015573
## 128 0.00034819    141   0.14002 0.37867 0.015567
## 129 0.00034302    142   0.13967 0.37992 0.015652
## 130 0.00033810    143   0.13933 0.38007 0.015615
## 131 0.00033694    144   0.13899 0.38016 0.015615
## 132 0.00033661    145   0.13865 0.37993 0.015613
## 133 0.00033647    146   0.13832 0.38008 0.015612
## 134 0.00032613    147   0.13798 0.38064 0.015617
## 135 0.00031906    148   0.13765 0.38110 0.015617
## 136 0.00031893    149   0.13733 0.38169 0.015620
## 137 0.00031861    150   0.13702 0.38162 0.015620
## 138 0.00031509    151   0.13670 0.38095 0.015605
## 139 0.00031365    152   0.13638 0.38102 0.015604
## 140 0.00031258    153   0.13607 0.38109 0.015604
## 141 0.00030788    154   0.13576 0.38086 0.015600
## 142 0.00029802    155   0.13545 0.38048 0.015598
## 143 0.00029594    156   0.13515 0.38015 0.015602
## 144 0.00027781    158   0.13456 0.38045 0.015612
## 145 0.00027672    159   0.13428 0.38094 0.015611
```

```
## 146 0.00027287    160   0.13400 0.38091 0.015611
## 147 0.00027246    161   0.13373 0.38137 0.015614
## 148 0.00027010    162   0.13346 0.38148 0.015615
## 149 0.00025144    163   0.13319 0.38222 0.015611
## 150 0.00024464    164   0.13294 0.38246 0.015629
## 151 0.00024056    165   0.13269 0.38263 0.015629
## 152 0.00023660    166   0.13245 0.38234 0.015629
## 153 0.00023198    167   0.13221 0.38279 0.015627
## 154 0.00023124    168   0.13198 0.38265 0.015628
## 155 0.00022249    169   0.13175 0.38253 0.015627
## 156 0.00021870    170   0.13153 0.38294 0.015631
## 157 0.00020670    173   0.13087 0.38286 0.015631
## 158 0.00020481    174   0.13067 0.38283 0.015635
## 159 0.00020284    175   0.13046 0.38294 0.015636
## 160 0.00020023    176   0.13026 0.38282 0.015635
## 161 0.00020000    177   0.13006 0.38282 0.015635
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC09
##  [6] HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.x
## [11] HC03_VC13.y HC03_VC130  HC03_VC132  HC03_VC133  HC03_VC134
## [16] HC03_VC15   HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC75
## [21] HC03_VC76   HC03_VC77   HC03_VC78   HC03_VC79   HC03_VC80
## [26] HC03_VC81   HC03_VC82   HC03_VC83   HC03_VC84   HC03_VC85
## [31] HC03_VC86   HC03_VC87   HC03_VC88   HC03_VC89   HC03_VC90
## [36] HC03_VC91   Latitude    Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##             CP nsplit rel error  xerror     xstd
## 1   0.26894188      0   1.00000 1.00026 0.031318
## 2   0.15300146      1   0.73106 0.73888 0.023588
## 3   0.04905506      2   0.57806 0.59374 0.021080
## 4   0.03483358      3   0.52900 0.54859 0.020155
## 5   0.02575390      4   0.49417 0.52550 0.019636
## 6   0.02255952      5   0.46841 0.51639 0.018787
## 7   0.02195363      6   0.44585 0.50911 0.018490
## 8   0.01812404      7   0.42390 0.49956 0.017997
## 9   0.01460201      8   0.40578 0.48108 0.017992
## 10  0.01373019      9   0.39117 0.47519 0.017947
## 11  0.01318438     10   0.37744 0.47101 0.017855
## 12  0.01113299     11   0.36426 0.46301 0.018096
## 13  0.01044189     12   0.35313 0.45632 0.017870
## 14  0.00876037     13   0.34269 0.45121 0.017705
## 15  0.00771485     14   0.33393 0.44417 0.017576
## 16  0.00737522     15   0.32621 0.43935 0.017202
## 17  0.00651898     16   0.31884 0.43323 0.017039
## 18  0.00609098     17   0.31232 0.43266 0.017048
```

```
## 19  0.00582690     18   0.30623 0.43512 0.017688
## 20  0.00523612     19   0.30040 0.42747 0.017460
## 21  0.00482466     20   0.29516 0.42395 0.017551
## 22  0.00479629     21   0.29034 0.42108 0.017581
## 23  0.00455377     22   0.28554 0.41800 0.017484
## 24  0.00406023     23   0.28099 0.41567 0.017349
## 25  0.00390075     24   0.27693 0.41353 0.017420
## 26  0.00386309     25   0.27303 0.41022 0.017151
## 27  0.00373690     26   0.26916 0.40768 0.017024
## 28  0.00323021     27   0.26543 0.40509 0.016955
## 29  0.00287222     28   0.26220 0.39924 0.016904
## 30  0.00276227     29   0.25932 0.39480 0.016561
## 31  0.00273625     30   0.25656 0.39218 0.016484
## 32  0.00269698     31   0.25383 0.39218 0.016484
## 33  0.00265792     32   0.25113 0.38891 0.015943
## 34  0.00261475     33   0.24847 0.38720 0.015900
## 35  0.00259402     34   0.24586 0.38650 0.015770
## 36  0.00242374     35   0.24326 0.38741 0.015888
## 37  0.00234022     36   0.24084 0.38835 0.015890
## 38  0.00226835     37   0.23850 0.38888 0.015940
## 39  0.00217154     39   0.23396 0.38951 0.015920
## 40  0.00203158     40   0.23179 0.39240 0.016011
## 41  0.00200903     41   0.22976 0.39187 0.016000
## 42  0.00197756     43   0.22574 0.39056 0.015887
## 43  0.00192518     44   0.22376 0.38930 0.015872
## 44  0.00180554     45   0.22184 0.39076 0.015921
## 45  0.00174488     47   0.21823 0.38766 0.015996
## 46  0.00171649     48   0.21648 0.38755 0.016060
## 47  0.00171513     49   0.21476 0.38804 0.016115
## 48  0.00163250     50   0.21305 0.38837 0.016145
## 49  0.00160365     51   0.21142 0.38799 0.016155
## 50  0.00154749     52   0.20981 0.39153 0.016299
## 51  0.00145486     53   0.20827 0.38734 0.015688
## 52  0.00145420     54   0.20681 0.38529 0.015684
## 53  0.00142485     55   0.20536 0.38495 0.015670
## 54  0.00140643     56   0.20393 0.38569 0.015724
## 55  0.00139441     58   0.20112 0.38585 0.015722
## 56  0.00136615     59   0.19972 0.38514 0.015720
## 57  0.00134432     62   0.19563 0.38457 0.015735
## 58  0.00132904     63   0.19428 0.38420 0.015702
## 59  0.00132240     64   0.19295 0.38420 0.015702
## 60  0.00123928     65   0.19163 0.38429 0.015740
## 61  0.00123217     66   0.19039 0.38354 0.015739
## 62  0.00116417     67   0.18916 0.38459 0.015767
## 63  0.00115728     68   0.18799 0.38487 0.015795
## 64  0.00111347     70   0.18568 0.38530 0.015800
## 65  0.00107149     71   0.18457 0.38817 0.016039
## 66  0.00106199     72   0.18350 0.38765 0.016030
## 67  0.00099296     73   0.18243 0.38738 0.016055
## 68  0.00097298     74   0.18144 0.38809 0.016133
## 69  0.00096571     75   0.18047 0.38799 0.016130
## 70  0.00094299     76   0.17950 0.38842 0.016129
## 71  0.00092604     77   0.17856 0.39098 0.016427
## 72  0.00086791     78   0.17763 0.38891 0.016409
```

```
## 73  0.00086082     80   0.17590 0.38574 0.016251
## 74  0.00083882     81   0.17504 0.38536 0.016258
## 75  0.00082185     82   0.17420 0.38493 0.016320
## 76  0.00081999     83   0.17338 0.38452 0.016321
## 77  0.00081510     84   0.17256 0.38363 0.016318
## 78  0.00080718     85   0.17174 0.38364 0.016319
## 79  0.00080253     86   0.17093 0.38315 0.016318
## 80  0.00080214     87   0.17013 0.38295 0.016304
## 81  0.00079999     88   0.16933 0.38230 0.016307
## 82  0.00077675     89   0.16853 0.38409 0.016346
## 83  0.00076692     90   0.16775 0.38353 0.016300
## 84  0.00075579     91   0.16698 0.38345 0.016304
## 85  0.00074858     92   0.16623 0.38376 0.016300
## 86  0.00070380     94   0.16473 0.38406 0.016326
## 87  0.00069992     95   0.16403 0.38428 0.016334
## 88  0.00069223     96   0.16333 0.38439 0.016328
## 89  0.00068902     97   0.16264 0.38430 0.016334
## 90  0.00068832     98   0.16195 0.38430 0.016334
## 91  0.00066915     99   0.16126 0.38498 0.016343
## 92  0.00066580    100   0.16059 0.38595 0.016387
## 93  0.00063716    103   0.15859 0.38728 0.016415
## 94  0.00063355    104   0.15795 0.38650 0.016435
## 95  0.00061813    105   0.15732 0.38686 0.016433
## 96  0.00061596    107   0.15609 0.38694 0.016430
## 97  0.00061410    108   0.15547 0.38675 0.016429
## 98  0.00060509    109   0.15486 0.38594 0.016432
## 99  0.00059366    110   0.15425 0.38699 0.016423
## 100 0.00057957    111   0.15366 0.38705 0.016426
## 101 0.00057526    112   0.15308 0.38741 0.016431
## 102 0.00055524    114   0.15193 0.38785 0.016444
## 103 0.00055083    115   0.15137 0.38742 0.016449
## 104 0.00054831    116   0.15082 0.38747 0.016448
## 105 0.00054719    117   0.15027 0.38753 0.016448
## 106 0.00049059    118   0.14972 0.38559 0.016355
## 107 0.00048324    119   0.14923 0.38634 0.016376
## 108 0.00048039    120   0.14875 0.38531 0.016358
## 109 0.00046824    121   0.14827 0.38523 0.016357
## 110 0.00046728    122   0.14780 0.38551 0.016367
## 111 0.00046207    123   0.14733 0.38566 0.016375
## 112 0.00044998    124   0.14687 0.38566 0.016383
## 113 0.00043664    125   0.14642 0.38706 0.016405
## 114 0.00043135    126   0.14599 0.38749 0.016404
## 115 0.00042858    127   0.14555 0.38757 0.016404
## 116 0.00041923    128   0.14513 0.38792 0.016402
## 117 0.00041219    129   0.14471 0.38720 0.016383
## 118 0.00040882    130   0.14429 0.38743 0.016386
## 119 0.00040711    131   0.14389 0.38737 0.016387
## 120 0.00040614    132   0.14348 0.38737 0.016387
## 121 0.00039895    133   0.14307 0.38780 0.016394
## 122 0.00038747    134   0.14267 0.38779 0.016375
## 123 0.00038480    135   0.14229 0.38793 0.016377
## 124 0.00038424    136   0.14190 0.38791 0.016376
## 125 0.00037809    137   0.14152 0.38795 0.016376
## 126 0.00037584    138   0.14114 0.38840 0.016374
```

```
## 127 0.00036870    140   0.14039 0.38921 0.016374
## 128 0.00034819    141   0.14002 0.38941 0.016368
## 129 0.00034302    142   0.13967 0.38924 0.016359
## 130 0.00033810    143   0.13933 0.38925 0.016358
## 131 0.00033694    144   0.13899 0.38916 0.016353
## 132 0.00033661    145   0.13865 0.38916 0.016353
## 133 0.00033647    146   0.13832 0.38916 0.016353
## 134 0.00032613    147   0.13798 0.38945 0.016365
## 135 0.00031906    148   0.13765 0.38924 0.016365
## 136 0.00031893    149   0.13733 0.38934 0.016363
## 137 0.00031861    150   0.13702 0.38934 0.016363
## 138 0.00031509    151   0.13670 0.38968 0.016371
## 139 0.00031365    152   0.13638 0.38956 0.016371
## 140 0.00031258    153   0.13607 0.38956 0.016371
## 141 0.00030788    154   0.13576 0.38962 0.016371
## 142 0.00030000    155   0.13545 0.38974 0.016370
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC09
##  [6] HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.x
## [11] HC03_VC13.y HC03_VC130  HC03_VC132  HC03_VC133  HC03_VC134
## [16] HC03_VC15   HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC75
## [21] HC03_VC76   HC03_VC79   HC03_VC80   HC03_VC81   HC03_VC82
## [26] HC03_VC83   HC03_VC84   HC03_VC85   HC03_VC86   HC03_VC87
## [31] HC03_VC88   HC03_VC89   HC03_VC90   HC03_VC91   Latitude
## [36] Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##              CP nsplit rel error  xerror     xstd
## 1   0.26894188      0   1.00000 1.00155 0.031361
## 2   0.15300146      1   0.73106 0.74155 0.023679
## 3   0.04905506      2   0.57806 0.59104 0.020994
## 4   0.03483358      3   0.52900 0.54520 0.020071
## 5   0.02575390      4   0.49417 0.51179 0.019595
## 6   0.02255952      5   0.46841 0.49739 0.018899
## 7   0.02195363      6   0.44585 0.49222 0.018343
## 8   0.01812404      7   0.42390 0.48633 0.018351
## 9   0.01460201      8   0.40578 0.47610 0.018274
## 10  0.01373019      9   0.39117 0.46353 0.017995
## 11  0.01318438     10   0.37744 0.46010 0.017916
## 12  0.01113299     11   0.36426 0.44495 0.017486
## 13  0.01044189     12   0.35313 0.42978 0.017668
## 14  0.00876037     13   0.34269 0.42365 0.017534
## 15  0.00771485     14   0.33393 0.42257 0.017363
## 16  0.00737522     15   0.32621 0.42177 0.017322
## 17  0.00651898     16   0.31884 0.42070 0.017387
## 18  0.00609098     17   0.31232 0.41403 0.016834
```

```
## 19  0.00582690     18   0.30623 0.41106 0.016709
## 20  0.00523612     19   0.30040 0.40988 0.016625
## 21  0.00482466     20   0.29516 0.41006 0.016587
## 22  0.00479629     21   0.29034 0.40727 0.016730
## 23  0.00455377     22   0.28554 0.40446 0.016698
## 24  0.00406023     23   0.28099 0.40026 0.016621
## 25  0.00390075     24   0.27693 0.39499 0.016603
## 26  0.00386309     25   0.27303 0.39462 0.016608
## 27  0.00373690     26   0.26916 0.38832 0.016202
## 28  0.00323021     27   0.26543 0.38951 0.016353
## 29  0.00287222     28   0.26220 0.38492 0.016080
## 30  0.00276227     29   0.25932 0.38388 0.016046
## 31  0.00273625     30   0.25656 0.38376 0.016047
## 32  0.00269698     31   0.25383 0.38389 0.016046
## 33  0.00265792     32   0.25113 0.38452 0.016094
## 34  0.00261475     33   0.24847 0.38361 0.016104
## 35  0.00259402     34   0.24586 0.38176 0.016063
## 36  0.00242374     35   0.24326 0.38063 0.016063
## 37  0.00234022     36   0.24084 0.37873 0.015939
## 38  0.00226835     37   0.23850 0.37934 0.015950
## 39  0.00217154     39   0.23396 0.37383 0.015559
## 40  0.00203158     40   0.23179 0.37251 0.015387
## 41  0.00200903     41   0.22976 0.37512 0.015594
## 42  0.00197756     43   0.22574 0.37646 0.015670
## 43  0.00192518     44   0.22376 0.37600 0.015725
## 44  0.00180554     45   0.22184 0.37429 0.015873
## 45  0.00174488     47   0.21823 0.37565 0.015992
## 46  0.00171649     48   0.21648 0.37151 0.015684
## 47  0.00171513     49   0.21476 0.37179 0.015683
## 48  0.00163250     50   0.21305 0.37065 0.015695
## 49  0.00160365     51   0.21142 0.36948 0.015467
## 50  0.00154749     52   0.20981 0.37041 0.015522
## 51  0.00145486     53   0.20827 0.36787 0.015443
## 52  0.00145420     54   0.20681 0.36606 0.015393
## 53  0.00142485     55   0.20536 0.36644 0.015395
## 54  0.00140643     56   0.20393 0.36792 0.015466
## 55  0.00139441     58   0.20112 0.36678 0.015426
## 56  0.00136615     59   0.19972 0.36640 0.015429
## 57  0.00134432     62   0.19563 0.36467 0.015415
## 58  0.00132904     63   0.19428 0.36558 0.015487
## 59  0.00132240     64   0.19295 0.36575 0.015521
## 60  0.00123928     65   0.19163 0.36524 0.015520
## 61  0.00123217     66   0.19039 0.36561 0.015526
## 62  0.00116417     67   0.18916 0.36426 0.015513
## 63  0.00115728     68   0.18799 0.36339 0.015500
## 64  0.00111347     70   0.18568 0.36386 0.015444
## 65  0.00107149     71   0.18457 0.36175 0.015405
## 66  0.00106199     72   0.18350 0.36313 0.015427
## 67  0.00099296     73   0.18243 0.36348 0.015547
## 68  0.00097298     74   0.18144 0.36297 0.015602
## 69  0.00096571     75   0.18047 0.36317 0.015635
## 70  0.00094299     76   0.17950 0.36297 0.015633
## 71  0.00092604     77   0.17856 0.36322 0.015640
## 72  0.00086791     78   0.17763 0.36160 0.015593
```

```
## 73  0.00086082     80   0.17590 0.35887 0.015618
## 74  0.00083882     81   0.17504 0.35908 0.015615
## 75  0.00082185     82   0.17420 0.35993 0.015535
## 76  0.00081999     83   0.17338 0.36040 0.015559
## 77  0.00081510     84   0.17256 0.36095 0.015596
## 78  0.00080718     85   0.17174 0.35999 0.015584
## 79  0.00080253     86   0.17093 0.36031 0.015580
## 80  0.00080214     87   0.17013 0.36003 0.015571
## 81  0.00079999     88   0.16933 0.35980 0.015571
## 82  0.00077675     89   0.16853 0.36091 0.015585
## 83  0.00076692     90   0.16775 0.36205 0.015637
## 84  0.00075579     91   0.16698 0.36116 0.015377
## 85  0.00074858     92   0.16623 0.36154 0.015362
## 86  0.00070380     94   0.16473 0.36112 0.015311
## 87  0.00069992     95   0.16403 0.36169 0.015322
## 88  0.00069223     96   0.16333 0.36185 0.015321
## 89  0.00068902     97   0.16264 0.36171 0.015319
## 90  0.00068832     98   0.16195 0.36145 0.015319
## 91  0.00066915     99   0.16126 0.36229 0.015338
## 92  0.00066580    100   0.16059 0.36314 0.015356
## 93  0.00063716    103   0.15859 0.36353 0.015362
## 94  0.00063355    104   0.15795 0.36428 0.015395
## 95  0.00061813    105   0.15732 0.36459 0.015393
## 96  0.00061596    107   0.15609 0.36416 0.015381
## 97  0.00061410    108   0.15547 0.36438 0.015380
## 98  0.00060509    109   0.15486 0.36422 0.015380
## 99  0.00059366    110   0.15425 0.36459 0.015414
## 100 0.00057957    111   0.15366 0.36502 0.015417
## 101 0.00057526    112   0.15308 0.36640 0.015426
## 102 0.00055524    114   0.15193 0.36723 0.015432
## 103 0.00055083    115   0.15137 0.36710 0.015427
## 104 0.00054831    116   0.15082 0.36723 0.015428
## 105 0.00054719    117   0.15027 0.36723 0.015428
## 106 0.00049059    118   0.14972 0.36719 0.015377
## 107 0.00048324    119   0.14923 0.36782 0.015381
## 108 0.00048039    120   0.14875 0.36780 0.015380
## 109 0.00046824    121   0.14827 0.36671 0.015269
## 110 0.00046728    122   0.14780 0.36672 0.015277
## 111 0.00046207    123   0.14733 0.36663 0.015277
## 112 0.00044998    124   0.14687 0.36603 0.015275
## 113 0.00043664    125   0.14642 0.36586 0.015264
## 114 0.00043135    126   0.14599 0.36628 0.015555
## 115 0.00042858    127   0.14555 0.36626 0.015556
## 116 0.00041923    128   0.14513 0.36637 0.015565
## 117 0.00041219    129   0.14471 0.36654 0.015610
## 118 0.00040882    130   0.14429 0.36612 0.015606
## 119 0.00040711    131   0.14389 0.36649 0.015621
## 120 0.00040614    132   0.14348 0.36668 0.015618
## 121 0.00040000    133   0.14307 0.36772 0.015687
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
```

