Analysing US Presidential Election

Data Input

setwd("/Users/akshata/Desktop/Digging Into Data/Project")  
results<-read.csv("primary\_results.csv")  
county.demo<-read.csv("county\_facts.csv")  
dictionary <- read\_csv("county\_facts\_dictionary.csv")  
head(results)

## state state\_abbreviation county fips party candidate votes  
## 1 Alabama AL Autauga 1001 Republican Donald Trump 5387  
## 2 Alabama AL Autauga 1001 Republican Ted Cruz 2482  
## 3 Alabama AL Autauga 1001 Republican Marco Rubio 1785  
## 4 Alabama AL Autauga 1001 Republican Ben Carson 1764  
## 5 Alabama AL Autauga 1001 Republican John Kasich 421  
## 6 Alabama AL Autauga 1001 Democrat Hillary Clinton 2387  
## fraction\_votes  
## 1 0.445  
## 2 0.205  
## 3 0.148  
## 4 0.146  
## 5 0.035  
## 6 0.800

head(county.demo)

## fips area\_name state\_abbreviation PST045214 PST040210 PST120214  
## 1 0 United States 318857056 308758105 3.3  
## 2 1000 Alabama 4849377 4780127 1.4  
## 3 1001 Autauga County AL 55395 54571 1.5  
## 4 1003 Baldwin County AL 200111 182265 9.8  
## 5 1005 Barbour County AL 26887 27457 -2.1  
## 6 1007 Bibb County AL 22506 22919 -1.8  
## POP010210 AGE135214 AGE295214 AGE775214 SEX255214 RHI125214 RHI225214  
## 1 308745538 6.2 23.1 14.5 50.8 77.4 13.2  
## 2 4779736 6.1 22.8 15.3 51.5 69.7 26.7  
## 3 54571 6.0 25.2 13.8 51.4 77.9 18.7  
## 4 182265 5.6 22.2 18.7 51.2 87.1 9.6  
## 5 27457 5.7 21.2 16.5 46.6 50.2 47.6  
## 6 22915 5.3 21.0 14.8 45.9 76.3 22.1  
## RHI325214 RHI425214 RHI525214 RHI625214 RHI725214 RHI825214 POP715213  
## 1 1.2 5.4 0.2 2.5 17.4 62.1 84.9  
## 2 0.7 1.3 0.1 1.5 4.1 66.2 85.0  
## 3 0.5 1.1 0.1 1.8 2.7 75.6 85.0  
## 4 0.7 0.9 0.1 1.6 4.6 83.0 82.1  
## 5 0.6 0.5 0.2 0.9 4.5 46.6 84.8  
## 6 0.4 0.2 0.1 0.9 2.1 74.5 86.6  
## POP645213 POP815213 EDU635213 EDU685213 VET605213 LFE305213 HSG010214  
## 1 12.9 20.7 86.0 28.8 21263779 25.5 133957180  
## 2 3.5 5.2 83.1 22.6 388865 24.2 2207912  
## 3 1.6 3.5 85.6 20.9 5922 26.2 22751  
## 4 3.6 5.5 89.1 27.7 19346 25.9 107374  
## 5 2.9 5.0 73.7 13.4 2120 24.6 11799  
## 6 1.2 2.1 77.5 12.1 1327 27.6 8978  
## HSG445213 HSG096213 HSG495213 HSD410213 HSD310213 INC910213 INC110213  
## 1 64.9 26.0 176700 115610216 2.63 28155 53046  
## 2 69.7 15.9 122500 1838683 2.55 23680 43253  
## 3 76.8 8.3 136200 20071 2.71 24571 53682  
## 4 72.6 24.4 168600 73283 2.52 26766 50221  
## 5 67.7 10.6 89200 9200 2.66 16829 32911  
## 6 79.0 7.3 90500 7091 3.03 17427 36447  
## PVY020213 BZA010213 BZA110213 BZA115213 NES010213 SBO001207 SBO315207  
## 1 15.4 7488353 118266253 2.0 23005620 27092908 7.1  
## 2 18.6 97578 1603100 1.1 311578 382350 14.8  
## 3 12.1 817 10120 2.1 2947 4067 15.2  
## 4 13.9 4871 54988 3.7 16508 19035 2.7  
## 5 26.7 464 6611 -5.6 1546 1667 0.0  
## 6 18.1 275 3145 7.5 1126 1385 14.9  
## SBO115207 SBO215207 SBO515207 SBO415207 SBO015207 MAN450207 WTN220207  
## 1 0.9 5.7 0.1 8.3 28.8 5319456312 4174286516  
## 2 0.8 1.8 0.1 1.2 28.1 112858843 52252752  
## 3 0.0 1.3 0.0 0.7 31.7 0 0  
## 4 0.4 1.0 0.0 1.3 27.3 1410273 0  
## 5 0.0 0.0 0.0 0.0 27.0 0 0  
## 6 0.0 0.0 0.0 0.0 0.0 0 0  
## RTN130207 RTN131207 AFN120207 BPS030214 LND110210 POP060210  
## 1 3917663456 12990 613795732 1046363 3531905.43 87.4  
## 2 57344851 12364 6426342 13369 50645.33 94.4  
## 3 598175 12003 88157 131 594.44 91.8  
## 4 2966489 17166 436955 1384 1589.78 114.6  
## 5 188337 6334 0 8 884.88 31.0  
## 6 124707 5804 10757 19 622.58 36.8