```
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC09
##  [6] HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.y
## [11] HC03_VC130  HC03_VC132  HC03_VC133  HC03_VC134  HC03_VC15
## [16] HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC75   HC03_VC76
## [21] HC03_VC79   HC03_VC82   HC03_VC83   HC03_VC84   HC03_VC85
## [26] HC03_VC86   HC03_VC87   HC03_VC89   HC03_VC90   HC03_VC91
## [31] Latitude    Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##             CP nsplit rel error  xerror     xstd
## 1   0.26894188      0   1.00000 1.00180 0.031369
## 2   0.15300146      1   0.73106 0.74176 0.023924
## 3   0.04905506      2   0.57806 0.60123 0.021652
## 4   0.03483358      3   0.52900 0.55611 0.020830
## 5   0.02575390      4   0.49417 0.53240 0.020343
## 6   0.02255952      5   0.46841 0.51769 0.019450
## 7   0.02195363      6   0.44585 0.50936 0.018819
## 8   0.01812404      7   0.42390 0.50542 0.018670
## 9   0.01460201      8   0.40578 0.49427 0.018466
## 10  0.01373019      9   0.39117 0.48279 0.018634
## 11  0.01318438     10   0.37744 0.47520 0.018415
## 12  0.01113299     11   0.36426 0.46022 0.018035
## 13  0.01044189     12   0.35313 0.45204 0.018129
## 14  0.00876037     13   0.34269 0.45075 0.018258
## 15  0.00771485     14   0.33393 0.44708 0.018553
## 16  0.00737522     15   0.32621 0.44371 0.018336
## 17  0.00651898     16   0.31884 0.44509 0.018425
## 18  0.00609098     17   0.31232 0.44015 0.018303
## 19  0.00582690     18   0.30623 0.43784 0.018761
## 20  0.00523612     19   0.30040 0.43040 0.018717
## 21  0.00482466     20   0.29516 0.43121 0.018479
## 22  0.00479629     21   0.29034 0.42938 0.018443
## 23  0.00455377     22   0.28554 0.42706 0.018382
## 24  0.00406023     23   0.28099 0.43046 0.018613
## 25  0.00390075     24   0.27693 0.42810 0.018556
## 26  0.00386309     25   0.27303 0.42560 0.018554
## 27  0.00373690     26   0.26916 0.42564 0.018555
## 28  0.00323021     27   0.26543 0.41830 0.018025
## 29  0.00287222     28   0.26220 0.40850 0.017612
## 30  0.00276227     29   0.25932 0.40673 0.017558
## 31  0.00273625     30   0.25656 0.40541 0.017531
## 32  0.00269698     31   0.25383 0.40510 0.017554
## 33  0.00265792     32   0.25113 0.40411 0.017483
## 34  0.00261475     33   0.24847 0.40506 0.017534
## 35  0.00259402     34   0.24586 0.40519 0.017524
## 36  0.00242374     35   0.24326 0.40686 0.017587
## 37  0.00234022     36   0.24084 0.40466 0.018218
## 38  0.00226835     37   0.23850 0.40289 0.018153
## 39  0.00217154     39   0.23396 0.40023 0.018028
## 40  0.00203158     40   0.23179 0.40011 0.018070
```

```
## 41  0.00200903     41   0.22976 0.40111 0.018096
## 42  0.00197756     43   0.22574 0.40108 0.018092
## 43  0.00192518     44   0.22376 0.39951 0.018115
## 44  0.00180554     45   0.22184 0.39863 0.018094
## 45  0.00174488     47   0.21823 0.39759 0.018098
## 46  0.00171649     48   0.21648 0.39574 0.018072
## 47  0.00171513     49   0.21476 0.39598 0.018123
## 48  0.00163250     50   0.21305 0.39669 0.018183
## 49  0.00160365     51   0.21142 0.39728 0.018152
## 50  0.00154749     52   0.20981 0.39802 0.018302
## 51  0.00145486     53   0.20827 0.39683 0.018223
## 52  0.00145420     54   0.20681 0.39421 0.018211
## 53  0.00142485     55   0.20536 0.39411 0.018214
## 54  0.00140643     56   0.20393 0.39463 0.018220
## 55  0.00139441     58   0.20112 0.39319 0.018178
## 56  0.00136615     59   0.19972 0.39006 0.017657
## 57  0.00134432     62   0.19563 0.38978 0.017630
## 58  0.00132904     63   0.19428 0.38983 0.017630
## 59  0.00132240     64   0.19295 0.39001 0.017672
## 60  0.00123928     65   0.19163 0.38856 0.017614
## 61  0.00123217     66   0.19039 0.38571 0.017396
## 62  0.00116417     67   0.18916 0.38592 0.017406
## 63  0.00115728     68   0.18799 0.38773 0.017483
## 64  0.00111347     70   0.18568 0.38740 0.017473
## 65  0.00107149     71   0.18457 0.38703 0.017500
## 66  0.00106199     72   0.18350 0.38709 0.017485
## 67  0.00099296     73   0.18243 0.38600 0.017442
## 68  0.00097298     74   0.18144 0.38581 0.017404
## 69  0.00096571     75   0.18047 0.38572 0.017404
## 70  0.00094299     76   0.17950 0.38631 0.017394
## 71  0.00092604     77   0.17856 0.38698 0.017400
## 72  0.00086791     78   0.17763 0.38639 0.017416
## 73  0.00086082     80   0.17590 0.38654 0.017431
## 74  0.00083882     81   0.17504 0.38655 0.017436
## 75  0.00082185     82   0.17420 0.38862 0.017460
## 76  0.00081999     83   0.17338 0.38794 0.017453
## 77  0.00081510     84   0.17256 0.38774 0.017454
## 78  0.00080718     85   0.17174 0.38781 0.017437
## 79  0.00080253     86   0.17093 0.38747 0.017439
## 80  0.00080214     87   0.17013 0.38790 0.017444
## 81  0.00079999     88   0.16933 0.38781 0.017444
## 82  0.00077675     89   0.16853 0.38729 0.017439
## 83  0.00076692     90   0.16775 0.38804 0.017466
## 84  0.00075579     91   0.16698 0.38891 0.017500
## 85  0.00074858     92   0.16623 0.38906 0.017493
## 86  0.00070380     94   0.16473 0.39040 0.017516
## 87  0.00069992     95   0.16403 0.39086 0.017502
## 88  0.00069223     96   0.16333 0.39052 0.017503
## 89  0.00068902     97   0.16264 0.39006 0.017486
## 90  0.00068832     98   0.16195 0.39025 0.017484
## 91  0.00066915     99   0.16126 0.39068 0.017488
## 92  0.00066580    100   0.16059 0.38920 0.017439
## 93  0.00063716    103   0.15859 0.38895 0.017463
## 94  0.00063355    104   0.15795 0.38857 0.017453
```

```
## 95  0.00061813    105   0.15732 0.38861 0.017487
## 96  0.00061596    107   0.15609 0.38888 0.017484
## 97  0.00061410    108   0.15547 0.38863 0.017483
## 98  0.00060509    109   0.15486 0.38846 0.017462
## 99  0.00059366    110   0.15425 0.38743 0.017464
## 100 0.00057957    111   0.15366 0.38884 0.017477
## 101 0.00057526    112   0.15308 0.38885 0.017474
## 102 0.00055524    114   0.15193 0.38955 0.017458
## 103 0.00055083    115   0.15137 0.38929 0.017452
## 104 0.00054831    116   0.15082 0.38987 0.017479
## 105 0.00054719    117   0.15027 0.38983 0.017479
## 106 0.00050000    118   0.14972 0.39096 0.017495
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC09
##  [6] HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.y
## [11] HC03_VC130  HC03_VC132  HC03_VC133  HC03_VC134  HC03_VC15
## [16] HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC75   HC03_VC76
## [21] HC03_VC79   HC03_VC82   HC03_VC83   HC03_VC84   HC03_VC85
## [26] HC03_VC86   HC03_VC87   HC03_VC89   HC03_VC90   HC03_VC91
## [31] Latitude    Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##             CP nsplit rel error  xerror     xstd
## 1  0.26894188      0   1.00000 1.00036 0.031323
## 2  0.15300146      1   0.73106 0.74209 0.023963
## 3  0.04905506      2   0.57806 0.59321 0.021511
## 4  0.03483358      3   0.52900 0.55061 0.020632
## 5  0.02575390      4   0.49417 0.52761 0.020155
## 6  0.02255952      5   0.46841 0.50854 0.018410
## 7  0.02195363      6   0.44585 0.50697 0.018360
## 8  0.01812404      7   0.42390 0.50124 0.018194
## 9  0.01460201      8   0.40578 0.48594 0.018214
## 10 0.01373019      9   0.39117 0.47474 0.018158
## 11 0.01318438     10   0.37744 0.46953 0.018007
## 12 0.01113299     11   0.36426 0.45804 0.017656
## 13 0.01044189     12   0.35313 0.44732 0.017499
## 14 0.00876037     13   0.34269 0.43172 0.016869
## 15 0.00771485     14   0.33393 0.42361 0.016594
## 16 0.00737522     15   0.32621 0.41623 0.016236
## 17 0.00651898     16   0.31884 0.41348 0.016085
## 18 0.00609098     17   0.31232 0.40844 0.015578
## 19 0.00582690     18   0.30623 0.40841 0.015578
## 20 0.00523612     19   0.30040 0.40606 0.015561
## 21 0.00482466     20   0.29516 0.40678 0.015475
## 22 0.00479629     21   0.29034 0.39928 0.015079
## 23 0.00455377     22   0.28554 0.39890 0.015069
```

```
## 24 0.00406023     23   0.28099 0.39426 0.015164
## 25 0.00390075     24   0.27693 0.39292 0.015140
## 26 0.00386309     25   0.27303 0.39233 0.015190
## 27 0.00373690     26   0.26916 0.39052 0.015113
## 28 0.00323021     27   0.26543 0.38603 0.015108
## 29 0.00287222     28   0.26220 0.38450 0.015049
## 30 0.00276227     29   0.25932 0.38295 0.015485
## 31 0.00273625     30   0.25656 0.38008 0.015471
## 32 0.00269698     31   0.25383 0.38013 0.015487
## 33 0.00265792     32   0.25113 0.38061 0.015492
## 34 0.00261475     33   0.24847 0.38135 0.015584
## 35 0.00259402     34   0.24586 0.38165 0.015716
## 36 0.00242374     35   0.24326 0.38212 0.016001
## 37 0.00234022     36   0.24084 0.37944 0.015967
## 38 0.00226835     37   0.23850 0.37868 0.015992
## 39 0.00217154     39   0.23396 0.37914 0.015971
## 40 0.00203158     40   0.23179 0.37785 0.015960
## 41 0.00200903     41   0.22976 0.37682 0.015783
## 42 0.00197756     43   0.22574 0.37799 0.015810
## 43 0.00192518     44   0.22376 0.37689 0.015803
## 44 0.00180554     45   0.22184 0.37626 0.015869
## 45 0.00174488     47   0.21823 0.37498 0.015889
## 46 0.00171649     48   0.21648 0.37412 0.015948
## 47 0.00171513     49   0.21476 0.37410 0.015947
## 48 0.00163250     50   0.21305 0.37572 0.016108
## 49 0.00160365     51   0.21142 0.37792 0.016161
## 50 0.00154749     52   0.20981 0.37746 0.016099
## 51 0.00145486     53   0.20827 0.37819 0.016054
## 52 0.00145420     54   0.20681 0.37839 0.016107
## 53 0.00142485     55   0.20536 0.38045 0.016158
## 54 0.00140643     56   0.20393 0.38052 0.016166
## 55 0.00139441     58   0.20112 0.37975 0.016153
## 56 0.00136615     59   0.19972 0.37999 0.016150
## 57 0.00134432     62   0.19563 0.37977 0.016228
## 58 0.00132904     63   0.19428 0.37916 0.016187
## 59 0.00132240     64   0.19295 0.37916 0.016187
## 60 0.00123928     65   0.19163 0.37849 0.016091
## 61 0.00123217     66   0.19039 0.37881 0.016083
## 62 0.00116417     67   0.18916 0.38131 0.016104
## 63 0.00115728     68   0.18799 0.38035 0.016102
## 64 0.00111347     70   0.18568 0.38119 0.016206
## 65 0.00107149     71   0.18457 0.38149 0.016237
## 66 0.00106199     72   0.18350 0.38317 0.016278
## 67 0.00099296     73   0.18243 0.38373 0.016346
## 68 0.00097298     74   0.18144 0.38472 0.016342
## 69 0.00096571     75   0.18047 0.38392 0.016334
## 70 0.00094299     76   0.17950 0.38477 0.016337
## 71 0.00092604     77   0.17856 0.38535 0.016335
## 72 0.00086791     78   0.17763 0.38574 0.016336
## 73 0.00086082     80   0.17590 0.38583 0.016328
## 74 0.00083882     81   0.17504 0.38594 0.016354
## 75 0.00082185     82   0.17420 0.38537 0.016329
## 76 0.00081999     83   0.17338 0.38628 0.016341
## 77 0.00081510     84   0.17256 0.38651 0.016367
```

```
## 78 0.00080718     85   0.17174 0.38675 0.016374
## 79 0.00080253     86   0.17093 0.38718 0.016370
## 80 0.00080214     87   0.17013 0.38617 0.016286
## 81 0.00079999     88   0.16933 0.38617 0.016286
## 82 0.00077675     89   0.16853 0.38706 0.016305
## 83 0.00076692     90   0.16775 0.38619 0.016286
## 84 0.00075579     91   0.16698 0.38632 0.016289
## 85 0.00074858     92   0.16623 0.38641 0.016296
## 86 0.00070380     94   0.16473 0.38583 0.016253
## 87 0.00069992     95   0.16403 0.38489 0.016224
## 88 0.00069223     96   0.16333 0.38428 0.016235
## 89 0.00068902     97   0.16264 0.38397 0.016230
## 90 0.00068832     98   0.16195 0.38393 0.016231
## 91 0.00066915     99   0.16126 0.38457 0.016249
## 92 0.00066580    100   0.16059 0.38384 0.016239
## 93 0.00063716    103   0.15859 0.38352 0.016250
## 94 0.00063355    104   0.15795 0.38356 0.016254
## 95 0.00061813    105   0.15732 0.38470 0.016468
## 96 0.00061596    107   0.15609 0.38465 0.016479
## 97 0.00061410    108   0.15547 0.38496 0.016478
## 98 0.00060509    109   0.15486 0.38527 0.016493
## 99 0.00060000    110   0.15425 0.38498 0.016475
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC09
##  [6] HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.y
## [11] HC03_VC130  HC03_VC132  HC03_VC133  HC03_VC134  HC03_VC156
## [16] HC03_VC17   HC03_VC18   HC03_VC76   HC03_VC79   HC03_VC82
## [21] HC03_VC83   HC03_VC84   HC03_VC85   HC03_VC86   HC03_VC87
## [26] HC03_VC89   HC03_VC90   HC03_VC91   Latitude    Longitude
## [31] State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##            CP nsplit rel error  xerror     xstd
## 1  0.26894188      0   1.00000 1.00174 0.031379
## 2  0.15300146      1   0.73106 0.73790 0.023692
## 3  0.04905506      2   0.57806 0.58845 0.021003
## 4  0.03483358      3   0.52900 0.54058 0.020064
## 5  0.02575390      4   0.49417 0.53661 0.020482
## 6  0.02255952      5   0.46841 0.52597 0.020172
## 7  0.02195363      6   0.44585 0.52312 0.020073
## 8  0.01812404      7   0.42390 0.51667 0.019658
## 9  0.01460201      8   0.40578 0.48858 0.019163
## 10 0.01373019      9   0.39117 0.47236 0.018786
## 11 0.01318438     10   0.37744 0.46268 0.018625
## 12 0.01113299     11   0.36426 0.46317 0.018893
## 13 0.01044189     12   0.35313 0.45268 0.018681
```

```
## 14 0.00876037     13   0.34269 0.44645 0.018356
## 15 0.00771485     14   0.33393 0.44070 0.018214
## 16 0.00737522     15   0.32621 0.43540 0.017799
## 17 0.00651898     16   0.31884 0.42657 0.017095
## 18 0.00609098     17   0.31232 0.41812 0.016514
## 19 0.00582690     18   0.30623 0.41745 0.016728
## 20 0.00523612     19   0.30040 0.41403 0.016553
## 21 0.00482466     20   0.29516 0.41427 0.016592
## 22 0.00479629     21   0.29034 0.41377 0.016595
## 23 0.00455377     22   0.28554 0.41363 0.016639
## 24 0.00406023     23   0.28099 0.40941 0.016453
## 25 0.00390075     24   0.27693 0.40296 0.016151
## 26 0.00386309     25   0.27303 0.40175 0.016076
## 27 0.00373690     26   0.26916 0.40036 0.016050
## 28 0.00323021     27   0.26543 0.39671 0.015814
## 29 0.00287222     28   0.26220 0.39250 0.015546
## 30 0.00276227     29   0.25932 0.38749 0.015543
## 31 0.00273625     30   0.25656 0.38669 0.015592
## 32 0.00269698     31   0.25383 0.38628 0.015635
## 33 0.00265792     32   0.25113 0.38691 0.015652
## 34 0.00261475     33   0.24847 0.38681 0.015646
## 35 0.00259402     34   0.24586 0.38711 0.015692
## 36 0.00242374     35   0.24326 0.38332 0.015182
## 37 0.00234022     36   0.24084 0.38068 0.015115
## 38 0.00226835     37   0.23850 0.37972 0.015047
## 39 0.00217154     39   0.23396 0.38042 0.015037
## 40 0.00203158     40   0.23179 0.38143 0.015068
## 41 0.00200903     41   0.22976 0.37865 0.015009
## 42 0.00197756     43   0.22574 0.37780 0.014956
## 43 0.00192518     44   0.22376 0.37779 0.014968
## 44 0.00180554     45   0.22184 0.37769 0.015010
## 45 0.00174488     47   0.21823 0.37680 0.015000
## 46 0.00171649     48   0.21648 0.37509 0.014956
## 47 0.00171513     49   0.21476 0.37448 0.014956
## 48 0.00163250     50   0.21305 0.37390 0.014951
## 49 0.00160365     51   0.21142 0.37231 0.014916
## 50 0.00154749     52   0.20981 0.37145 0.014888
## 51 0.00145486     53   0.20827 0.37115 0.014918
## 52 0.00145420     54   0.20681 0.37085 0.014944
## 53 0.00142485     55   0.20536 0.37076 0.014943
## 54 0.00140643     56   0.20393 0.37216 0.015021
## 55 0.00139441     58   0.20112 0.37259 0.015027
## 56 0.00136615     59   0.19972 0.37289 0.015036
## 57 0.00134432     62   0.19563 0.37258 0.015017
## 58 0.00132904     63   0.19428 0.37166 0.014950
## 59 0.00132240     64   0.19295 0.37257 0.014972
## 60 0.00123928     65   0.19163 0.37331 0.015011
## 61 0.00123217     66   0.19039 0.37255 0.014998
## 62 0.00116417     67   0.18916 0.37006 0.014975
## 63 0.00115728     68   0.18799 0.36999 0.014982
## 64 0.00111347     70   0.18568 0.37040 0.014982
## 65 0.00107149     71   0.18457 0.36888 0.014906
## 66 0.00106199     72   0.18350 0.36907 0.014945
## 67 0.00099296     73   0.18243 0.37039 0.014953
```

```
## 68 0.00097298     74   0.18144 0.37082 0.014963
## 69 0.00096571     75   0.18047 0.36993 0.014969
## 70 0.00094299     76   0.17950 0.36979 0.014954
## 71 0.00092604     77   0.17856 0.37016 0.015006
## 72 0.00086791     78   0.17763 0.37145 0.015109
## 73 0.00086082     80   0.17590 0.37209 0.015131
## 74 0.00083882     81   0.17504 0.37109 0.015143
## 75 0.00082185     82   0.17420 0.37111 0.015097
## 76 0.00081999     83   0.17338 0.37114 0.015107
## 77 0.00081510     84   0.17256 0.37104 0.015108
## 78 0.00080718     85   0.17174 0.37142 0.015071
## 79 0.00080253     86   0.17093 0.37169 0.015070
## 80 0.00080214     87   0.17013 0.37169 0.015140
## 81 0.00079999     88   0.16933 0.37178 0.015139
## 82 0.00077675     89   0.16853 0.37209 0.015159
## 83 0.00076692     90   0.16775 0.37272 0.015178
## 84 0.00075579     91   0.16698 0.37319 0.015201
## 85 0.00074858     92   0.16623 0.37306 0.015177
## 86 0.00070380     94   0.16473 0.37191 0.015250
## 87 0.00070000     95   0.16403 0.37243 0.015238
## 
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)], 
##     method = "anova", control = rpart.control(cp = cps[j]))
## 
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC10  
##  [6] HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.y HC03_VC130 
## [11] HC03_VC133  HC03_VC134  HC03_VC156  HC03_VC17   HC03_VC18  
## [16] HC03_VC79   HC03_VC82   HC03_VC83   HC03_VC84   HC03_VC85  
## [21] HC03_VC86   HC03_VC87   HC03_VC89   HC03_VC90   HC03_VC91  
## [26] Latitude    Longitude   State      
## 
## Root node error: 67.293/2547 = 0.02642
## 
## n= 2547 
## 
##             CP nsplit rel error  xerror     xstd
## 1  0.26894188      0   1.00000 1.00257 0.031368
## 2  0.15300146      1   0.73106 0.74095 0.023726
## 3  0.04905506      2   0.57806 0.59824 0.021515
## 4  0.03483358      3   0.52900 0.55089 0.020667
## 5  0.02575390      4   0.49417 0.52010 0.019395
## 6  0.02255952      5   0.46841 0.49866 0.018540
## 7  0.02195363      6   0.44585 0.48792 0.017866
## 8  0.01812404      7   0.42390 0.48447 0.017924
## 9  0.01460201      8   0.40578 0.47573 0.017806
## 10 0.01373019      9   0.39117 0.46621 0.017805
## 11 0.01318438     10   0.37744 0.45954 0.017679
## 12 0.01113299     11   0.36426 0.45061 0.017285
## 13 0.01044189     12   0.35313 0.43447 0.017183
## 14 0.00876037     13   0.34269 0.42867 0.017352
## 15 0.00771485     14   0.33393 0.42533 0.017060
## 16 0.00737522     15   0.32621 0.42173 0.016900
```

```
## 17 0.00651898     16   0.31884 0.42172 0.016887
## 18 0.00609098     17   0.31232 0.41766 0.016771
## 19 0.00582690     18   0.30623 0.41111 0.016323
## 20 0.00523612     19   0.30040 0.41170 0.016421
## 21 0.00482466     20   0.29516 0.41004 0.016887
## 22 0.00479629     21   0.29034 0.40955 0.016920
## 23 0.00455377     22   0.28554 0.40894 0.016933
## 24 0.00406023     23   0.28099 0.40240 0.016819
## 25 0.00390075     24   0.27693 0.39921 0.016561
## 26 0.00386309     25   0.27303 0.39605 0.016152
## 27 0.00373690     26   0.26916 0.40181 0.016566
## 28 0.00323021     27   0.26543 0.40228 0.016622
## 29 0.00287222     28   0.26220 0.39671 0.016433
## 30 0.00276227     29   0.25932 0.39447 0.016457
## 31 0.00273625     30   0.25656 0.39496 0.016469
## 32 0.00269698     31   0.25383 0.39679 0.016722
## 33 0.00265792     32   0.25113 0.39748 0.016767
## 34 0.00261475     33   0.24847 0.39767 0.016779
## 35 0.00259402     34   0.24586 0.39797 0.016775
## 36 0.00242374     35   0.24326 0.39418 0.016595
## 37 0.00234022     36   0.24084 0.39217 0.016582
## 38 0.00226835     37   0.23850 0.39396 0.016680
## 39 0.00217154     39   0.23396 0.39443 0.016645
## 40 0.00203158     40   0.23179 0.39212 0.016616
## 41 0.00200903     41   0.22976 0.39143 0.016713
## 42 0.00197756     43   0.22574 0.39080 0.016706
## 43 0.00192518     44   0.22376 0.39276 0.016736
## 44 0.00180554     45   0.22184 0.39064 0.016705
## 45 0.00174488     47   0.21823 0.38883 0.016572
## 46 0.00171649     48   0.21648 0.38921 0.016635
## 47 0.00171513     49   0.21476 0.38927 0.016629
## 48 0.00163250     50   0.21305 0.39045 0.016631
## 49 0.00160365     51   0.21142 0.39135 0.016682
## 50 0.00154749     52   0.20981 0.39023 0.016682
## 51 0.00145486     53   0.20827 0.39049 0.016713
## 52 0.00145420     54   0.20681 0.38908 0.016586
## 53 0.00142485     55   0.20536 0.38872 0.016581
## 54 0.00140643     56   0.20393 0.38773 0.016575
## 55 0.00139441     58   0.20112 0.38859 0.016648
## 56 0.00136615     59   0.19972 0.38746 0.016600
## 57 0.00134432     62   0.19563 0.38711 0.016592
## 58 0.00132904     63   0.19428 0.38653 0.016476
## 59 0.00132240     64   0.19295 0.38700 0.016467
## 60 0.00123928     65   0.19163 0.38724 0.016449
## 61 0.00123217     66   0.19039 0.38780 0.016578
## 62 0.00116417     67   0.18916 0.38865 0.016617
## 63 0.00115728     68   0.18799 0.38803 0.016647
## 64 0.00111347     70   0.18568 0.38743 0.016617
## 65 0.00107149     71   0.18457 0.38750 0.016655
## 66 0.00106199     72   0.18350 0.38848 0.016677
## 67 0.00099296     73   0.18243 0.38780 0.016689
## 68 0.00097298     74   0.18144 0.38955 0.016716
## 69 0.00096571     75   0.18047 0.39117 0.016747
## 70 0.00094299     76   0.17950 0.39021 0.016686
```

```
## 71 0.00092604     77   0.17856 0.38971 0.016641
## 72 0.00086791     78   0.17763 0.38902 0.016649
## 73 0.00086082     80   0.17590 0.38913 0.016692
## 74 0.00083882     81   0.17504 0.38903 0.016695
## 75 0.00082185     82   0.17420 0.38895 0.016710
## 76 0.00081999     83   0.17338 0.38829 0.016700
## 77 0.00081510     84   0.17256 0.38829 0.016700
## 78 0.00080718     85   0.17174 0.38842 0.016712
## 79 0.00080253     86   0.17093 0.38945 0.016731
## 80 0.00080214     87   0.17013 0.38937 0.016731
## 81 0.00080000     88   0.16933 0.38937 0.016731
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC10
##  [6] HC03_VC11   HC03_VC12   HC03_VC13.y HC03_VC130  HC03_VC133
## [11] HC03_VC134  HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC79
## [16] HC03_VC82   HC03_VC83   HC03_VC85   HC03_VC86   HC03_VC87
## [21] HC03_VC89   HC03_VC90   HC03_VC91   Latitude    Longitude
## [26] State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##            CP nsplit rel error  xerror     xstd
## 1  0.26894188      0   1.00000 1.00096 0.031319
## 2  0.15300146      1   0.73106 0.74146 0.023909
## 3  0.04905506      2   0.57806 0.59841 0.021574
## 4  0.03483358      3   0.52900 0.55429 0.020681
## 5  0.02575390      4   0.49417 0.53329 0.020643
## 6  0.02255952      5   0.46841 0.52226 0.020115
## 7  0.02195363      6   0.44585 0.50053 0.018383
## 8  0.01812404      7   0.42390 0.49486 0.018044
## 9  0.01460201      8   0.40578 0.47986 0.017778
## 10 0.01373019      9   0.39117 0.47636 0.017775
## 11 0.01318438     10   0.37744 0.47565 0.017793
## 12 0.01113299     11   0.36426 0.45447 0.017575
## 13 0.01044189     12   0.35313 0.44361 0.017387
## 14 0.00876037     13   0.34269 0.43818 0.017300
## 15 0.00771485     14   0.33393 0.42520 0.017156
## 16 0.00737522     15   0.32621 0.42066 0.016854
## 17 0.00651898     16   0.31884 0.41760 0.016643
## 18 0.00609098     17   0.31232 0.41644 0.016783
## 19 0.00582690     18   0.30623 0.41578 0.016840
## 20 0.00523612     19   0.30040 0.41463 0.016811
## 21 0.00482466     20   0.29516 0.40618 0.016910
## 22 0.00479629     21   0.29034 0.40089 0.016237
## 23 0.00455377     22   0.28554 0.40076 0.016327
## 24 0.00406023     23   0.28099 0.39844 0.016408
## 25 0.00390075     24   0.27693 0.39892 0.016421
```

```
## 26 0.00386309     25   0.27303 0.39892 0.016421
## 27 0.00373690     26   0.26916 0.39756 0.016402
## 28 0.00323021     27   0.26543 0.39274 0.016049
## 29 0.00287222     28   0.26220 0.38289 0.015693
## 30 0.00276227     29   0.25932 0.37711 0.015574
## 31 0.00273625     30   0.25656 0.37903 0.015647
## 32 0.00269698     31   0.25383 0.37964 0.015640
## 33 0.00265792     32   0.25113 0.37873 0.015644
## 34 0.00261475     33   0.24847 0.37583 0.015517
## 35 0.00259402     34   0.24586 0.37549 0.015535
## 36 0.00242374     35   0.24326 0.37569 0.015474
## 37 0.00234022     36   0.24084 0.37645 0.015588
## 38 0.00226835     37   0.23850 0.37799 0.015640
## 39 0.00217154     39   0.23396 0.37868 0.015620
## 40 0.00203158     40   0.23179 0.37634 0.015660
## 41 0.00200903     41   0.22976 0.37730 0.015661
## 42 0.00197756     43   0.22574 0.37584 0.015582
## 43 0.00192518     44   0.22376 0.37753 0.015598
## 44 0.00180554     45   0.22184 0.37340 0.015321
## 45 0.00174488     47   0.21823 0.37335 0.015317
## 46 0.00171649     48   0.21648 0.37180 0.015305
## 47 0.00171513     49   0.21476 0.37024 0.015292
## 48 0.00163250     50   0.21305 0.37045 0.015370
## 49 0.00160365     51   0.21142 0.36819 0.015350
## 50 0.00154749     52   0.20981 0.36878 0.015315
## 51 0.00145486     53   0.20827 0.36503 0.015216
## 52 0.00145420     54   0.20681 0.36555 0.015274
## 53 0.00142485     55   0.20536 0.36643 0.015398
## 54 0.00140643     56   0.20393 0.36604 0.015402
## 55 0.00139441     58   0.20112 0.36611 0.015416
## 56 0.00136615     59   0.19972 0.36508 0.015416
## 57 0.00134432     62   0.19563 0.36565 0.015424
## 58 0.00132904     63   0.19428 0.36662 0.015426
## 59 0.00132240     64   0.19295 0.36559 0.015380
## 60 0.00123928     65   0.19163 0.36351 0.015295
## 61 0.00123217     66   0.19039 0.36316 0.015292
## 62 0.00116417     67   0.18916 0.36223 0.015319
## 63 0.00115728     68   0.18799 0.36249 0.015306
## 64 0.00111347     70   0.18568 0.36273 0.015385
## 65 0.00107149     71   0.18457 0.36143 0.015325
## 66 0.00106199     72   0.18350 0.36150 0.015342
## 67 0.00099296     73   0.18243 0.36182 0.015338
## 68 0.00097298     74   0.18144 0.36382 0.015573
## 69 0.00096571     75   0.18047 0.36411 0.015583
## 70 0.00094299     76   0.17950 0.36418 0.015611
## 71 0.00092604     77   0.17856 0.36601 0.015699
## 72 0.00090000     78   0.17763 0.36614 0.015696
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC05   HC03_VC07   HC03_VC10   HC03_VC11
```

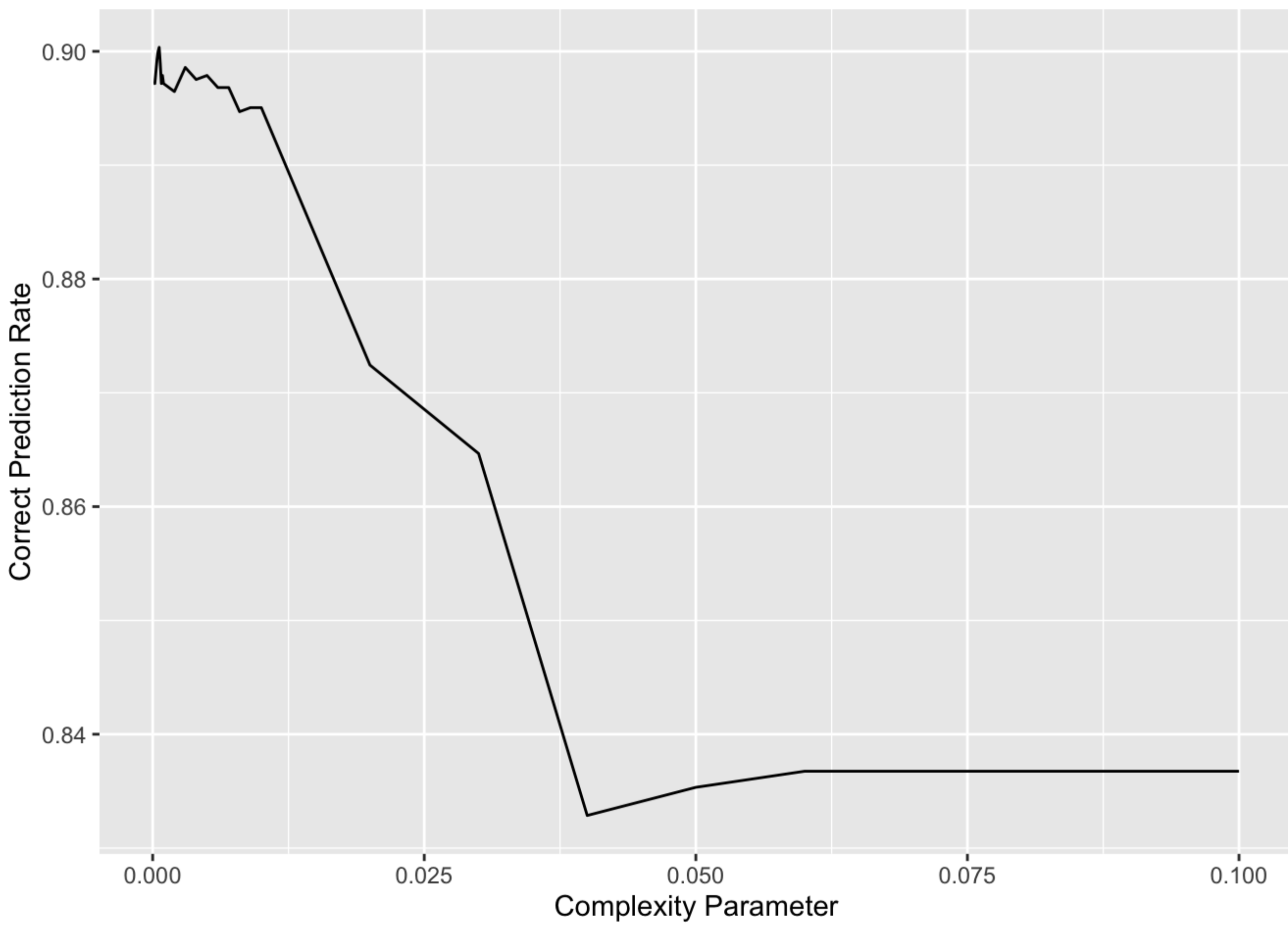
```
##  [6] HC03_VC12   HC03_VC13.y HC03_VC130  HC03_VC133  HC03_VC134
## [11] HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC79   HC03_VC82
## [16] HC03_VC83   HC03_VC85   HC03_VC86   HC03_VC87   HC03_VC89
## [21] HC03_VC90   HC03_VC91   Latitude    Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##           CP nsplit rel error  xerror     xstd
## 1  0.2689419      0   1.00000 1.00061 0.031308
## 2  0.1530015      1   0.73106 0.74017 0.023628
## 3  0.0490551      2   0.57806 0.59738 0.021119
## 4  0.0348336      3   0.52900 0.54261 0.020042
## 5  0.0257539      4   0.49417 0.51566 0.019329
## 6  0.0225595      5   0.46841 0.49506 0.018605
## 7  0.0219536      6   0.44585 0.48501 0.017841
## 8  0.0181240      7   0.42390 0.48042 0.017698
## 9  0.0146020      8   0.40578 0.45473 0.016749
## 10 0.0137302      9   0.39117 0.43944 0.016572
## 11 0.0131844     10   0.37744 0.43534 0.016529
## 12 0.0111330     11   0.36426 0.44227 0.017600
## 13 0.0104419     12   0.35313 0.43206 0.017502
## 14 0.0087604     13   0.34269 0.42979 0.017353
## 15 0.0077149     14   0.33393 0.42231 0.017378
## 16 0.0073752     15   0.32621 0.42185 0.017313
## 17 0.0065190     16   0.31884 0.41651 0.017153
## 18 0.0060910     17   0.31232 0.41601 0.017214
## 19 0.0058269     18   0.30623 0.41513 0.017258
## 20 0.0052361     19   0.30040 0.41075 0.017134
## 21 0.0048247     20   0.29516 0.40718 0.017089
## 22 0.0047963     21   0.29034 0.40379 0.017086
## 23 0.0045538     22   0.28554 0.40593 0.017100
## 24 0.0040602     23   0.28099 0.40729 0.017391
## 25 0.0039007     24   0.27693 0.40525 0.017370
## 26 0.0038631     25   0.27303 0.40547 0.017577
## 27 0.0037369     26   0.26916 0.40583 0.017582
## 28 0.0032302     27   0.26543 0.40219 0.017279
## 29 0.0028722     28   0.26220 0.39982 0.017260
## 30 0.0027623     29   0.25932 0.39424 0.016782
## 31 0.0027362     30   0.25656 0.39071 0.016330
## 32 0.0026970     31   0.25383 0.38969 0.016304
## 33 0.0026579     32   0.25113 0.38889 0.016289
## 34 0.0026147     33   0.24847 0.38883 0.016287
## 35 0.0025940     34   0.24586 0.38928 0.016343
## 36 0.0024237     35   0.24326 0.38973 0.016367
## 37 0.0023402     36   0.24084 0.38725 0.016288
## 38 0.0022683     37   0.23850 0.38623 0.016271
## 39 0.0021715     39   0.23396 0.38638 0.016305
## 40 0.0020316     40   0.23179 0.38646 0.016381
## 41 0.0020090     41   0.22976 0.38252 0.016251
## 42 0.0019776     43   0.22574 0.38251 0.016293
## 43 0.0019252     44   0.22376 0.38237 0.016304
## 44 0.0018055     45   0.22184 0.37929 0.016259
```

```
## 45 0.0017449     47   0.21823 0.38235 0.016430
## 46 0.0017165     48   0.21648 0.38275 0.016446
## 47 0.0017151     49   0.21476 0.38151 0.016430
## 48 0.0016325     50   0.21305 0.38254 0.016543
## 49 0.0016037     51   0.21142 0.38269 0.016574
## 50 0.0015475     52   0.20981 0.38367 0.016599
## 51 0.0014549     53   0.20827 0.38038 0.016447
## 52 0.0014542     54   0.20681 0.37897 0.016433
## 53 0.0014249     55   0.20536 0.37784 0.016416
## 54 0.0014064     56   0.20393 0.37854 0.016394
## 55 0.0013944     58   0.20112 0.37796 0.016398
## 56 0.0013662     59   0.19972 0.37794 0.016402
## 57 0.0013443     62   0.19563 0.37665 0.016361
## 58 0.0013290     63   0.19428 0.37470 0.016359
## 59 0.0013224     64   0.19295 0.37523 0.016348
## 60 0.0012393     65   0.19163 0.37489 0.016367
## 61 0.0012322     66   0.19039 0.37599 0.016428
## 62 0.0011642     67   0.18916 0.37623 0.016418
## 63 0.0011573     68   0.18799 0.37600 0.016399
## 64 0.0011135     70   0.18568 0.37479 0.016264
## 65 0.0010715     71   0.18457 0.37331 0.016227
## 66 0.0010620     72   0.18350 0.37354 0.016255
## 67 0.0010000     73   0.18243 0.37435 0.016238
```

```
#build tree again with new cp range

cps2 = c(seq(0.0005,0.0015,by = .0001))

preds2 = matrix(nrow = nbuild, ncol = length(cps))
tree2 = list()

for (i in 1:10) {
  buildfold = folds[, -i]
  testfold = folds[, i]

  for (j in 1:length(cps)) {
    tree2[[j]] = rpart( repvotes ~ .,
            data = buildset[buildfold, -(3)],
            method = "anova",
            control = rpart.control(cp = cps[j]))
    preds2[testfold, j] =
      predict(tree2[[j]],
              newdata = buildset[testfold, -c(1,3)],
              type = "vector")
  }
}

cpr2 = apply(preds2, 2, function(oneSet) {
  return(sum(c(oneSet>.5)==c(buildset[,1]>.5))/nbuild)
})

which.max(cpr2)
```

```
## [1] 6
```

```
cprdf = data.frame(cps, cpr2)
ggplot(data = cprdf, aes(x = cps, y = cpr2)) +
  geom_line() +
  labs(x = "Complexity Parameter", y = "Correct Prediction Rate")
```