Generating the Republican Data Frames

#setwd("/Users/akshata/Desktop/Digging Into Data/Project")  
results<-read.csv("primary\_results.csv")  
#Separating Republican Results  
Republican.votes<-results %>%  
 filter(party=="Republican")  
  
#Finding countywise winners for Republicans  
countyWinnerR<-Republican.votes %>%  
 group\_by(state, state\_abbreviation,county,fips) %>%  
 summarize(winner = candidate[which.max(fraction\_votes)],  
 fraction\_votes = max(fraction\_votes),  
 votes = max(votes))  
countyWinnerR$winner<-factor(countyWinnerR$winner)  
head(countyWinnerR)

## Source: local data frame [6 x 7]  
## Groups: state, state\_abbreviation, county [6]  
##   
## state state\_abbreviation county fips winner fraction\_votes  
## (fctr) (fctr) (fctr) (int) (fctr) (dbl)  
## 1 Alabama AL Autauga 1001 Donald Trump 0.445  
## 2 Alabama AL Baldwin 1003 Donald Trump 0.469  
## 3 Alabama AL Barbour 1005 Donald Trump 0.501  
## 4 Alabama AL Bibb 1007 Donald Trump 0.494  
## 5 Alabama AL Blount 1009 Donald Trump 0.487  
## 6 Alabama AL Bullock 1011 Donald Trump 0.565  
## Variables not shown: votes (int)

#Merging county winners and demographics  
votesR <- merge(countyWinnerR, county.demo, by = c("fips", "state\_abbreviation"))  
  
#Creating Train and Test sets   
index <- 1:nrow(votesR)  
testindex <- sample(index, trunc(length(index)/5))  
testR <- votesR[testindex,]  
trainR <- votesR[-testindex,]

Generating the Democratic Data Frames

#Separating Democratic Results  
Democrats.votes<-results %>%  
 filter(party=="Democrat")  
  
#Finding countywise winners for Democrats  
countyWinnerD<-Democrats.votes %>%  
 group\_by(state,state\_abbreviation,county,fips) %>%  
 summarize(winner = candidate[which.max(fraction\_votes)],  
 fraction\_votes = max(fraction\_votes),  
 votes = max(votes))  
countyWinnerD$winner<-factor(countyWinnerD$winner)  
head(countyWinnerD)

## Source: local data frame [6 x 7]  
## Groups: state, state\_abbreviation, county [6]  
##   
## state state\_abbreviation county fips winner fraction\_votes  
## (fctr) (fctr) (fctr) (int) (fctr) (dbl)  
## 1 Alabama AL Autauga 1001 Hillary Clinton 0.800  
## 2 Alabama AL Baldwin 1003 Hillary Clinton 0.647  
## 3 Alabama AL Barbour 1005 Hillary Clinton 0.906  
## 4 Alabama AL Bibb 1007 Hillary Clinton 0.755  
## 5 Alabama AL Blount 1009 Hillary Clinton 0.551  
## 6 Alabama AL Bullock 1011 Hillary Clinton 0.913  
## Variables not shown: votes (int)

#Merging county winners and demographics  
votesD <- merge(countyWinnerD, county.demo, by = c("fips", "state\_abbreviation"))  
  
#Creating Train and Test sets   
index <- 1:nrow(votesD)  
testindex <- sample(index, trunc(length(index)/5))  
testD <- votesD[testindex,]  
trainD <- votesD[-testindex,]

Analysing Trends

library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

library(rgeos)

## Warning: package 'rgeos' was built under R version 3.2.5

## rgeos version: 0.3-19, (SVN revision 524)  
## GEOS runtime version: 3.4.2-CAPI-1.8.2 r3921   
## Linking to sp version: 1.1-1   
## Polygon checking: TRUE

library(rgdal)  
library(maptools)

## Checking rgeos availability: TRUE

library(maps)

##   
## # maps v3.1: updated 'world': all lakes moved to separate new #  
## # 'lakes' database. Type '?world' or 'news(package="maps")'. #