```
for(i in 1:10){
  printcp(tree2[[i]])
}
```

```
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC08
##  [6] HC03_VC09   HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129
## [11] HC03_VC13.x HC03_VC13.y HC03_VC130  HC03_VC131  HC03_VC132
## [16] HC03_VC133  HC03_VC134  HC03_VC14   HC03_VC15   HC03_VC156
## [21] HC03_VC17   HC03_VC18   HC03_VC75   HC03_VC76   HC03_VC77
## [26] HC03_VC78   HC03_VC79   HC03_VC80   HC03_VC81   HC03_VC82
## [31] HC03_VC83   HC03_VC84   HC03_VC85   HC03_VC86   HC03_VC87
```

```
## [36] HC03_VC88   HC03_VC89   HC03_VC90   HC03_VC91   Latitude
## [41] Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##              CP nsplit rel error  xerror     xstd
## 1   0.26894188      0   1.00000 1.00045 0.031321
## 2   0.15300146      1   0.73106 0.74194 0.023900
## 3   0.04905506      2   0.57806 0.59182 0.021475
## 4   0.03483358      3   0.52900 0.54587 0.020513
## 5   0.02575390      4   0.49417 0.52939 0.020089
## 6   0.02255952      5   0.46841 0.51754 0.019705
## 7   0.02195363      6   0.44585 0.50741 0.019094
## 8   0.01812404      7   0.42390 0.49686 0.018547
## 9   0.01460201      8   0.40578 0.48644 0.018362
## 10  0.01373019      9   0.39117 0.47268 0.017853
## 11  0.01318438     10   0.37744 0.46683 0.017838
## 12  0.01113299     11   0.36426 0.46119 0.017736
## 13  0.01044189     12   0.35313 0.45142 0.017328
## 14  0.00876037     13   0.34269 0.43675 0.017226
## 15  0.00771485     14   0.33393 0.42940 0.017001
## 16  0.00737522     15   0.32621 0.42019 0.016381
## 17  0.00651898     16   0.31884 0.41523 0.016162
## 18  0.00609098     17   0.31232 0.41323 0.016092
## 19  0.00582690     18   0.30623 0.41749 0.017110
## 20  0.00523612     19   0.30040 0.41884 0.017175
## 21  0.00482466     20   0.29516 0.41304 0.017024
## 22  0.00479629     21   0.29034 0.41325 0.016984
## 23  0.00455377     22   0.28554 0.40600 0.016227
## 24  0.00406023     23   0.28099 0.40981 0.016437
## 25  0.00390075     24   0.27693 0.41180 0.016568
## 26  0.00386309     25   0.27303 0.41204 0.016563
## 27  0.00373690     26   0.26916 0.41045 0.016570
## 28  0.00323021     27   0.26543 0.40817 0.016546
## 29  0.00287222     28   0.26220 0.40546 0.016624
## 30  0.00276227     29   0.25932 0.40436 0.016660
## 31  0.00273625     30   0.25656 0.40384 0.016657
## 32  0.00269698     31   0.25383 0.40445 0.016861
## 33  0.00265792     32   0.25113 0.40370 0.016843
## 34  0.00261475     33   0.24847 0.40133 0.016851
## 35  0.00259402     34   0.24586 0.40109 0.016854
## 36  0.00242374     35   0.24326 0.40057 0.016768
## 37  0.00234022     36   0.24084 0.40043 0.016752
## 38  0.00226835     37   0.23850 0.40201 0.016753
## 39  0.00217154     39   0.23396 0.40277 0.016829
## 40  0.00203158     40   0.23179 0.40128 0.016882
## 41  0.00200903     41   0.22976 0.39875 0.016744
## 42  0.00197756     43   0.22574 0.39904 0.016784
## 43  0.00192518     44   0.22376 0.39825 0.016782
## 44  0.00180554     45   0.22184 0.39996 0.016910
## 45  0.00174488     47   0.21823 0.40271 0.017091
## 46  0.00171649     48   0.21648 0.40324 0.017077
```

```
## 47  0.00171513     49   0.21476 0.40256 0.017078
## 48  0.00163250     50   0.21305 0.40217 0.016962
## 49  0.00160365     51   0.21142 0.40479 0.017059
## 50  0.00154749     52   0.20981 0.40226 0.016949
## 51  0.00145486     53   0.20827 0.39998 0.016864
## 52  0.00145420     54   0.20681 0.39891 0.016913
## 53  0.00142485     55   0.20536 0.39874 0.016867
## 54  0.00140643     56   0.20393 0.39928 0.016876
## 55  0.00139441     58   0.20112 0.39919 0.016868
## 56  0.00136615     59   0.19972 0.39928 0.016870
## 57  0.00134432     62   0.19563 0.39690 0.016809
## 58  0.00132904     63   0.19428 0.39670 0.016767
## 59  0.00132240     64   0.19295 0.39666 0.016772
## 60  0.00123928     65   0.19163 0.39370 0.016655
## 61  0.00123217     66   0.19039 0.39263 0.016670
## 62  0.00116417     67   0.18916 0.39282 0.016674
## 63  0.00115728     68   0.18799 0.39187 0.016625
## 64  0.00111347     70   0.18568 0.39081 0.016614
## 65  0.00107149     71   0.18457 0.38994 0.016592
## 66  0.00106199     72   0.18350 0.39092 0.016603
## 67  0.00099296     73   0.18243 0.39058 0.016623
## 68  0.00097298     74   0.18144 0.39108 0.016756
## 69  0.00096571     75   0.18047 0.39400 0.016890
## 70  0.00094299     76   0.17950 0.39342 0.016891
## 71  0.00092604     77   0.17856 0.39372 0.016864
## 72  0.00086791     78   0.17763 0.39264 0.016902
## 73  0.00086082     80   0.17590 0.39251 0.016898
## 74  0.00083882     81   0.17504 0.39231 0.016903
## 75  0.00082185     82   0.17420 0.39396 0.016909
## 76  0.00081999     83   0.17338 0.39356 0.016817
## 77  0.00081510     84   0.17256 0.39344 0.016818
## 78  0.00080718     85   0.17174 0.39345 0.016859
## 79  0.00080253     86   0.17093 0.39323 0.016853
## 80  0.00080214     87   0.17013 0.39332 0.016855
## 81  0.00079999     88   0.16933 0.39332 0.016855
## 82  0.00077675     89   0.16853 0.39407 0.016862
## 83  0.00076692     90   0.16775 0.39329 0.016835
## 84  0.00075579     91   0.16698 0.39368 0.016830
## 85  0.00074858     92   0.16623 0.39278 0.016811
## 86  0.00070380     94   0.16473 0.39476 0.016837
## 87  0.00069992     95   0.16403 0.39581 0.016836
## 88  0.00069223     96   0.16333 0.39550 0.016835
## 89  0.00068902     97   0.16264 0.39535 0.016845
## 90  0.00068832     98   0.16195 0.39547 0.016845
## 91  0.00066915     99   0.16126 0.39599 0.016850
## 92  0.00066580    100   0.16059 0.39558 0.016852
## 93  0.00063716    103   0.15859 0.39508 0.016844
## 94  0.00063355    104   0.15795 0.39392 0.016806
## 95  0.00061813    105   0.15732 0.39384 0.016810
## 96  0.00061596    107   0.15609 0.39389 0.016802
## 97  0.00061410    108   0.15547 0.39433 0.016838
## 98  0.00060509    109   0.15486 0.39393 0.016838
## 99  0.00059366    110   0.15425 0.39420 0.016837
## 100 0.00057957    111   0.15366 0.39433 0.016860
```

```
## 101 0.00057526    112   0.15308 0.39613 0.016922
## 102 0.00055524    114   0.15193 0.39527 0.016883
## 103 0.00055083    115   0.15137 0.39535 0.016892
## 104 0.00054831    116   0.15082 0.39533 0.016890
## 105 0.00054719    117   0.15027 0.39541 0.016889
## 106 0.00049059    118   0.14972 0.39553 0.016884
## 107 0.00048324    119   0.14923 0.39623 0.016949
## 108 0.00048039    120   0.14875 0.39527 0.016773
## 109 0.00046824    121   0.14827 0.39457 0.016764
## 110 0.00046728    122   0.14780 0.39409 0.016765
## 111 0.00046207    123   0.14733 0.39424 0.016766
## 112 0.00044998    124   0.14687 0.39313 0.016748
## 113 0.00043664    125   0.14642 0.39306 0.016738
## 114 0.00043135    126   0.14599 0.39312 0.016728
## 115 0.00042858    127   0.14555 0.39363 0.016736
## 116 0.00041923    128   0.14513 0.39345 0.016720
## 117 0.00041219    129   0.14471 0.39393 0.016718
## 118 0.00040882    130   0.14429 0.39412 0.016717
## 119 0.00040711    131   0.14389 0.39366 0.016710
## 120 0.00040614    132   0.14348 0.39351 0.016713
## 121 0.00039895    133   0.14307 0.39319 0.016709
## 122 0.00038747    134   0.14267 0.39438 0.016741
## 123 0.00038480    135   0.14229 0.39408 0.016740
## 124 0.00038424    136   0.14190 0.39401 0.016741
## 125 0.00037809    137   0.14152 0.39388 0.016735
## 126 0.00037584    138   0.14114 0.39386 0.016735
## 127 0.00036870    140   0.14039 0.39378 0.016736
## 128 0.00034819    141   0.14002 0.39400 0.016732
## 129 0.00034302    142   0.13967 0.39441 0.016756
## 130 0.00033810    143   0.13933 0.39403 0.016682
## 131 0.00033694    144   0.13899 0.39420 0.016681
## 132 0.00033661    145   0.13865 0.39409 0.016682
## 133 0.00033647    146   0.13832 0.39409 0.016682
## 134 0.00032613    147   0.13798 0.39462 0.016685
## 135 0.00031906    148   0.13765 0.39599 0.016696
## 136 0.00031893    149   0.13733 0.39626 0.016695
## 137 0.00031861    150   0.13702 0.39626 0.016695
## 138 0.00031509    151   0.13670 0.39629 0.016697
## 139 0.00031365    152   0.13638 0.39643 0.016697
## 140 0.00031258    153   0.13607 0.39680 0.016704
## 141 0.00030788    154   0.13576 0.39646 0.016702
## 142 0.00029802    155   0.13545 0.39601 0.016658
## 143 0.00029594    156   0.13515 0.39599 0.016659
## 144 0.00027781    158   0.13456 0.39602 0.016660
## 145 0.00027672    159   0.13428 0.39603 0.016661
## 146 0.00027287    160   0.13400 0.39636 0.016675
## 147 0.00027246    161   0.13373 0.39636 0.016675
## 148 0.00027010    162   0.13346 0.39630 0.016675
## 149 0.00025144    163   0.13319 0.39637 0.016673
## 150 0.00024464    164   0.13294 0.39705 0.016655
## 151 0.00024056    165   0.13269 0.39692 0.016652
## 152 0.00023660    166   0.13245 0.39726 0.016653
## 153 0.00023198    167   0.13221 0.39739 0.016653
## 154 0.00023124    168   0.13198 0.39758 0.016656
```

```
## 155 0.00022249    169   0.13175 0.39778 0.016664
## 156 0.00021870    170   0.13153 0.39803 0.016666
## 157 0.00020670    173   0.13087 0.39808 0.016667
## 158 0.00020481    174   0.13067 0.39765 0.016652
## 159 0.00020284    175   0.13046 0.39781 0.016658
## 160 0.00020023    176   0.13026 0.39765 0.016646
## 161 0.00019943    177   0.13006 0.39779 0.016645
## 162 0.00019887    178   0.12986 0.39782 0.016645
## 163 0.00019599    179   0.12966 0.39782 0.016645
## 164 0.00019341    180   0.12946 0.39781 0.016648
## 165 0.00019139    182   0.12908 0.39812 0.016657
## 166 0.00018900    183   0.12889 0.39812 0.016657
## 167 0.00018848    184   0.12870 0.39810 0.016657
## 168 0.00018555    185   0.12851 0.39811 0.016653
## 169 0.00018454    186   0.12832 0.39821 0.016654
## 170 0.00018155    187   0.12814 0.39744 0.016634
## 171 0.00017521    188   0.12796 0.39731 0.016632
## 172 0.00016557    189   0.12778 0.39769 0.016611
## 173 0.00015708    190   0.12762 0.39821 0.016617
## 174 0.00015606    191   0.12746 0.39833 0.016614
## 175 0.00014319    192   0.12730 0.39872 0.016618
## 176 0.00013881    194   0.12702 0.39887 0.016617
## 177 0.00013062    195   0.12688 0.39933 0.016621
## 178 0.00013040    196   0.12675 0.39924 0.016623
## 179 0.00012943    197   0.12662 0.39928 0.016623
## 180 0.00012822    198   0.12649 0.39928 0.016623
## 181 0.00012368    199   0.12636 0.39948 0.016622
## 182 0.00012341    200   0.12624 0.39951 0.016622
## 183 0.00010871    201   0.12611 0.39969 0.016622
## 184 0.00010297    202   0.12600 0.40000 0.016619
## 185 0.00010000    203   0.12590 0.39997 0.016618
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC08
##  [6] HC03_VC09   HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129
## [11] HC03_VC13.x HC03_VC13.y HC03_VC130  HC03_VC131  HC03_VC132
## [16] HC03_VC133  HC03_VC134  HC03_VC14   HC03_VC15   HC03_VC156
## [21] HC03_VC17   HC03_VC18   HC03_VC75   HC03_VC76   HC03_VC77
## [26] HC03_VC78   HC03_VC79   HC03_VC80   HC03_VC81   HC03_VC82
## [31] HC03_VC83   HC03_VC84   HC03_VC85   HC03_VC86   HC03_VC87
## [36] HC03_VC88   HC03_VC89   HC03_VC90   HC03_VC91   Latitude
## [41] Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##             CP nsplit rel error  xerror     xstd
## 1   0.26894188      0   1.00000 1.00124 0.031353
## 2   0.15300146      1   0.73106 0.74319 0.023934
```

```
## 3   0.04905506      2   0.57806 0.59506 0.021450
## 4   0.03483358      3   0.52900 0.54913 0.020519
## 5   0.02575390      4   0.49417 0.53287 0.020115
## 6   0.02255952      5   0.46841 0.52345 0.019590
## 7   0.02195363      6   0.44585 0.50381 0.018691
## 8   0.01812404      7   0.42390 0.50079 0.018779
## 9   0.01460201      8   0.40578 0.48614 0.018335
## 10  0.01373019      9   0.39117 0.47243 0.018373
## 11  0.01318438     10   0.37744 0.47090 0.018428
## 12  0.01113299     11   0.36426 0.46228 0.018262
## 13  0.01044189     12   0.35313 0.44786 0.018341
## 14  0.00876037     13   0.34269 0.43944 0.018209
## 15  0.00771485     14   0.33393 0.43521 0.018000
## 16  0.00737522     15   0.32621 0.43085 0.017762
## 17  0.00651898     16   0.31884 0.42604 0.017676
## 18  0.00609098     17   0.31232 0.42026 0.017512
## 19  0.00582690     18   0.30623 0.41524 0.017280
## 20  0.00523612     19   0.30040 0.41381 0.017253
## 21  0.00482466     20   0.29516 0.40642 0.016863
## 22  0.00479629     21   0.29034 0.39979 0.016749
## 23  0.00455377     22   0.28554 0.40142 0.016786
## 24  0.00406023     23   0.28099 0.39703 0.016489
## 25  0.00390075     24   0.27693 0.39186 0.016182
## 26  0.00386309     25   0.27303 0.38951 0.016084
## 27  0.00373690     26   0.26916 0.39036 0.016052
## 28  0.00323021     27   0.26543 0.38762 0.015891
## 29  0.00287222     28   0.26220 0.38386 0.016016
## 30  0.00276227     29   0.25932 0.38210 0.015920
## 31  0.00273625     30   0.25656 0.38076 0.015858
## 32  0.00269698     31   0.25383 0.37995 0.015844
## 33  0.00265792     32   0.25113 0.37956 0.015832
## 34  0.00261475     33   0.24847 0.37980 0.015840
## 35  0.00259402     34   0.24586 0.37859 0.015715
## 36  0.00242374     35   0.24326 0.37959 0.015779
## 37  0.00234022     36   0.24084 0.37503 0.015513
## 38  0.00226835     37   0.23850 0.37287 0.015447
## 39  0.00217154     39   0.23396 0.37288 0.015448
## 40  0.00203158     40   0.23179 0.37243 0.015346
## 41  0.00200903     41   0.22976 0.37431 0.015472
## 42  0.00197756     43   0.22574 0.37505 0.015577
## 43  0.00192518     44   0.22376 0.37442 0.015573
## 44  0.00180554     45   0.22184 0.37678 0.015624
## 45  0.00174488     47   0.21823 0.37611 0.015527
## 46  0.00171649     48   0.21648 0.37830 0.015566
## 47  0.00171513     49   0.21476 0.38068 0.015755
## 48  0.00163250     50   0.21305 0.38104 0.015963
## 49  0.00160365     51   0.21142 0.38176 0.016181
## 50  0.00154749     52   0.20981 0.38265 0.016209
## 51  0.00145486     53   0.20827 0.38162 0.016262
## 52  0.00145420     54   0.20681 0.38016 0.016381
## 53  0.00142485     55   0.20536 0.37991 0.016379
## 54  0.00140643     56   0.20393 0.38211 0.016568
## 55  0.00139441     58   0.20112 0.38218 0.016571
## 56  0.00136615     59   0.19972 0.38015 0.016515
```

```
## 57  0.00134432     62   0.19563 0.37871 0.016357
## 58  0.00132904     63   0.19428 0.37837 0.016352
## 59  0.00132240     64   0.19295 0.37834 0.016352
## 60  0.00123928     65   0.19163 0.37777 0.016352
## 61  0.00123217     66   0.19039 0.37504 0.016298
## 62  0.00116417     67   0.18916 0.37569 0.016361
## 63  0.00115728     68   0.18799 0.37632 0.016366
## 64  0.00111347     70   0.18568 0.37569 0.016354
## 65  0.00107149     71   0.18457 0.37511 0.016345
## 66  0.00106199     72   0.18350 0.37538 0.016356
## 67  0.00099296     73   0.18243 0.37490 0.016329
## 68  0.00097298     74   0.18144 0.37556 0.016330
## 69  0.00096571     75   0.18047 0.37542 0.016292
## 70  0.00094299     76   0.17950 0.37649 0.016308
## 71  0.00092604     77   0.17856 0.37593 0.016295
## 72  0.00086791     78   0.17763 0.37502 0.016268
## 73  0.00086082     80   0.17590 0.37548 0.016281
## 74  0.00083882     81   0.17504 0.37629 0.016343
## 75  0.00082185     82   0.17420 0.37630 0.016378
## 76  0.00081999     83   0.17338 0.37582 0.016374
## 77  0.00081510     84   0.17256 0.37592 0.016374
## 78  0.00080718     85   0.17174 0.37617 0.016378
## 79  0.00080253     86   0.17093 0.37614 0.016377
## 80  0.00080214     87   0.17013 0.37614 0.016377
## 81  0.00079999     88   0.16933 0.37614 0.016377
## 82  0.00077675     89   0.16853 0.37659 0.016384
## 83  0.00076692     90   0.16775 0.37645 0.016385
## 84  0.00075579     91   0.16698 0.37653 0.016389
## 85  0.00074858     92   0.16623 0.37532 0.016196
## 86  0.00070380     94   0.16473 0.37485 0.016141
## 87  0.00069992     95   0.16403 0.37471 0.016189
## 88  0.00069223     96   0.16333 0.37467 0.016190
## 89  0.00068902     97   0.16264 0.37435 0.016191
## 90  0.00068832     98   0.16195 0.37435 0.016191
## 91  0.00066915     99   0.16126 0.37481 0.016188
## 92  0.00066580    100   0.16059 0.37511 0.016189
## 93  0.00063716    103   0.15859 0.37537 0.016187
## 94  0.00063355    104   0.15795 0.37582 0.016187
## 95  0.00061813    105   0.15732 0.37580 0.016186
## 96  0.00061596    107   0.15609 0.37610 0.016189
## 97  0.00061410    108   0.15547 0.37672 0.016200
## 98  0.00060509    109   0.15486 0.37629 0.016183
## 99  0.00059366    110   0.15425 0.37641 0.016191
## 100 0.00057957    111   0.15366 0.37546 0.016174
## 101 0.00057526    112   0.15308 0.37577 0.016175
## 102 0.00055524    114   0.15193 0.37570 0.016163
## 103 0.00055083    115   0.15137 0.37681 0.016163
## 104 0.00054831    116   0.15082 0.37681 0.016176
## 105 0.00054719    117   0.15027 0.37787 0.016204
## 106 0.00049059    118   0.14972 0.37830 0.016209
## 107 0.00048324    119   0.14923 0.37958 0.016219
## 108 0.00048039    120   0.14875 0.37966 0.016232
## 109 0.00046824    121   0.14827 0.37941 0.016222
## 110 0.00046728    122   0.14780 0.37890 0.016205
```

```
## 111 0.00046207    123   0.14733 0.37944 0.016215
## 112 0.00044998    124   0.14687 0.38030 0.016223
## 113 0.00043664    125   0.14642 0.38083 0.016200
## 114 0.00043135    126   0.14599 0.38023 0.016189
## 115 0.00042858    127   0.14555 0.38059 0.016189
## 116 0.00041923    128   0.14513 0.38118 0.016187
## 117 0.00041219    129   0.14471 0.38148 0.016204
## 118 0.00040882    130   0.14429 0.38160 0.016206
## 119 0.00040711    131   0.14389 0.38111 0.016204
## 120 0.00040614    132   0.14348 0.38096 0.016204
## 121 0.00039895    133   0.14307 0.38127 0.016218
## 122 0.00038747    134   0.14267 0.38171 0.016204
## 123 0.00038480    135   0.14229 0.38204 0.016211
## 124 0.00038424    136   0.14190 0.38195 0.016212
## 125 0.00037809    137   0.14152 0.38171 0.016210
## 126 0.00037584    138   0.14114 0.38200 0.016212
## 127 0.00036870    140   0.14039 0.38254 0.016246
## 128 0.00034819    141   0.14002 0.38310 0.016276
## 129 0.00034302    142   0.13967 0.38367 0.016279
## 130 0.00033810    143   0.13933 0.38394 0.016283
## 131 0.00033694    144   0.13899 0.38396 0.016283
## 132 0.00033661    145   0.13865 0.38393 0.016284
## 133 0.00033647    146   0.13832 0.38393 0.016284
## 134 0.00032613    147   0.13798 0.38402 0.016283
## 135 0.00031906    148   0.13765 0.38458 0.016288
## 136 0.00031893    149   0.13733 0.38459 0.016274
## 137 0.00031861    150   0.13702 0.38459 0.016274
## 138 0.00031509    151   0.13670 0.38476 0.016277
## 139 0.00031365    152   0.13638 0.38492 0.016277
## 140 0.00031258    153   0.13607 0.38477 0.016277
## 141 0.00030788    154   0.13576 0.38479 0.016277
## 142 0.00029802    155   0.13545 0.38523 0.016274
## 143 0.00029594    156   0.13515 0.38566 0.016274
## 144 0.00027781    158   0.13456 0.38604 0.016263
## 145 0.00027672    159   0.13428 0.38569 0.016174
## 146 0.00027287    160   0.13400 0.38549 0.016161
## 147 0.00027246    161   0.13373 0.38549 0.016161
## 148 0.00027010    162   0.13346 0.38547 0.016161
## 149 0.00025144    163   0.13319 0.38559 0.016151
## 150 0.00024464    164   0.13294 0.38657 0.016150
## 151 0.00024056    165   0.13269 0.38642 0.016150
## 152 0.00023660    166   0.13245 0.38727 0.016161
## 153 0.00023198    167   0.13221 0.38759 0.016156
## 154 0.00023124    168   0.13198 0.38742 0.016156
## 155 0.00022249    169   0.13175 0.38728 0.016155
## 156 0.00021870    170   0.13153 0.38719 0.016158
## 157 0.00020670    173   0.13087 0.38740 0.016158
## 158 0.00020481    174   0.13067 0.38749 0.016161
## 159 0.00020284    175   0.13046 0.38745 0.016162
## 160 0.00020023    176   0.13026 0.38756 0.016161
## 161 0.00020000    177   0.13006 0.38767 0.016161
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
```

```
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC09
##  [6] HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.x
## [11] HC03_VC13.y HC03_VC130  HC03_VC132  HC03_VC133  HC03_VC134
## [16] HC03_VC15   HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC75
## [21] HC03_VC76   HC03_VC77   HC03_VC78   HC03_VC79   HC03_VC80
## [26] HC03_VC81   HC03_VC82   HC03_VC83   HC03_VC84   HC03_VC85
## [31] HC03_VC86   HC03_VC87   HC03_VC88   HC03_VC89   HC03_VC90
## [36] HC03_VC91   Latitude    Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##             CP nsplit rel error  xerror     xstd
## 1   0.26894188      0   1.00000 1.00133 0.031335
## 2   0.15300146      1   0.73106 0.73646 0.023571
## 3   0.04905506      2   0.57806 0.58767 0.021008
## 4   0.03483358      3   0.52900 0.53770 0.019945
## 5   0.02575390      4   0.49417 0.51441 0.019475
## 6   0.02255952      5   0.46841 0.50731 0.019115
## 7   0.02195363      6   0.44585 0.49091 0.017793
## 8   0.01812404      7   0.42390 0.49023 0.017805
## 9   0.01460201      8   0.40578 0.47412 0.017316
## 10  0.01373019      9   0.39117 0.45665 0.017246
## 11  0.01318438     10   0.37744 0.45007 0.017105
## 12  0.01113299     11   0.36426 0.44542 0.017206
## 13  0.01044189     12   0.35313 0.44211 0.017433
## 14  0.00876037     13   0.34269 0.43582 0.017236
## 15  0.00771485     14   0.33393 0.42560 0.017027
## 16  0.00737522     15   0.32621 0.42484 0.017021
## 17  0.00651898     16   0.31884 0.42004 0.016913
## 18  0.00609098     17   0.31232 0.41644 0.016923
## 19  0.00582690     18   0.30623 0.40979 0.016929
## 20  0.00523612     19   0.30040 0.41367 0.017277
## 21  0.00482466     20   0.29516 0.40742 0.017128
## 22  0.00479629     21   0.29034 0.40728 0.017126
## 23  0.00455377     22   0.28554 0.40819 0.017127
## 24  0.00406023     23   0.28099 0.40427 0.017173
## 25  0.00390075     24   0.27693 0.40443 0.017546
## 26  0.00386309     25   0.27303 0.40309 0.017491
## 27  0.00373690     26   0.26916 0.40344 0.017505
## 28  0.00323021     27   0.26543 0.39891 0.017132
## 29  0.00287222     28   0.26220 0.39503 0.017073
## 30  0.00276227     29   0.25932 0.39864 0.017162
## 31  0.00273625     30   0.25656 0.39972 0.017177
## 32  0.00269698     31   0.25383 0.39877 0.017150
## 33  0.00265792     32   0.25113 0.39715 0.017129
## 34  0.00261475     33   0.24847 0.39673 0.017129
## 35  0.00259402     34   0.24586 0.39583 0.017151
## 36  0.00242374     35   0.24326 0.39343 0.016660
## 37  0.00234022     36   0.24084 0.39186 0.016550
```

```
## 38  0.00226835     37   0.23850 0.38945 0.016427
## 39  0.00217154     39   0.23396 0.38989 0.016392
## 40  0.00203158     40   0.23179 0.38776 0.016214
## 41  0.00200903     41   0.22976 0.38472 0.015995
## 42  0.00197756     43   0.22574 0.38393 0.015936
## 43  0.00192518     44   0.22376 0.38246 0.015927
## 44  0.00180554     45   0.22184 0.38250 0.015869
## 45  0.00174488     47   0.21823 0.37904 0.015735
## 46  0.00171649     48   0.21648 0.38131 0.015852
## 47  0.00171513     49   0.21476 0.38141 0.015851
## 48  0.00163250     50   0.21305 0.38130 0.015938
## 49  0.00160365     51   0.21142 0.38112 0.015947
## 50  0.00154749     52   0.20981 0.38255 0.015971
## 51  0.00145486     53   0.20827 0.38074 0.015888
## 52  0.00145420     54   0.20681 0.37962 0.015899
## 53  0.00142485     55   0.20536 0.37958 0.015898
## 54  0.00140643     56   0.20393 0.37940 0.015900
## 55  0.00139441     58   0.20112 0.37929 0.015938
## 56  0.00136615     59   0.19972 0.37840 0.015964
## 57  0.00134432     62   0.19563 0.37781 0.015996
## 58  0.00132904     63   0.19428 0.37819 0.015997
## 59  0.00132240     64   0.19295 0.37818 0.015997
## 60  0.00123928     65   0.19163 0.37661 0.015913
## 61  0.00123217     66   0.19039 0.37581 0.015899
## 62  0.00116417     67   0.18916 0.37435 0.015840
## 63  0.00115728     68   0.18799 0.37433 0.015873
## 64  0.00111347     70   0.18568 0.37435 0.015764
## 65  0.00107149     71   0.18457 0.37177 0.015710
## 66  0.00106199     72   0.18350 0.37289 0.015735
## 67  0.00099296     73   0.18243 0.37279 0.015575
## 68  0.00097298     74   0.18144 0.36966 0.015452
## 69  0.00096571     75   0.18047 0.36894 0.015427
## 70  0.00094299     76   0.17950 0.36846 0.015410
## 71  0.00092604     77   0.17856 0.36805 0.015408
## 72  0.00086791     78   0.17763 0.36870 0.015422
## 73  0.00086082     80   0.17590 0.36805 0.015413
## 74  0.00083882     81   0.17504 0.36722 0.015396
## 75  0.00082185     82   0.17420 0.36745 0.015430
## 76  0.00081999     83   0.17338 0.36647 0.015439
## 77  0.00081510     84   0.17256 0.36647 0.015439
## 78  0.00080718     85   0.17174 0.36711 0.015476
## 79  0.00080253     86   0.17093 0.36716 0.015476
## 80  0.00080214     87   0.17013 0.36716 0.015476
## 81  0.00079999     88   0.16933 0.36716 0.015476
## 82  0.00077675     89   0.16853 0.36648 0.015443
## 83  0.00076692     90   0.16775 0.36734 0.015447
## 84  0.00075579     91   0.16698 0.36918 0.015506
## 85  0.00074858     92   0.16623 0.37072 0.015521
## 86  0.00070380     94   0.16473 0.36958 0.015473
## 87  0.00069992     95   0.16403 0.37019 0.015443
## 88  0.00069223     96   0.16333 0.37017 0.015458
## 89  0.00068902     97   0.16264 0.36959 0.015421
## 90  0.00068832     98   0.16195 0.36936 0.015422
## 91  0.00066915     99   0.16126 0.36885 0.015420
```

```
## 92  0.00066580    100   0.16059 0.36835 0.015247
## 93  0.00063716    103   0.15859 0.36832 0.015248
## 94  0.00063355    104   0.15795 0.36842 0.015290
## 95  0.00061813    105   0.15732 0.36812 0.015287
## 96  0.00061596    107   0.15609 0.36869 0.015298
## 97  0.00061410    108   0.15547 0.36866 0.015298
## 98  0.00060509    109   0.15486 0.36869 0.015298
## 99  0.00059366    110   0.15425 0.36785 0.015273
## 100 0.00057957    111   0.15366 0.36838 0.015316
## 101 0.00057526    112   0.15308 0.36859 0.015463
## 102 0.00055524    114   0.15193 0.36876 0.015461
## 103 0.00055083    115   0.15137 0.36938 0.015466
## 104 0.00054831    116   0.15082 0.36927 0.015467
## 105 0.00054719    117   0.15027 0.36927 0.015467
## 106 0.00049059    118   0.14972 0.37019 0.015488
## 107 0.00048324    119   0.14923 0.37351 0.015566
## 108 0.00048039    120   0.14875 0.37411 0.015569
## 109 0.00046824    121   0.14827 0.37527 0.015584
## 110 0.00046728    122   0.14780 0.37483 0.015580
## 111 0.00046207    123   0.14733 0.37541 0.015610
## 112 0.00044998    124   0.14687 0.37655 0.015620
## 113 0.00043664    125   0.14642 0.37687 0.015617
## 114 0.00043135    126   0.14599 0.37703 0.015641
## 115 0.00042858    127   0.14555 0.37702 0.015641
## 116 0.00041923    128   0.14513 0.37695 0.015634
## 117 0.00041219    129   0.14471 0.37773 0.015637
## 118 0.00040882    130   0.14429 0.37776 0.015637
## 119 0.00040711    131   0.14389 0.37792 0.015638
## 120 0.00040614    132   0.14348 0.37792 0.015638
## 121 0.00039895    133   0.14307 0.37809 0.015652
## 122 0.00038747    134   0.14267 0.37830 0.015556
## 123 0.00038480    135   0.14229 0.37873 0.015562
## 124 0.00038424    136   0.14190 0.37864 0.015561
## 125 0.00037809    137   0.14152 0.37875 0.015563
## 126 0.00037584    138   0.14114 0.37816 0.015563
## 127 0.00036870    140   0.14039 0.37860 0.015567
## 128 0.00034819    141   0.14002 0.37805 0.015563
## 129 0.00034302    142   0.13967 0.37851 0.015659
## 130 0.00033810    143   0.13933 0.37822 0.015648
## 131 0.00033694    144   0.13899 0.37818 0.015645
## 132 0.00033661    145   0.13865 0.37816 0.015645
## 133 0.00033647    146   0.13832 0.37816 0.015645
## 134 0.00032613    147   0.13798 0.37834 0.015639
## 135 0.00031906    148   0.13765 0.37844 0.015638
## 136 0.00031893    149   0.13733 0.37829 0.015638
## 137 0.00031861    150   0.13702 0.37829 0.015638
## 138 0.00031509    151   0.13670 0.37829 0.015638
## 139 0.00031365    152   0.13638 0.37905 0.015643
## 140 0.00031258    153   0.13607 0.37898 0.015643
## 141 0.00030788    154   0.13576 0.37911 0.015643
## 142 0.00030000    155   0.13545 0.37880 0.015631
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
```

```
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC09
##  [6] HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.x
## [11] HC03_VC13.y HC03_VC130  HC03_VC132  HC03_VC133  HC03_VC134
## [16] HC03_VC15   HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC75
## [21] HC03_VC76   HC03_VC79   HC03_VC80   HC03_VC81   HC03_VC82
## [26] HC03_VC83   HC03_VC84   HC03_VC85   HC03_VC86   HC03_VC87
## [31] HC03_VC88   HC03_VC89   HC03_VC90   HC03_VC91   Latitude
## [36] Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##             CP nsplit rel error  xerror     xstd
## 1   0.26894188      0   1.00000 1.00081 0.031346
## 2   0.15300146      1   0.73106 0.73980 0.023892
## 3   0.04905506      2   0.57806 0.59109 0.021456
## 4   0.03483358      3   0.52900 0.54339 0.020548
## 5   0.02575390      4   0.49417 0.52001 0.019567
## 6   0.02255952      5   0.46841 0.50708 0.019246
## 7   0.02195363      6   0.44585 0.49595 0.018858
## 8   0.01812404      7   0.42390 0.48499 0.018363
## 9   0.01460201      8   0.40578 0.47541 0.018145
## 10  0.01373019      9   0.39117 0.46087 0.018012
## 11  0.01318438     10   0.37744 0.45836 0.018072
## 12  0.01113299     11   0.36426 0.45966 0.018439
## 13  0.01044189     12   0.35313 0.45659 0.018382
## 14  0.00876037     13   0.34269 0.44672 0.017959
## 15  0.00771485     14   0.33393 0.44149 0.017727
## 16  0.00737522     15   0.32621 0.43854 0.017694
## 17  0.00651898     16   0.31884 0.43065 0.017486
## 18  0.00609098     17   0.31232 0.42930 0.017374
## 19  0.00582690     18   0.30623 0.42894 0.017453
## 20  0.00523612     19   0.30040 0.42519 0.017205
## 21  0.00482466     20   0.29516 0.42208 0.017448
## 22  0.00479629     21   0.29034 0.42208 0.017423
## 23  0.00455377     22   0.28554 0.42148 0.017218
## 24  0.00406023     23   0.28099 0.41791 0.017051
## 25  0.00390075     24   0.27693 0.41531 0.017085
## 26  0.00386309     25   0.27303 0.41387 0.017069
## 27  0.00373690     26   0.26916 0.41307 0.017026
## 28  0.00323021     27   0.26543 0.40488 0.016657
## 29  0.00287222     28   0.26220 0.40023 0.016537
## 30  0.00276227     29   0.25932 0.40002 0.016745
## 31  0.00273625     30   0.25656 0.40068 0.016779
## 32  0.00269698     31   0.25383 0.39987 0.016776
## 33  0.00265792     32   0.25113 0.39861 0.016710
## 34  0.00261475     33   0.24847 0.39750 0.016629
## 35  0.00259402     34   0.24586 0.39757 0.016639
## 36  0.00242374     35   0.24326 0.39911 0.016660
## 37  0.00234022     36   0.24084 0.39906 0.016641
```

```
## 38  0.00226835     37   0.23850 0.39822 0.016668
## 39  0.00217154     39   0.23396 0.39537 0.016438
## 40  0.00203158     40   0.23179 0.39599 0.016448
## 41  0.00200903     41   0.22976 0.39536 0.016518
## 42  0.00197756     43   0.22574 0.39741 0.016542
## 43  0.00192518     44   0.22376 0.39693 0.016486
## 44  0.00180554     45   0.22184 0.39619 0.016509
## 45  0.00174488     47   0.21823 0.39445 0.016470
## 46  0.00171649     48   0.21648 0.39210 0.016501
## 47  0.00171513     49   0.21476 0.39146 0.016481
## 48  0.00163250     50   0.21305 0.39157 0.016483
## 49  0.00160365     51   0.21142 0.39158 0.016488
## 50  0.00154749     52   0.20981 0.39137 0.016495
## 51  0.00145486     53   0.20827 0.38930 0.016424
## 52  0.00145420     54   0.20681 0.38697 0.016400
## 53  0.00142485     55   0.20536 0.38676 0.016366
## 54  0.00140643     56   0.20393 0.38836 0.016482
## 55  0.00139441     58   0.20112 0.38840 0.016491
## 56  0.00136615     59   0.19972 0.38877 0.016493
## 57  0.00134432     62   0.19563 0.38698 0.016566
## 58  0.00132904     63   0.19428 0.38707 0.016564
## 59  0.00132240     64   0.19295 0.38571 0.016541
## 60  0.00123928     65   0.19163 0.38519 0.016537
## 61  0.00123217     66   0.19039 0.38450 0.016521
## 62  0.00116417     67   0.18916 0.38225 0.016048
## 63  0.00115728     68   0.18799 0.38193 0.016030
## 64  0.00111347     70   0.18568 0.38389 0.016083
## 65  0.00107149     71   0.18457 0.38336 0.016082
## 66  0.00106199     72   0.18350 0.38242 0.016025
## 67  0.00099296     73   0.18243 0.38132 0.016048
## 68  0.00097298     74   0.18144 0.38050 0.015933
## 69  0.00096571     75   0.18047 0.38014 0.015936
## 70  0.00094299     76   0.17950 0.37968 0.015914
## 71  0.00092604     77   0.17856 0.38074 0.015982
## 72  0.00086791     78   0.17763 0.38089 0.015933
## 73  0.00086082     80   0.17590 0.38061 0.015935
## 74  0.00083882     81   0.17504 0.37969 0.015925
## 75  0.00082185     82   0.17420 0.37975 0.015945
## 76  0.00081999     83   0.17338 0.37883 0.015935
## 77  0.00081510     84   0.17256 0.37887 0.015925
## 78  0.00080718     85   0.17174 0.37873 0.015922
## 79  0.00080253     86   0.17093 0.37865 0.015923
## 80  0.00080214     87   0.17013 0.37907 0.015930
## 81  0.00079999     88   0.16933 0.37907 0.015930
## 82  0.00077675     89   0.16853 0.37937 0.015934
## 83  0.00076692     90   0.16775 0.37773 0.015890
## 84  0.00075579     91   0.16698 0.37758 0.015892
## 85  0.00074858     92   0.16623 0.37763 0.015844
## 86  0.00070380     94   0.16473 0.37688 0.015894
## 87  0.00069992     95   0.16403 0.37823 0.015930
## 88  0.00069223     96   0.16333 0.37863 0.015950
## 89  0.00068902     97   0.16264 0.37857 0.015947
## 90  0.00068832     98   0.16195 0.37870 0.015947
## 91  0.00066915     99   0.16126 0.37863 0.016005
```

```
## 92  0.00066580    100   0.16059 0.37868 0.015944
## 93  0.00063716    103   0.15859 0.37921 0.015964
## 94  0.00063355    104   0.15795 0.37897 0.015961
## 95  0.00061813    105   0.15732 0.37864 0.015951
## 96  0.00061596    107   0.15609 0.37803 0.015945
## 97  0.00061410    108   0.15547 0.37762 0.015946
## 98  0.00060509    109   0.15486 0.37812 0.015995
## 99  0.00059366    110   0.15425 0.37875 0.016016
## 100 0.00057957    111   0.15366 0.38001 0.016037
## 101 0.00057526    112   0.15308 0.37983 0.016036
## 102 0.00055524    114   0.15193 0.37992 0.016038
## 103 0.00055083    115   0.15137 0.37970 0.016043
## 104 0.00054831    116   0.15082 0.37977 0.016045
## 105 0.00054719    117   0.15027 0.37991 0.016044
## 106 0.00049059    118   0.14972 0.38077 0.016032
## 107 0.00048324    119   0.14923 0.37922 0.016024
## 108 0.00048039    120   0.14875 0.38015 0.016038
## 109 0.00046824    121   0.14827 0.38045 0.016036
## 110 0.00046728    122   0.14780 0.38066 0.016036
## 111 0.00046207    123   0.14733 0.38042 0.016031
## 112 0.00044998    124   0.14687 0.38060 0.016038
## 113 0.00043664    125   0.14642 0.38102 0.016040
## 114 0.00043135    126   0.14599 0.38098 0.016088
## 115 0.00042858    127   0.14555 0.38048 0.016056
## 116 0.00041923    128   0.14513 0.38032 0.016057
## 117 0.00041219    129   0.14471 0.38001 0.016060
## 118 0.00040882    130   0.14429 0.38042 0.016060
## 119 0.00040711    131   0.14389 0.38031 0.016061
## 120 0.00040614    132   0.14348 0.38031 0.016061
## 121 0.00040000    133   0.14307 0.38011 0.016061
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC09
##  [6] HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.y
## [11] HC03_VC130  HC03_VC132  HC03_VC133  HC03_VC134  HC03_VC15
## [16] HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC75   HC03_VC76
## [21] HC03_VC79   HC03_VC82   HC03_VC83   HC03_VC84   HC03_VC85
## [26] HC03_VC86   HC03_VC87   HC03_VC89   HC03_VC90   HC03_VC91
## [31] Latitude    Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##             CP nsplit rel error  xerror     xstd
## 1   0.26894188      0   1.00000 1.00022 0.031311
## 2   0.15300146      1   0.73106 0.74471 0.024177
## 3   0.04905506      2   0.57806 0.59817 0.022035
## 4   0.03483358      3   0.52900 0.55347 0.021163
## 5   0.02575390      4   0.49417 0.53739 0.020497
```

```
## 6   0.02255952      5   0.46841 0.52245 0.019968
## 7   0.02195363      6   0.44585 0.51215 0.019193
## 8   0.01812404      7   0.42390 0.50232 0.019626
## 9   0.01460201      8   0.40578 0.48120 0.018655
## 10  0.01373019      9   0.39117 0.46738 0.018491
## 11  0.01318438     10   0.37744 0.46477 0.018467
## 12  0.01113299     11   0.36426 0.45810 0.018225
## 13  0.01044189     12   0.35313 0.45648 0.018662
## 14  0.00876037     13   0.34269 0.45102 0.018431
## 15  0.00771485     14   0.33393 0.44553 0.018223
## 16  0.00737522     15   0.32621 0.44112 0.018053
## 17  0.00651898     16   0.31884 0.44046 0.018052
## 18  0.00609098     17   0.31232 0.43693 0.018007
## 19  0.00582690     18   0.30623 0.43398 0.017970
## 20  0.00523612     19   0.30040 0.42767 0.017756
## 21  0.00482466     20   0.29516 0.42320 0.017691
## 22  0.00479629     21   0.29034 0.42542 0.018165
## 23  0.00455377     22   0.28554 0.42530 0.018278
## 24  0.00406023     23   0.28099 0.42346 0.018193
## 25  0.00390075     24   0.27693 0.41815 0.018309
## 26  0.00386309     25   0.27303 0.41544 0.018133
## 27  0.00373690     26   0.26916 0.41564 0.018160
## 28  0.00323021     27   0.26543 0.41017 0.017765
## 29  0.00287222     28   0.26220 0.40285 0.017632
## 30  0.00276227     29   0.25932 0.40291 0.017623
## 31  0.00273625     30   0.25656 0.40119 0.017580
## 32  0.00269698     31   0.25383 0.40509 0.017664
## 33  0.00265792     32   0.25113 0.40607 0.017714
## 34  0.00261475     33   0.24847 0.40566 0.017703
## 35  0.00259402     34   0.24586 0.40633 0.017795
## 36  0.00242374     35   0.24326 0.40284 0.017698
## 37  0.00234022     36   0.24084 0.39962 0.017499
## 38  0.00226835     37   0.23850 0.39969 0.017484
## 39  0.00217154     39   0.23396 0.39973 0.017535
## 40  0.00203158     40   0.23179 0.39977 0.017730
## 41  0.00200903     41   0.22976 0.40165 0.017936
## 42  0.00197756     43   0.22574 0.39967 0.017864
## 43  0.00192518     44   0.22376 0.39591 0.017658
## 44  0.00180554     45   0.22184 0.39574 0.017656
## 45  0.00174488     47   0.21823 0.39433 0.017645
## 46  0.00171649     48   0.21648 0.39450 0.017654
## 47  0.00171513     49   0.21476 0.39511 0.017706
## 48  0.00163250     50   0.21305 0.39686 0.017864
## 49  0.00160365     51   0.21142 0.39667 0.017845
## 50  0.00154749     52   0.20981 0.39575 0.017842
## 51  0.00145486     53   0.20827 0.39602 0.017705
## 52  0.00145420     54   0.20681 0.39728 0.017763
## 53  0.00142485     55   0.20536 0.39725 0.017773
## 54  0.00140643     56   0.20393 0.39803 0.017861
## 55  0.00139441     58   0.20112 0.39777 0.017861
## 56  0.00136615     59   0.19972 0.39826 0.017895
## 57  0.00134432     62   0.19563 0.39686 0.017877
## 58  0.00132904     63   0.19428 0.39724 0.017879
## 59  0.00132240     64   0.19295 0.39686 0.017880
```

```
## 60  0.00123928     65   0.19163 0.39762 0.017953
## 61  0.00123217     66   0.19039 0.39781 0.017957
## 62  0.00116417     67   0.18916 0.39929 0.018061
## 63  0.00115728     68   0.18799 0.40038 0.018107
## 64  0.00111347     70   0.18568 0.39982 0.018115
## 65  0.00107149     71   0.18457 0.40152 0.018165
## 66  0.00106199     72   0.18350 0.40123 0.018132
## 67  0.00099296     73   0.18243 0.40071 0.018125
## 68  0.00097298     74   0.18144 0.39981 0.018078
## 69  0.00096571     75   0.18047 0.39959 0.018080
## 70  0.00094299     76   0.17950 0.39942 0.018022
## 71  0.00092604     77   0.17856 0.39930 0.018022
## 72  0.00086791     78   0.17763 0.39867 0.018002
## 73  0.00086082     80   0.17590 0.39793 0.017982
## 74  0.00083882     81   0.17504 0.39704 0.017972
## 75  0.00082185     82   0.17420 0.39636 0.017954
## 76  0.00081999     83   0.17338 0.39633 0.017944
## 77  0.00081510     84   0.17256 0.39621 0.017944
## 78  0.00080718     85   0.17174 0.39645 0.017926
## 79  0.00080253     86   0.17093 0.39622 0.017922
## 80  0.00080214     87   0.17013 0.39596 0.017922
## 81  0.00079999     88   0.16933 0.39603 0.017922
## 82  0.00077675     89   0.16853 0.39615 0.017919
## 83  0.00076692     90   0.16775 0.39681 0.017932
## 84  0.00075579     91   0.16698 0.39492 0.017810
## 85  0.00074858     92   0.16623 0.39438 0.017804
## 86  0.00070380     94   0.16473 0.39464 0.017789
## 87  0.00069992     95   0.16403 0.39574 0.017848
## 88  0.00069223     96   0.16333 0.39565 0.017847
## 89  0.00068902     97   0.16264 0.39645 0.017849
## 90  0.00068832     98   0.16195 0.39645 0.017849
## 91  0.00066915     99   0.16126 0.39752 0.017841
## 92  0.00066580    100   0.16059 0.39770 0.017851
## 93  0.00063716    103   0.15859 0.39786 0.017869
## 94  0.00063355    104   0.15795 0.40026 0.017895
## 95  0.00061813    105   0.15732 0.40030 0.017878
## 96  0.00061596    107   0.15609 0.40010 0.017837
## 97  0.00061410    108   0.15547 0.40010 0.017834
## 98  0.00060509    109   0.15486 0.40023 0.017838
## 99  0.00059366    110   0.15425 0.40011 0.017909
## 100 0.00057957    111   0.15366 0.40046 0.017909
## 101 0.00057526    112   0.15308 0.40072 0.017917
## 102 0.00055524    114   0.15193 0.40051 0.017914
## 103 0.00055083    115   0.15137 0.40082 0.017923
## 104 0.00054831    116   0.15082 0.40078 0.017923
## 105 0.00054719    117   0.15027 0.40080 0.017923
## 106 0.00050000    118   0.14972 0.40046 0.017892
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC09
```

```
##  [6] HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.y
## [11] HC03_VC130  HC03_VC132  HC03_VC133  HC03_VC134  HC03_VC15
## [16] HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC75   HC03_VC76
## [21] HC03_VC79   HC03_VC82   HC03_VC83   HC03_VC84   HC03_VC85
## [26] HC03_VC86   HC03_VC87   HC03_VC89   HC03_VC90   HC03_VC91
## [31] Latitude    Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##             CP nsplit rel error  xerror     xstd
## 1  0.26894188      0   1.00000 1.00047 0.031320
## 2  0.15300146      1   0.73106 0.74013 0.023771
## 3  0.04905506      2   0.57806 0.59665 0.021569
## 4  0.03483358      3   0.52900 0.55348 0.020697
## 5  0.02575390      4   0.49417 0.52405 0.020313
## 6  0.02255952      5   0.46841 0.51486 0.019362
## 7  0.02195363      6   0.44585 0.50738 0.019098
## 8  0.01812404      7   0.42390 0.49945 0.018709
## 9  0.01460201      8   0.40578 0.48887 0.018613
## 10 0.01373019      9   0.39117 0.46961 0.018232
## 11 0.01318438     10   0.37744 0.46865 0.018194
## 12 0.01113299     11   0.36426 0.45803 0.017965
## 13 0.01044189     12   0.35313 0.46251 0.018727
## 14 0.00876037     13   0.34269 0.44983 0.018298
## 15 0.00771485     14   0.33393 0.44494 0.017849
## 16 0.00737522     15   0.32621 0.43564 0.017258
## 17 0.00651898     16   0.31884 0.42600 0.016849
## 18 0.00609098     17   0.31232 0.42382 0.017003
## 19 0.00582690     18   0.30623 0.42050 0.016961
## 20 0.00523612     19   0.30040 0.41970 0.016912
## 21 0.00482466     20   0.29516 0.41688 0.016875
## 22 0.00479629     21   0.29034 0.41248 0.016885
## 23 0.00455377     22   0.28554 0.41247 0.016904
## 24 0.00406023     23   0.28099 0.41249 0.017295
## 25 0.00390075     24   0.27693 0.40967 0.017266
## 26 0.00386309     25   0.27303 0.40992 0.017289
## 27 0.00373690     26   0.26916 0.40910 0.017200
## 28 0.00323021     27   0.26543 0.40527 0.017192
## 29 0.00287222     28   0.26220 0.39842 0.016849
## 30 0.00276227     29   0.25932 0.39799 0.016730
## 31 0.00273625     30   0.25656 0.39782 0.016730
## 32 0.00269698     31   0.25383 0.39921 0.016756
## 33 0.00265792     32   0.25113 0.39844 0.016742
## 34 0.00261475     33   0.24847 0.39895 0.016761
## 35 0.00259402     34   0.24586 0.39830 0.016723
## 36 0.00242374     35   0.24326 0.39868 0.016718
## 37 0.00234022     36   0.24084 0.39910 0.016722
## 38 0.00226835     37   0.23850 0.39886 0.016721
## 39 0.00217154     39   0.23396 0.39758 0.016833
## 40 0.00203158     40   0.23179 0.39824 0.017126
## 41 0.00200903     41   0.22976 0.39626 0.016920
## 42 0.00197756     43   0.22574 0.39644 0.016996
```

```
## 43 0.00192518     44   0.22376 0.39632 0.017015
## 44 0.00180554     45   0.22184 0.39756 0.017079
## 45 0.00174488     47   0.21823 0.39760 0.017065
## 46 0.00171649     48   0.21648 0.39825 0.017088
## 47 0.00171513     49   0.21476 0.39788 0.017087
## 48 0.00163250     50   0.21305 0.39734 0.016994
## 49 0.00160365     51   0.21142 0.39752 0.017133
## 50 0.00154749     52   0.20981 0.39611 0.017128
## 51 0.00145486     53   0.20827 0.39399 0.017001
## 52 0.00145420     54   0.20681 0.39350 0.017032
## 53 0.00142485     55   0.20536 0.39297 0.017054
## 54 0.00140643     56   0.20393 0.39253 0.017043
## 55 0.00139441     58   0.20112 0.39270 0.017008
## 56 0.00136615     59   0.19972 0.39284 0.017006
## 57 0.00134432     62   0.19563 0.39354 0.017032
## 58 0.00132904     63   0.19428 0.39373 0.017037
## 59 0.00132240     64   0.19295 0.39398 0.017029
## 60 0.00123928     65   0.19163 0.39138 0.016974
## 61 0.00123217     66   0.19039 0.38979 0.017015
## 62 0.00116417     67   0.18916 0.39093 0.017014
## 63 0.00115728     68   0.18799 0.39083 0.017000
## 64 0.00111347     70   0.18568 0.39119 0.017003
## 65 0.00107149     71   0.18457 0.39112 0.017076
## 66 0.00106199     72   0.18350 0.39064 0.017099
## 67 0.00099296     73   0.18243 0.39098 0.017126
## 68 0.00097298     74   0.18144 0.39147 0.017090
## 69 0.00096571     75   0.18047 0.39161 0.017090
## 70 0.00094299     76   0.17950 0.39171 0.017078
## 71 0.00092604     77   0.17856 0.39195 0.017094
## 72 0.00086791     78   0.17763 0.39282 0.017070
## 73 0.00086082     80   0.17590 0.39267 0.017008
## 74 0.00083882     81   0.17504 0.39232 0.016992
## 75 0.00082185     82   0.17420 0.39184 0.016976
## 76 0.00081999     83   0.17338 0.39192 0.016980
## 77 0.00081510     84   0.17256 0.39209 0.016981
## 78 0.00080718     85   0.17174 0.39180 0.016981
## 79 0.00080253     86   0.17093 0.39116 0.016979
## 80 0.00080214     87   0.17013 0.39100 0.016977
## 81 0.00079999     88   0.16933 0.39124 0.016993
## 82 0.00077675     89   0.16853 0.39105 0.016991
## 83 0.00076692     90   0.16775 0.39027 0.016986
## 84 0.00075579     91   0.16698 0.39057 0.016981
## 85 0.00074858     92   0.16623 0.39074 0.017005
## 86 0.00070380     94   0.16473 0.39101 0.017032
## 87 0.00069992     95   0.16403 0.39300 0.017071
## 88 0.00069223     96   0.16333 0.39263 0.017068
## 89 0.00068902     97   0.16264 0.39304 0.017115
## 90 0.00068832     98   0.16195 0.39146 0.016902
## 91 0.00066915     99   0.16126 0.39170 0.016896
## 92 0.00066580    100   0.16059 0.39147 0.016900
## 93 0.00063716    103   0.15859 0.39135 0.016905
## 94 0.00063355    104   0.15795 0.39004 0.016892
## 95 0.00061813    105   0.15732 0.38989 0.016891
## 96 0.00061596    107   0.15609 0.39049 0.016899
```

```
## 97 0.00061410    108   0.15547 0.39066 0.016901
## 98 0.00060509    109   0.15486 0.39141 0.016936
## 99 0.00060000    110   0.15425 0.39181 0.016941
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC09
##  [6] HC03_VC10   HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.y
## [11] HC03_VC130  HC03_VC132  HC03_VC133  HC03_VC134  HC03_VC156
## [16] HC03_VC17   HC03_VC18   HC03_VC76   HC03_VC79   HC03_VC82
## [21] HC03_VC83   HC03_VC84   HC03_VC85   HC03_VC86   HC03_VC87
## [26] HC03_VC89   HC03_VC90   HC03_VC91   Latitude    Longitude
## [31] State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##             CP nsplit rel error  xerror     xstd
## 1  0.26894188      0   1.00000 1.00064 0.031325
## 2  0.15300146      1   0.73106 0.73655 0.023601
## 3  0.04905506      2   0.57806 0.59150 0.020925
## 4  0.03483358      3   0.52900 0.54281 0.019910
## 5  0.02575390      4   0.49417 0.53638 0.020117
## 6  0.02255952      5   0.46841 0.51865 0.019716
## 7  0.02195363      6   0.44585 0.50125 0.019247
## 8  0.01812404      7   0.42390 0.50099 0.019246
## 9  0.01460201      8   0.40578 0.47829 0.017816
## 10 0.01373019      9   0.39117 0.46712 0.017787
## 11 0.01318438     10   0.37744 0.46386 0.017784
## 12 0.01113299     11   0.36426 0.45116 0.017541
## 13 0.01044189     12   0.35313 0.43514 0.016980
## 14 0.00876037     13   0.34269 0.43011 0.016827
## 15 0.00771485     14   0.33393 0.41994 0.016148
## 16 0.00737522     15   0.32621 0.41875 0.016145
## 17 0.00651898     16   0.31884 0.41454 0.016195
## 18 0.00609098     17   0.31232 0.41463 0.016275
## 19 0.00582690     18   0.30623 0.41215 0.016283
## 20 0.00523612     19   0.30040 0.41206 0.016301
## 21 0.00482466     20   0.29516 0.40275 0.015851
## 22 0.00479629     21   0.29034 0.40041 0.015860
## 23 0.00455377     22   0.28554 0.39867 0.015808
## 24 0.00406023     23   0.28099 0.39651 0.015952
## 25 0.00390075     24   0.27693 0.39687 0.015969
## 26 0.00386309     25   0.27303 0.39347 0.015355
## 27 0.00373690     26   0.26916 0.38738 0.015234
## 28 0.00323021     27   0.26543 0.38517 0.015090
## 29 0.00287222     28   0.26220 0.37835 0.014959
## 30 0.00276227     29   0.25932 0.38077 0.015346
## 31 0.00273625     30   0.25656 0.38020 0.015350
## 32 0.00269698     31   0.25383 0.38200 0.015390
```

```
## 33 0.00265792     32   0.25113 0.38158 0.015381
## 34 0.00261475     33   0.24847 0.38158 0.015391
## 35 0.00259402     34   0.24586 0.37999 0.015343
## 36 0.00242374     35   0.24326 0.37913 0.015449
## 37 0.00234022     36   0.24084 0.37843 0.015447
## 38 0.00226835     37   0.23850 0.37805 0.015450
## 39 0.00217154     39   0.23396 0.37808 0.015462
## 40 0.00203158     40   0.23179 0.37788 0.015590
## 41 0.00200903     41   0.22976 0.37617 0.015570
## 42 0.00197756     43   0.22574 0.37638 0.015607
## 43 0.00192518     44   0.22376 0.37388 0.015477
## 44 0.00180554     45   0.22184 0.37546 0.015513
## 45 0.00174488     47   0.21823 0.37486 0.015494
## 46 0.00171649     48   0.21648 0.37355 0.015485
## 47 0.00171513     49   0.21476 0.37252 0.015465
## 48 0.00163250     50   0.21305 0.37293 0.015563
## 49 0.00160365     51   0.21142 0.36837 0.014834
## 50 0.00154749     52   0.20981 0.36801 0.014812
## 51 0.00145486     53   0.20827 0.36542 0.014696
## 52 0.00145420     54   0.20681 0.36515 0.014666
## 53 0.00142485     55   0.20536 0.36553 0.014679
## 54 0.00140643     56   0.20393 0.36393 0.014606
## 55 0.00139441     58   0.20112 0.36347 0.014605
## 56 0.00136615     59   0.19972 0.36277 0.014626
## 57 0.00134432     62   0.19563 0.36227 0.014626
## 58 0.00132904     63   0.19428 0.36286 0.014644
## 59 0.00132240     64   0.19295 0.36251 0.014573
## 60 0.00123928     65   0.19163 0.36086 0.014444
## 61 0.00123217     66   0.19039 0.35771 0.014364
## 62 0.00116417     67   0.18916 0.35646 0.014366
## 63 0.00115728     68   0.18799 0.35542 0.014336
## 64 0.00111347     70   0.18568 0.35600 0.014390
## 65 0.00107149     71   0.18457 0.35394 0.014319
## 66 0.00106199     72   0.18350 0.35420 0.014319
## 67 0.00099296     73   0.18243 0.35554 0.014367
## 68 0.00097298     74   0.18144 0.35523 0.014382
## 69 0.00096571     75   0.18047 0.35512 0.014382
## 70 0.00094299     76   0.17950 0.35500 0.014394
## 71 0.00092604     77   0.17856 0.35488 0.014404
## 72 0.00086791     78   0.17763 0.35459 0.014378
## 73 0.00086082     80   0.17590 0.35391 0.014407
## 74 0.00083882     81   0.17504 0.35327 0.014361
## 75 0.00082185     82   0.17420 0.35324 0.014444
## 76 0.00081999     83   0.17338 0.35308 0.014445
## 77 0.00081510     84   0.17256 0.35324 0.014441
## 78 0.00080718     85   0.17174 0.35424 0.014482
## 79 0.00080253     86   0.17093 0.35433 0.014487
## 80 0.00080214     87   0.17013 0.35389 0.014462
## 81 0.00079999     88   0.16933 0.35349 0.014447
## 82 0.00077675     89   0.16853 0.35294 0.014440
## 83 0.00076692     90   0.16775 0.35440 0.014533
## 84 0.00075579     91   0.16698 0.35437 0.014539
## 85 0.00074858     92   0.16623 0.35446 0.014543
## 86 0.00070380     94   0.16473 0.35518 0.014558
```

```
## 87 0.00070000     95   0.16403 0.35524 0.014553
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC10
##  [6] HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.y HC03_VC130
## [11] HC03_VC133  HC03_VC134  HC03_VC156  HC03_VC17   HC03_VC18
## [16] HC03_VC79   HC03_VC82   HC03_VC83   HC03_VC84   HC03_VC85
## [21] HC03_VC86   HC03_VC87   HC03_VC89   HC03_VC90   HC03_VC91
## [26] Latitude    Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##            CP nsplit rel error  xerror     xstd
## 1  0.26894188      0   1.00000 1.00069 0.031342
## 2  0.15300146      1   0.73106 0.74520 0.024259
## 3  0.04905506      2   0.57806 0.59680 0.022070
## 4  0.03483358      3   0.52900 0.54997 0.021172
## 5  0.02575390      4   0.49417 0.53233 0.020738
## 6  0.02255952      5   0.46841 0.51746 0.019652
## 7  0.02195363      6   0.44585 0.50673 0.019104
## 8  0.01812404      7   0.42390 0.49817 0.018487
## 9  0.01460201      8   0.40578 0.49166 0.018312
## 10 0.01373019      9   0.39117 0.47992 0.018319
## 11 0.01318438     10   0.37744 0.46696 0.018013
## 12 0.01113299     11   0.36426 0.45942 0.018058
## 13 0.01044189     12   0.35313 0.44518 0.018120
## 14 0.00876037     13   0.34269 0.43926 0.017916
## 15 0.00771485     14   0.33393 0.43474 0.017703
## 16 0.00737522     15   0.32621 0.43186 0.017597
## 17 0.00651898     16   0.31884 0.42905 0.017726
## 18 0.00609098     17   0.31232 0.42659 0.017760
## 19 0.00582690     18   0.30623 0.42527 0.017781
## 20 0.00523612     19   0.30040 0.41941 0.017763
## 21 0.00482466     20   0.29516 0.41733 0.017481
## 22 0.00479629     21   0.29034 0.41496 0.017415
## 23 0.00455377     22   0.28554 0.41342 0.017322
## 24 0.00406023     23   0.28099 0.40965 0.016860
## 25 0.00390075     24   0.27693 0.40755 0.016925
## 26 0.00386309     25   0.27303 0.40415 0.016249
## 27 0.00373690     26   0.26916 0.40185 0.016141
## 28 0.00323021     27   0.26543 0.40039 0.016142
## 29 0.00287222     28   0.26220 0.40222 0.016234
## 30 0.00276227     29   0.25932 0.40408 0.016271
## 31 0.00273625     30   0.25656 0.40312 0.016057
## 32 0.00269698     31   0.25383 0.40185 0.016009
## 33 0.00265792     32   0.25113 0.40301 0.016015
## 34 0.00261475     33   0.24847 0.40242 0.016005
## 35 0.00259402     34   0.24586 0.40339 0.016049
```

```
## 36 0.00242374     35   0.24326 0.40262 0.016030
## 37 0.00234022     36   0.24084 0.40037 0.016073
## 38 0.00226835     37   0.23850 0.39812 0.016042
## 39 0.00217154     39   0.23396 0.39751 0.016083
## 40 0.00203158     40   0.23179 0.39745 0.016188
## 41 0.00200903     41   0.22976 0.39378 0.015957
## 42 0.00197756     43   0.22574 0.39144 0.015876
## 43 0.00192518     44   0.22376 0.39222 0.016035
## 44 0.00180554     45   0.22184 0.38858 0.015680
## 45 0.00174488     47   0.21823 0.39021 0.015813
## 46 0.00171649     48   0.21648 0.38979 0.015808
## 47 0.00171513     49   0.21476 0.38996 0.015821
## 48 0.00163250     50   0.21305 0.38989 0.015797
## 49 0.00160365     51   0.21142 0.39023 0.015849
## 50 0.00154749     52   0.20981 0.38843 0.015776
## 51 0.00145486     53   0.20827 0.38978 0.015789
## 52 0.00145420     54   0.20681 0.38791 0.015747
## 53 0.00142485     55   0.20536 0.38847 0.015760
## 54 0.00140643     56   0.20393 0.38713 0.015744
## 55 0.00139441     58   0.20112 0.38878 0.015793
## 56 0.00136615     59   0.19972 0.38831 0.015800
## 57 0.00134432     62   0.19563 0.38853 0.015798
## 58 0.00132904     63   0.19428 0.38819 0.015794
## 59 0.00132240     64   0.19295 0.38826 0.015793
## 60 0.00123928     65   0.19163 0.38879 0.015870
## 61 0.00123217     66   0.19039 0.38655 0.015883
## 62 0.00116417     67   0.18916 0.38758 0.015918
## 63 0.00115728     68   0.18799 0.38729 0.015902
## 64 0.00111347     70   0.18568 0.38667 0.015869
## 65 0.00107149     71   0.18457 0.38625 0.015867
## 66 0.00106199     72   0.18350 0.38530 0.015865
## 67 0.00099296     73   0.18243 0.38403 0.015815
## 68 0.00097298     74   0.18144 0.38438 0.015825
## 69 0.00096571     75   0.18047 0.38550 0.015859
## 70 0.00094299     76   0.17950 0.38525 0.015861
## 71 0.00092604     77   0.17856 0.38417 0.015833
## 72 0.00086791     78   0.17763 0.38617 0.015891
## 73 0.00086082     80   0.17590 0.38563 0.015861
## 74 0.00083882     81   0.17504 0.38502 0.015861
## 75 0.00082185     82   0.17420 0.38460 0.015855
## 76 0.00081999     83   0.17338 0.38535 0.015855
## 77 0.00081510     84   0.17256 0.38535 0.015855
## 78 0.00080718     85   0.17174 0.38569 0.015856
## 79 0.00080253     86   0.17093 0.38531 0.015831
## 80 0.00080214     87   0.17013 0.38565 0.015837
## 81 0.00080000     88   0.16933 0.38555 0.015841
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC10
##  [6] HC03_VC11   HC03_VC12   HC03_VC13.y HC03_VC130  HC03_VC133
```

```
## [11] HC03_VC134  HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC79
## [16] HC03_VC82   HC03_VC83   HC03_VC85   HC03_VC86   HC03_VC87
## [21] HC03_VC89   HC03_VC90   HC03_VC91   Latitude    Longitude
## [26] State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##            CP nsplit rel error  xerror     xstd
## 1  0.26894188      0   1.00000 1.00154 0.031375
## 2  0.15300146      1   0.73106 0.74193 0.024006
## 3  0.04905506      2   0.57806 0.60040 0.021839
## 4  0.03483358      3   0.52900 0.54919 0.020811
## 5  0.02575390      4   0.49417 0.52523 0.019844
## 6  0.02255952      5   0.46841 0.50853 0.019085
## 7  0.02195363      6   0.44585 0.49695 0.018337
## 8  0.01812404      7   0.42390 0.48408 0.017791
## 9  0.01460201      8   0.40578 0.47081 0.017293
## 10 0.01373019      9   0.39117 0.45858 0.017087
## 11 0.01318438     10   0.37744 0.45375 0.017034
## 12 0.01113299     11   0.36426 0.43844 0.016856
## 13 0.01044189     12   0.35313 0.43220 0.016854
## 14 0.00876037     13   0.34269 0.42338 0.016625
## 15 0.00771485     14   0.33393 0.40997 0.015717
## 16 0.00737522     15   0.32621 0.40938 0.015710
## 17 0.00651898     16   0.31884 0.40591 0.015716
## 18 0.00609098     17   0.31232 0.39755 0.015128
## 19 0.00582690     18   0.30623 0.39372 0.015127
## 20 0.00523612     19   0.30040 0.39383 0.015146
## 21 0.00482466     20   0.29516 0.39214 0.015549
## 22 0.00479629     21   0.29034 0.38968 0.015459
## 23 0.00455377     22   0.28554 0.38978 0.015523
## 24 0.00406023     23   0.28099 0.38136 0.015446
## 25 0.00390075     24   0.27693 0.37640 0.015343
## 26 0.00386309     25   0.27303 0.37738 0.015384
## 27 0.00373690     26   0.26916 0.37563 0.015264
## 28 0.00323021     27   0.26543 0.37798 0.015334
## 29 0.00287222     28   0.26220 0.37637 0.015318
## 30 0.00276227     29   0.25932 0.37129 0.014928
## 31 0.00273625     30   0.25656 0.37000 0.015139
## 32 0.00269698     31   0.25383 0.37002 0.015194
## 33 0.00265792     32   0.25113 0.37031 0.015215
## 34 0.00261475     33   0.24847 0.37041 0.015214
## 35 0.00259402     34   0.24586 0.37255 0.015257
## 36 0.00242374     35   0.24326 0.36919 0.015186
## 37 0.00234022     36   0.24084 0.36562 0.015092
## 38 0.00226835     37   0.23850 0.36872 0.015219
## 39 0.00217154     39   0.23396 0.36698 0.015176
## 40 0.00203158     40   0.23179 0.36836 0.015277
## 41 0.00200903     41   0.22976 0.37012 0.015306
## 42 0.00197756     43   0.22574 0.36879 0.015283
## 43 0.00192518     44   0.22376 0.36848 0.015262
## 44 0.00180554     45   0.22184 0.36692 0.015273
```

```
## 45 0.00174488     47   0.21823 0.36560 0.015152
## 46 0.00171649     48   0.21648 0.36524 0.015004
## 47 0.00171513     49   0.21476 0.36565 0.015005
## 48 0.00163250     50   0.21305 0.36711 0.015028
## 49 0.00160365     51   0.21142 0.36630 0.015027
## 50 0.00154749     52   0.20981 0.36669 0.015057
## 51 0.00145486     53   0.20827 0.36473 0.015007
## 52 0.00145420     54   0.20681 0.36607 0.015037
## 53 0.00142485     55   0.20536 0.36517 0.015017
## 54 0.00140643     56   0.20393 0.36588 0.015101
## 55 0.00139441     58   0.20112 0.36597 0.015103
## 56 0.00136615     59   0.19972 0.36578 0.015101
## 57 0.00134432     62   0.19563 0.36615 0.015117
## 58 0.00132904     63   0.19428 0.36606 0.015112
## 59 0.00132240     64   0.19295 0.36608 0.015129
## 60 0.00123928     65   0.19163 0.36528 0.015209
## 61 0.00123217     66   0.19039 0.36369 0.015246
## 62 0.00116417     67   0.18916 0.36574 0.015433
## 63 0.00115728     68   0.18799 0.36582 0.015474
## 64 0.00111347     70   0.18568 0.36530 0.015459
## 65 0.00107149     71   0.18457 0.36557 0.015480
## 66 0.00106199     72   0.18350 0.36576 0.015503
## 67 0.00099296     73   0.18243 0.36563 0.015522
## 68 0.00097298     74   0.18144 0.36382 0.015525
## 69 0.00096571     75   0.18047 0.36360 0.015532
## 70 0.00094299     76   0.17950 0.36442 0.015583
## 71 0.00092604     77   0.17856 0.36396 0.015583
## 72 0.00090000     78   0.17763 0.36409 0.015564
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[buildfold, -(3)],
##     method = "anova", control = rpart.control(cp = cps[j]))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC05   HC03_VC07   HC03_VC10   HC03_VC11
##  [6] HC03_VC12   HC03_VC13.y HC03_VC130  HC03_VC133  HC03_VC134
## [11] HC03_VC156  HC03_VC17   HC03_VC18   HC03_VC79   HC03_VC82
## [16] HC03_VC83   HC03_VC85   HC03_VC86   HC03_VC87   HC03_VC89
## [21] HC03_VC90   HC03_VC91   Latitude    Longitude   State
##
## Root node error: 67.293/2547 = 0.02642
##
## n= 2547
##
##           CP nsplit rel error  xerror     xstd
## 1  0.2689419      0   1.00000 1.00064 0.031322
## 2  0.1530015      1   0.73106 0.73863 0.023592
## 3  0.0490551      2   0.57806 0.58950 0.020964
## 4  0.0348336      3   0.52900 0.54224 0.020035
## 5  0.0257539      4   0.49417 0.51936 0.019720
## 6  0.0225595      5   0.46841 0.51707 0.019653
## 7  0.0219536      6   0.44585 0.50743 0.019300
## 8  0.0181240      7   0.42390 0.49991 0.019205
## 9  0.0146020      8   0.40578 0.47303 0.018059
```

```
## 10 0.0137302      9   0.39117 0.46348 0.017931
## 11 0.0131844     10   0.37744 0.45089 0.017739
## 12 0.0111330     11   0.36426 0.44466 0.017458
## 13 0.0104419     12   0.35313 0.44596 0.018508
## 14 0.0087604     13   0.34269 0.43305 0.018025
## 15 0.0077149     14   0.33393 0.42625 0.017800
## 16 0.0073752     15   0.32621 0.42354 0.017647
## 17 0.0065190     16   0.31884 0.41906 0.017394
## 18 0.0060910     17   0.31232 0.41402 0.017202
## 19 0.0058269     18   0.30623 0.41519 0.017600
## 20 0.0052361     19   0.30040 0.40711 0.017151
## 21 0.0048247     20   0.29516 0.39833 0.016984
## 22 0.0047963     21   0.29034 0.39845 0.016968
## 23 0.0045538     22   0.28554 0.40120 0.017034
## 24 0.0040602     23   0.28099 0.40489 0.017190
## 25 0.0039007     24   0.27693 0.40299 0.017150
## 26 0.0038631     25   0.27303 0.40329 0.017153
## 27 0.0037369     26   0.26916 0.40324 0.017165
## 28 0.0032302     27   0.26543 0.39544 0.016663
## 29 0.0028722     28   0.26220 0.39411 0.016679
## 30 0.0027623     29   0.25932 0.39026 0.016677
## 31 0.0027362     30   0.25656 0.39184 0.016750
## 32 0.0026970     31   0.25383 0.39184 0.016750
## 33 0.0026579     32   0.25113 0.38949 0.016636
## 34 0.0026147     33   0.24847 0.39036 0.016674
## 35 0.0025940     34   0.24586 0.39115 0.016634
## 36 0.0024237     35   0.24326 0.39163 0.016642
## 37 0.0023402     36   0.24084 0.38540 0.016184
## 38 0.0022683     37   0.23850 0.38439 0.016180
## 39 0.0021715     39   0.23396 0.38440 0.016174
## 40 0.0020316     40   0.23179 0.38257 0.016150
## 41 0.0020090     41   0.22976 0.38144 0.016116
## 42 0.0019776     43   0.22574 0.38044 0.016109
## 43 0.0019252     44   0.22376 0.38073 0.016112
## 44 0.0018055     45   0.22184 0.38247 0.016279
## 45 0.0017449     47   0.21823 0.38144 0.016274
## 46 0.0017165     48   0.21648 0.38122 0.016274
## 47 0.0017151     49   0.21476 0.38152 0.016301
## 48 0.0016325     50   0.21305 0.38260 0.016358
## 49 0.0016037     51   0.21142 0.38293 0.016395
## 50 0.0015475     52   0.20981 0.37954 0.015717
## 51 0.0014549     53   0.20827 0.37950 0.015705
## 52 0.0014542     54   0.20681 0.38034 0.015640
## 53 0.0014249     55   0.20536 0.38077 0.015664
## 54 0.0014064     56   0.20393 0.38094 0.015666
## 55 0.0013944     58   0.20112 0.38098 0.015661
## 56 0.0013662     59   0.19972 0.38007 0.015639
## 57 0.0013443     62   0.19563 0.37934 0.015632
## 58 0.0013290     63   0.19428 0.37934 0.015632
## 59 0.0013224     64   0.19295 0.37959 0.015661
## 60 0.0012393     65   0.19163 0.37752 0.015883
## 61 0.0012322     66   0.19039 0.37645 0.015845
## 62 0.0011642     67   0.18916 0.37770 0.015933
## 63 0.0011573     68   0.18799 0.37729 0.015928
```

```
## 64 0.0011135     70   0.18568 0.37734 0.015937
## 65 0.0010715     71   0.18457 0.37756 0.016005
## 66 0.0010620     72   0.18350 0.37587 0.016010
## 67 0.0010000     73   0.18243 0.37560 0.016026
```

```
#chose cp = 0.0008
tree3 = rpart(repvotes ~ .,
                  data = buildset[,-3],
                  method = "anova",
                  control = rpart.control(cp = 0.0008))

preds3 = predict(tree3,
              newdata = testset[ , -c(1,3)],
              type = "vector")

cpr3 = sum(c(preds3>.5)==c(testset[,1]>.5))/nrow(testset)


printcp(tree3)
```

```
##
## Regression tree:
## rpart(formula = repvotes ~ ., data = buildset[, -3], method = "anova",
##     control = rpart.control(cp = 8e-04))
##
## Variables actually used in tree construction:
##  [1] HC01_VC85.y HC03_VC04   HC03_VC05   HC03_VC07   HC03_VC10
##  [6] HC03_VC11   HC03_VC12   HC03_VC129  HC03_VC13.y HC03_VC130
## [11] HC03_VC131  HC03_VC133  HC03_VC134  HC03_VC14   HC03_VC15
## [16] HC03_VC18   HC03_VC76   HC03_VC77   HC03_VC79   HC03_VC81
## [21] HC03_VC82   HC03_VC84   HC03_VC85   HC03_VC86   HC03_VC87
## [26] HC03_VC90   HC03_VC91   Latitude    Longitude   State
##
## Root node error: 75.157/2830 = 0.026557
##
## n= 2830
##
##            CP nsplit rel error  xerror     xstd
## 1  0.27103707      0   1.00000 1.00085 0.029715
## 2  0.15006942      1   0.72896 0.73571 0.022044
## 3  0.06140401      2   0.57889 0.59609 0.019936
## 4  0.03531472      3   0.51749 0.55164 0.019189
## 5  0.03203458      4   0.48217 0.52686 0.018826
## 6  0.02692045      5   0.45014 0.50838 0.018486
## 7  0.01707621      6   0.42322 0.47044 0.016550
## 8  0.01637915      7   0.40614 0.45766 0.015741
## 9  0.01455037      8   0.38976 0.44766 0.015553
## 10 0.01263077      9   0.37521 0.43359 0.015315
## 11 0.01174684     10   0.36258 0.42270 0.015032
## 12 0.01123131     11   0.35084 0.41280 0.014932
## 13 0.00775256     12   0.33961 0.40408 0.014654
## 14 0.00771592     13   0.33185 0.39684 0.014547
## 15 0.00745246     14   0.32414 0.39642 0.014654
```

```
## 16 0.00725216     15   0.31668 0.39542 0.014631
## 17 0.00461573     16   0.30943 0.38024 0.014560
## 18 0.00456156     18   0.30020 0.37771 0.014534
## 19 0.00425267     19   0.29564 0.37447 0.014402
## 20 0.00420336     20   0.29139 0.37480 0.014399
## 21 0.00363156     21   0.28718 0.37612 0.014519
## 22 0.00360496     23   0.27992 0.37222 0.014227
## 23 0.00358421     24   0.27631 0.37157 0.014218
## 24 0.00348820     25   0.27273 0.37100 0.014216
## 25 0.00325680     26   0.26924 0.37027 0.014102
## 26 0.00322808     27   0.26599 0.36778 0.014044
## 27 0.00312031     28   0.26276 0.36797 0.014074
## 28 0.00301376     29   0.25964 0.36651 0.013651
## 29 0.00290753     30   0.25662 0.36583 0.013621
## 30 0.00285765     31   0.25372 0.36519 0.013687
## 31 0.00275150     32   0.25086 0.36584 0.013670
## 32 0.00259130     33   0.24811 0.36553 0.013638
## 33 0.00245805     34   0.24552 0.36417 0.013668
## 34 0.00240691     35   0.24306 0.36194 0.013607
## 35 0.00222514     36   0.24065 0.35827 0.013489
## 36 0.00222113     38   0.23620 0.35628 0.013439
## 37 0.00222038     39   0.23398 0.35627 0.013439
## 38 0.00217296     40   0.23176 0.35616 0.013446
## 39 0.00207740     41   0.22959 0.35633 0.013469
## 40 0.00204385     42   0.22751 0.35582 0.013468
## 41 0.00204384     43   0.22546 0.35684 0.013483
## 42 0.00201100     44   0.22342 0.35643 0.013503
## 43 0.00195237     45   0.22141 0.35406 0.013283
## 44 0.00194676     46   0.21946 0.35485 0.013398
## 45 0.00187613     47   0.21751 0.35412 0.013377
## 46 0.00183986     49   0.21376 0.35262 0.013201
## 47 0.00178331     50   0.21192 0.35182 0.013158
## 48 0.00164002     51   0.21014 0.35142 0.013252
## 49 0.00163768     52   0.20850 0.35202 0.013315
## 50 0.00158772     53   0.20686 0.35220 0.013316
## 51 0.00154858     55   0.20368 0.35031 0.013218
## 52 0.00154194     56   0.20213 0.35118 0.013236
## 53 0.00153562     57   0.20059 0.35118 0.013236
## 54 0.00127384     58   0.19906 0.34832 0.013258
## 55 0.00122809     59   0.19778 0.35157 0.013500
## 56 0.00122657     60   0.19655 0.35279 0.013577
## 57 0.00122284     61   0.19533 0.35279 0.013577
## 58 0.00120959     63   0.19288 0.35286 0.013577
## 59 0.00119443     64   0.19167 0.35350 0.013577
## 60 0.00115202     65   0.19048 0.35542 0.013614
## 61 0.00114033     66   0.18933 0.35655 0.013637
## 62 0.00113546     67   0.18819 0.35667 0.013688
## 63 0.00112381     68   0.18705 0.35704 0.013724
## 64 0.00109711     69   0.18593 0.35734 0.013727
## 65 0.00109699     70   0.18483 0.35661 0.013725
## 66 0.00109626     71   0.18373 0.35661 0.013725
## 67 0.00102859     72   0.18264 0.35594 0.013655
## 68 0.00101004     73   0.18161 0.35842 0.013901
## 69 0.00099825     74   0.18060 0.35848 0.013902
```

```
## 70 0.00095738     75   0.17960 0.35681 0.013865
## 71 0.00093630     76   0.17864 0.35700 0.013872
## 72 0.00090111     77   0.17771 0.35619 0.013850
## 73 0.00086940     78   0.17680 0.35345 0.013784
## 74 0.00085289     79   0.17593 0.35215 0.013744
## 75 0.00084852     80   0.17508 0.35225 0.013709
## 76 0.00083866     81   0.17423 0.35351 0.013717
## 77 0.00082664     82   0.17339 0.35330 0.013707
## 78 0.00080198     83   0.17257 0.35232 0.013688
## 79 0.00080000     84   0.17177 0.35260 0.013693
```

5. Predicting the change from 2012 to 2016 (Made by Judiel Salandanan and Rick Chen)

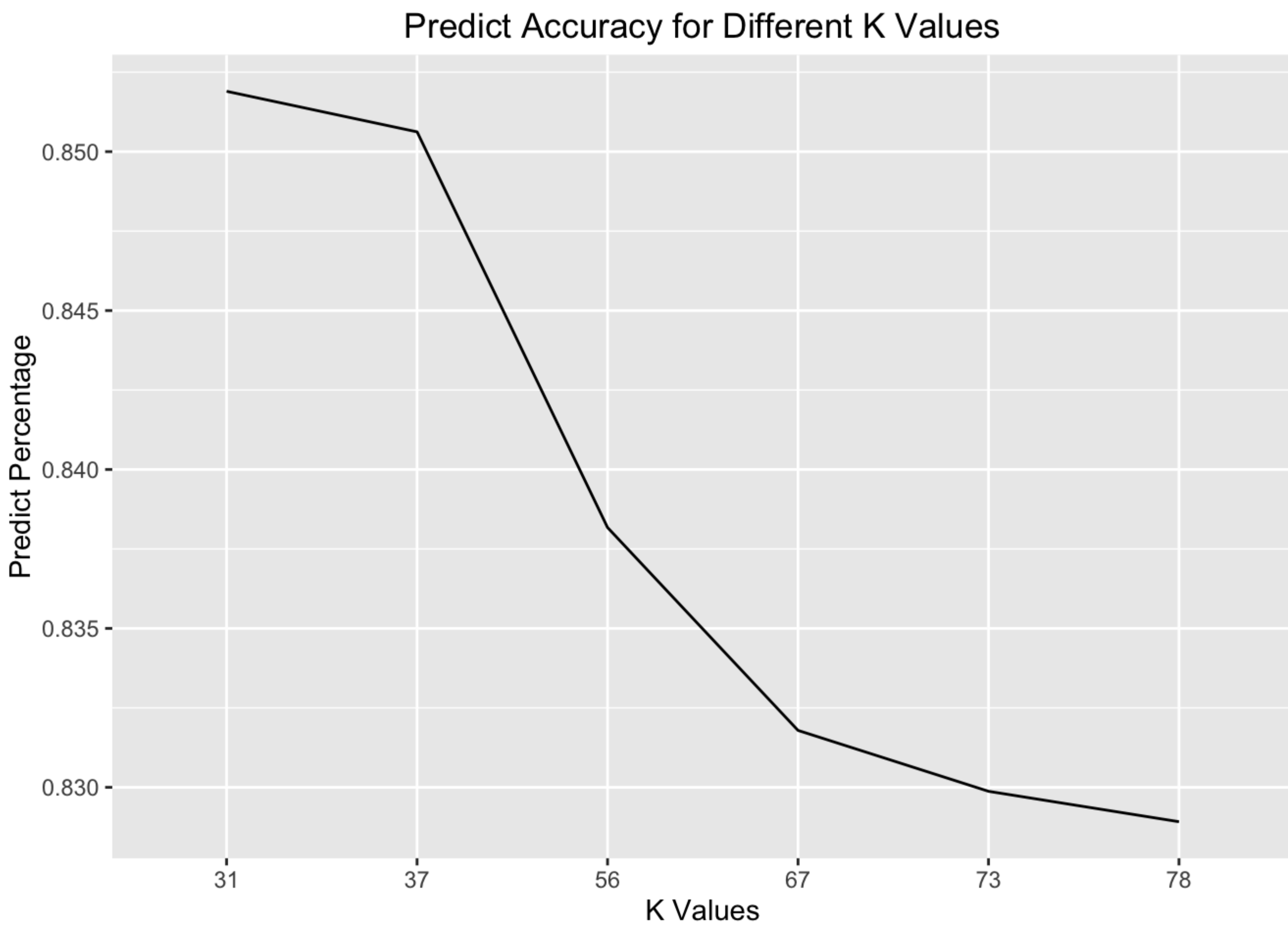
Method Use KNN method to predict 2016 election data from 2012 election data. First our team make a new dataframe “knnDF” by subsetting “”State“,”County“,”2016 Clinton Votes“,”2016 Trump Votes“,”2012 Democrat Votes“,”2012 Republican Votes“,”Longitude“,”Latitude“,”HC01_VC90“,”HC01_VC08“,”HC01_VC83“” from our primary dataframe “bigDF”. “HC01_VC90”, “HC01_VC08”, “HC01_VC83” each stands for number of bachelor degree voters, unemployed voters and rich voters. Later we organize our training set and test set. We make a training set dataframe called “trainingDF” by subseting longitude, latitude, employment, bachelor degree and wealth(rich) information of year 2012 from the “knnDF”. In contrast, our testset is made with the exact same information of year 2016. Later on, we use knn function of knn package to execute prediction. The result we get from prediction is a categorical vector with “republican” or “democrat” Moreover. We run with multiple values of k and plot accuracy for plotting part 5. The way to test accuracy of different k value is to compare the accuracy percent. From the youtube video where we learned how to use knn, it suggested us to take the value of squareroot of number of rows of our dataframe as the value of K. However, when we arbitrary choose different K values, some K values actually predict better than the K value youtube suggested. Futhermore, the prediction percentage we saw might not as good as it looks. There is a problem in machine learning called overfitting. Our team believe there is a reason on all the sources that suggest us to use the value of squareroot of number of rows of our dataframe as the value of K. The K value might not looks as fancy as other overfitting K values, but at least it is more precise. For K is the value of squareroot of number of rows of our dataframe, which in our case is 56, we have our accuracy percentage 0.8397702.

By comparing our to predictions, we have 0.82823 for using 2016 results to predict 2016 election and 0.8397702 for using 2012 election results to predict 2016 election.

```
#KNN
library(class)

normalize <- function(x) {
num <- x - min(x)
denom <- max(x) - min(x)
return (num/denom)
}

knnDF = bigDF[, c("State", "County","2016 Clinton Votes","2016 Trump Votes","2012 Democrat Votes","2012 Republican Votes", "Longit
ude", "Latitude", "HC01_VC90", "HC01_VC08", "HC01_VC83")]
knnDF=na.omit(knnDF)
names(knnDF) = c("State", "County", "dem.2016", "rep.2016", "dem.2012","rep.2012", "lon","lat", "bachelor", "unemployed","rich")
knnDF$winner.2012 = factor(ifelse(knnDF$dem.2012 > knnDF$rep.2012, 1, 2), levels = c(1,2), labels = c("Democrat", "Republican"))
knnDF$winner.2016 = factor(ifelse(knnDF$dem.2016 > knnDF$rep.2016, 1, 2), levels = c(1,2), labels = c("Democrat", "Republican"))
knnDF[,7] = as.numeric(knnDF[,7])
knnDF[,8] = as.numeric(knnDF[,8])
knnDF[,9] = as.numeric(knnDF[,9])
```

```
knnDF[,10] = as.numeric(knnDF[,10])
knnDF[,11] = as.numeric(knnDF[,11])

# training and test data do not have the labels of who won

# training data is 2012
trainingDF = knnDF[,c(7,8,9,10,11)]
# testing data is 2016
testDF = knnDF[,c(7,8,9,10,11)]

train_target = knnDF[,12]
test_target = knnDF[,13]
k = round(sqrt(nrow(knnDF)))


election.predictor.2016 = knn(train = trainingDF, test = testDF, cl = train_target, k = k)

# run with multiple values of k and plot accuracy for plotting part 5
# test different k values,31， 37， 56， 67， 73， 78
election.predictor.2016.k31 = knn(train = trainingDF, test = testDF, cl = train_target, k = 31)
election.predictor.2016.k37 = knn(train = trainingDF, test = testDF, cl = train_target, k = 37)
election.predictor.2016.k56 = knn(train = trainingDF, test = testDF, cl = train_target, k = 56)
election.predictor.2016.k67 = knn(train = trainingDF, test = testDF, cl = train_target, k = 67)
election.predictor.2016.k73 = knn(train = trainingDF, test = testDF, cl = train_target, k = 73)
election.predictor.2016.k78 = knn(train = trainingDF, test = testDF, cl = train_target, k = 78)

#see the difference of the accuracy of different k values.
accuracy.table.k31 = table(test_target, election.predictor.2016.k31)
accuracy.percent.k31 = ( accuracy.table.k31[1,1] + accuracy.table.k31[2,2] ) / sum(accuracy.table.k31)

accuracy.table.k37 = table(test_target, election.predictor.2016.k37)
accuracy.percent.k37 = ( accuracy.table.k37[1,1] + accuracy.table.k37[2,2] ) / sum(accuracy.table.k37)

accuracy.table.k56 = table(test_target, election.predictor.2016.k56)
accuracy.percent.k56 = ( accuracy.table.k56[1,1] + accuracy.table.k56[2,2] ) / sum(accuracy.table.k56)

accuracy.table.k67 = table(test_target, election.predictor.2016.k67)
accuracy.percent.k67 = ( accuracy.table.k67[1,1] + accuracy.table.k67[2,2] ) / sum(accuracy.table.k67)

accuracy.table.k73 = table(test_target, election.predictor.2016.k73)
accuracy.percent.k73 = ( accuracy.table.k73[1,1] + accuracy.table.k73[2,2] ) / sum(accuracy.table.k73)

accuracy.table.k78 = table(test_target, election.predictor.2016.k78)
accuracy.percent.k78 = ( accuracy.table.k78[1,1] + accuracy.table.k78[2,2] ) / sum(accuracy.table.k78)

# Put all the accuracy of different K values into a data frame for plotting purpose.

accuframe1=data.frame(c(accuracy.percent.k31,accuracy.percent.k37,accuracy.percent.k56,accuracy.percent.k67,accuracy.percent.k73,accur
acy.percent.k78))

accuframe2=data.frame(c("31","37","56","67","73","78"))

accuframe=cbind(accuframe1,accuframe2)
names(accuframe)= c("Precentages", "K Values")
```

```
ggplot(data=accuframe,aes(group=1))+geom_line(aes(x= accuframe$`K Values`, y= accuframe$Precentages))+labs(x= "K Values", y= "Pred
ict Percentage" , title= "Predict Accuracy for Different K Values")
```




```
# verifying accuracy
accuracy.table = table(test_target, election.predictor.2016)

accuracy.percent = ( accuracy.table[1,1] + accuracy.table[2,2] ) / sum(accuracy.table)
```

6 Discussion

After adjusting the radius of each circle, the map suggested that most votes are distributed on the west and east coast.There are more portion of vote for Clinton on west coast, and there are more blue for west coast and middle. And most dots are evenly distributed. When we look at our predictor from step 4, there are a lot of counties that are having close percentage distribution for Clinton and Trump. And from the map, most of the dots in the middle and east coast are having same portion of area in pie. The predictor suggested Trump will have about 80 percent rate to win, and the pies on map also shows that Trump will have more portion for vote in each county. Therefore, we believe our predictors did well.

7 References

1. Mowry, Curtis, Adam Pimentel, Elizabeth Sparks, and Brittany Hanlon. “Materials Characterization Activities https://cran.r-project.org/web/packages/mapplots/mapplots.pdf
2. “R: Value Matching.” R: Value Matching. N.p., n.d. Web. 11 Dec. 2016. https://stat.ethz.ch/R-manual/R-devel/library/base/html/match.html.
3. “State Abbreviations.” States and Capitals. N.p., n.d. Web. 11 Dec. 2016. http://www.50states.com/abbreviations.htm. 4.“R: Get and Set Row Names for Data Frames.” R: Get and Set Row

Names for Data Frames. N.p., n.d. Web. 11 Dec. 2016. https://stat.ethz.ch/R-manual/R-devel/library/base/html/row.names.html.

4. “Aggregate – A Powerful Tool for Data Frame in R.” R-bloggers. N.p., 15 Nov. 2015. Web. 11 Dec. 2016. https://www.r-bloggers.com/aggregate-a-powerful-tool-for-data-frame-in-r/.
5. “Adding a Legend to a Plot.” R-bloggers. N.p., 20 July 2009. Web. 11 Dec. 2016. https://www.r-bloggers.com/adding-a-legend-to-a-plot/.
6. “Mastering R Plot – Part 1: Colors, Legends and Lines.” R-bloggers. N.p., 19 Jan. 2016. Web. 11 Dec. 2016. https://www.r-bloggers.com/mastering-r-plot-part-1-colors-legends-and-lines/.
7. “R: Maxima and Minima.” R: Maxima and Minima. N.p., n.d. Web. 11 Dec. 2016. https://stat.ethz.ch/R-manual/R-devel/library/base/html/Extremes.html.
8. “R Tutorial Series: Simple Linear Regression.” R-bloggers. N.p., 12 Feb. 2015. Web. 11 Dec. 2016. https://www.r-bloggers.com/r-tutorial-series-simple-linear-regression/.
9. “Quick-R.” Quick-R: Multiple Regression. N.p., n.d. Web. 11 Dec. 2016. http://www.statmethods.net/stats/regression.html.
10. “Regression Models in R.” Regression Models in R. N.p., n.d. Web. 11 Dec. 2016. http://tutorials.iq.harvard.edu/R/Rstatistics/Rstatistics.html.
11. 1. “Machine Learning in R for Beginners.” DataCamp Community. N.p., n.d. Web. 11 Dec. 2016. https://www.datacamp.com/community/tutorials/machine-learning-in-r#gs.9IUrRxU.
12. Rdjalayer. “R - KNN - K Nearest Neighbor (part 1).” YouTube. YouTube, 09 Mar. 2015. Web. 11 Dec. 2016. https://www.youtube.com/watch?v=GtgJEVxl7DY
13. Rdjalayer. “R - KNN - K Nearest Neighbor (part 2).” YouTube. YouTube, 09 Mar. 2015. Web. 11 Dec. 2016. https://www.youtube.com/watch?v=DkLNb0CXw84.

This notes is for our team to see the coresponding catagory in the original file for convenience purpose. DP02: HC01_VC84,Estimate; EDUCATIONAL ATTAINMENT - Population 25 years and over

HC01_VC85,Estimate; EDUCATIONAL ATTAINMENT - Less than 9th grade

HC01_VC86,“Estimate; EDUCATIONAL ATTAINMENT - 9th to 12th grade, no diploma”

HC01_VC87,Estimate; EDUCATIONAL ATTAINMENT - High school graduate (includes equivalency)

HC01_VC88,“Estimate; EDUCATIONAL ATTAINMENT - Some college, no degree”

HC01_VC89,Estimate; EDUCATIONAL ATTAINMENT - Associate’s degree

HC01_VC90,Estimate; EDUCATIONAL ATTAINMENT - Bachelor’s degree

HC01_VC91,Estimate; EDUCATIONAL ATTAINMENT - Graduate or professional degree

HC01_VC128,Estimate; PLACE OF BIRTH - Total population

HC01_VC129,Estimate; PLACE OF BIRTH – Native

HC01_VC130,Estimate; PLACE OF BIRTH - Native - Born in United States

HC01_VC131,Estimadte; PLACE OF BIRTH - Native - Born in United States - State of residence

HC01_VC132,Estimate; PLACE OF BIRTH - Native - Born in United States - Different state

HC01_VC133,“Estimate; PLACE OF BIRTH - Native - Born in Puerto Rico, U.S. Island areas, or born abroad to American parent(s)”

HC02_VC134,Estimate Margin of Error; PLACE OF BIRTH - Foreign born

DPO3:

HC01_VC04,Estimate; EMPLOYMENT STATUS - Population 16 years and over

HC01_VC05,Estimate; EMPLOYMENT STATUS - In labor force

HC01_VC06,Estimate; EMPLOYMENT STATUS - In labor force - Civilian labor force

HC01_VC07,Estimate; EMPLOYMENT STATUS - In labor force - Civilian labor force – Employed

HC01_VC08,Estimate; EMPLOYMENT STATUS - In labor force - Civilian labor force – Unemployed

HC01_VC10,Estimate; EMPLOYMENT STATUS - Not in labor force

HC01_VC74,Estimate; INCOME AND BENEFITS (IN 2010 INFLATION-ADJUSTED DOLLARS) - Total households

HC01_VC75,“Estimate; INCOME AND BENEFITS (IN 2010 INFLATION-ADJUSTED DOLLARS) - Less than $10,000”

HC01_VC76,“Estimate; INCOME AND BENEFITS (IN 2010 INFLATION-ADJUSTED DOLLARS) - $10,000 to $14,999”

HC01_VC77,“Estimate; INCOME AND BENEFITS (IN 2010 INFLATION-ADJUSTED DOLLARS) - $15,000 to $24,999”

HC01_VC78,“Estimate; INCOME AND BENEFITS (IN 2010 INFLATION-ADJUSTED DOLLARS) - $25,000 to $34,999”

HC01_VC79,“Estimate; INCOME AND BENEFITS (IN 2010 INFLATION-ADJUSTED DOLLARS) - $35,000 to $49,999”

HC01_VC80,“Estimate; INCOME AND BENEFITS (IN 2010 INFLATION-ADJUSTED DOLLARS) - $50,000 to $74,999”

HC02_VC81,“Estimate Margin of Error; INCOME AND BENEFITS (IN 2010 INFLATION-ADJUSTED DOLLARS) - $75,000 to $99,999”

HC01_VC82,“Estimate; INCOME AND BENEFITS (IN 2010 INFLATION-ADJUSTED DOLLARS) - $100,000 to $149,999”

HC01_VC83,“Estimate; INCOME AND BENEFITS (IN 2010 INFLATION-ADJUSTED DOLLARS) - $150,000 to $199,999”

HC01_VC84,“Estimate; INCOME AND BENEFITS (IN 2010 INFLATION-ADJUSTED DOLLARS) - $200,000 or more”