##   
## Attaching package: 'maps'

## The following object is masked from 'package:plyr':  
##   
## ozone

library(RColorBrewer)  
library(gpclib)

## General Polygon Clipper Library for R (version 1.5-5)  
## Type 'class ? gpc.poly' for help

##   
## Attaching package: 'gpclib'

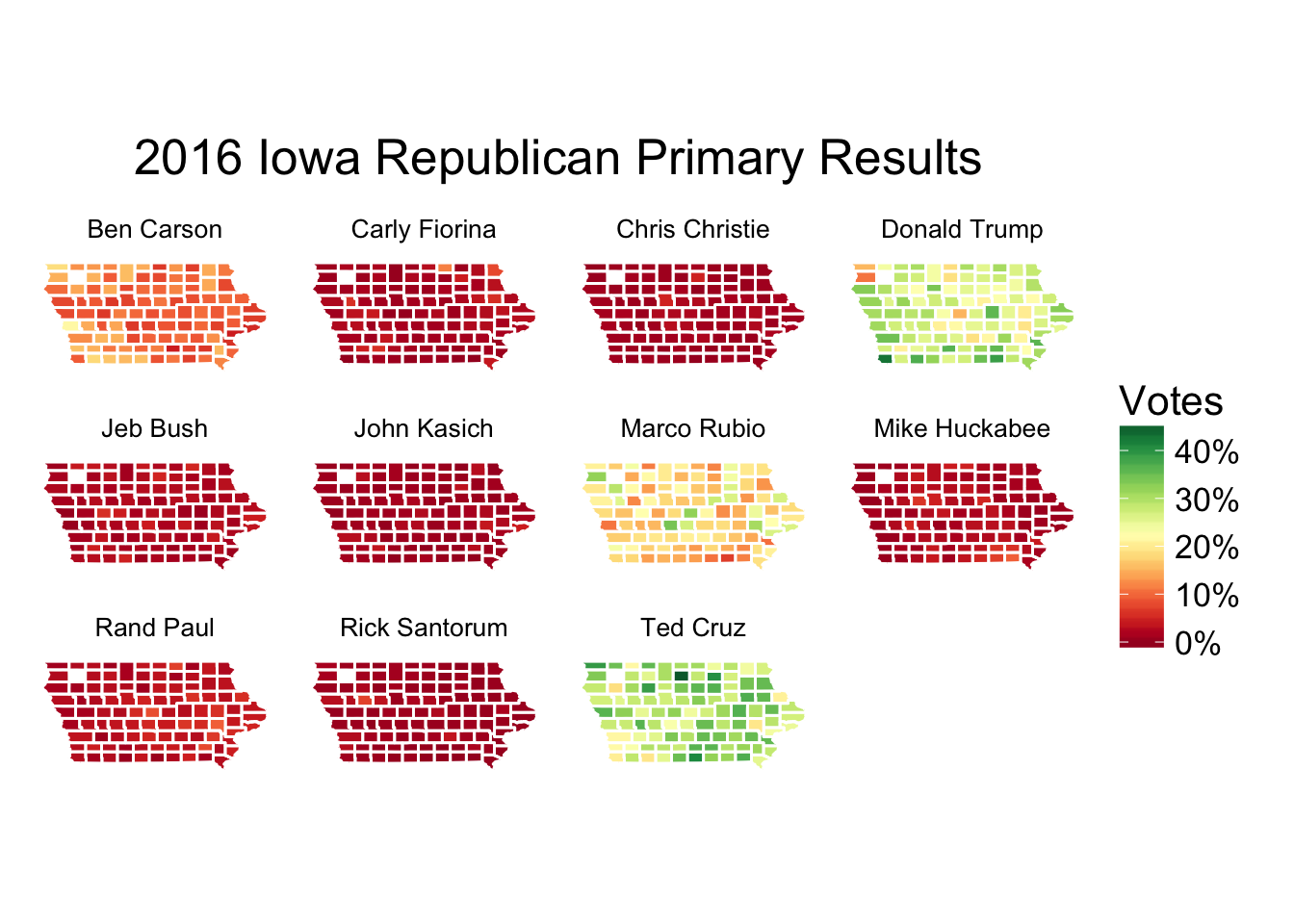
## The following objects are masked from 'package:rgeos':  
##   
## append.poly, area.poly, get.bbox, get.pts, read.polyfile,  
## scale.poly, triangulate, tristrip, write.polyfile

stateAbbreviation <- "IA"  
primaryResults <- read.csv("primary\_results.csv")  
counties <- readOGR(dsn="county\_shapefiles", layer="cb\_2014\_us\_county\_500k")

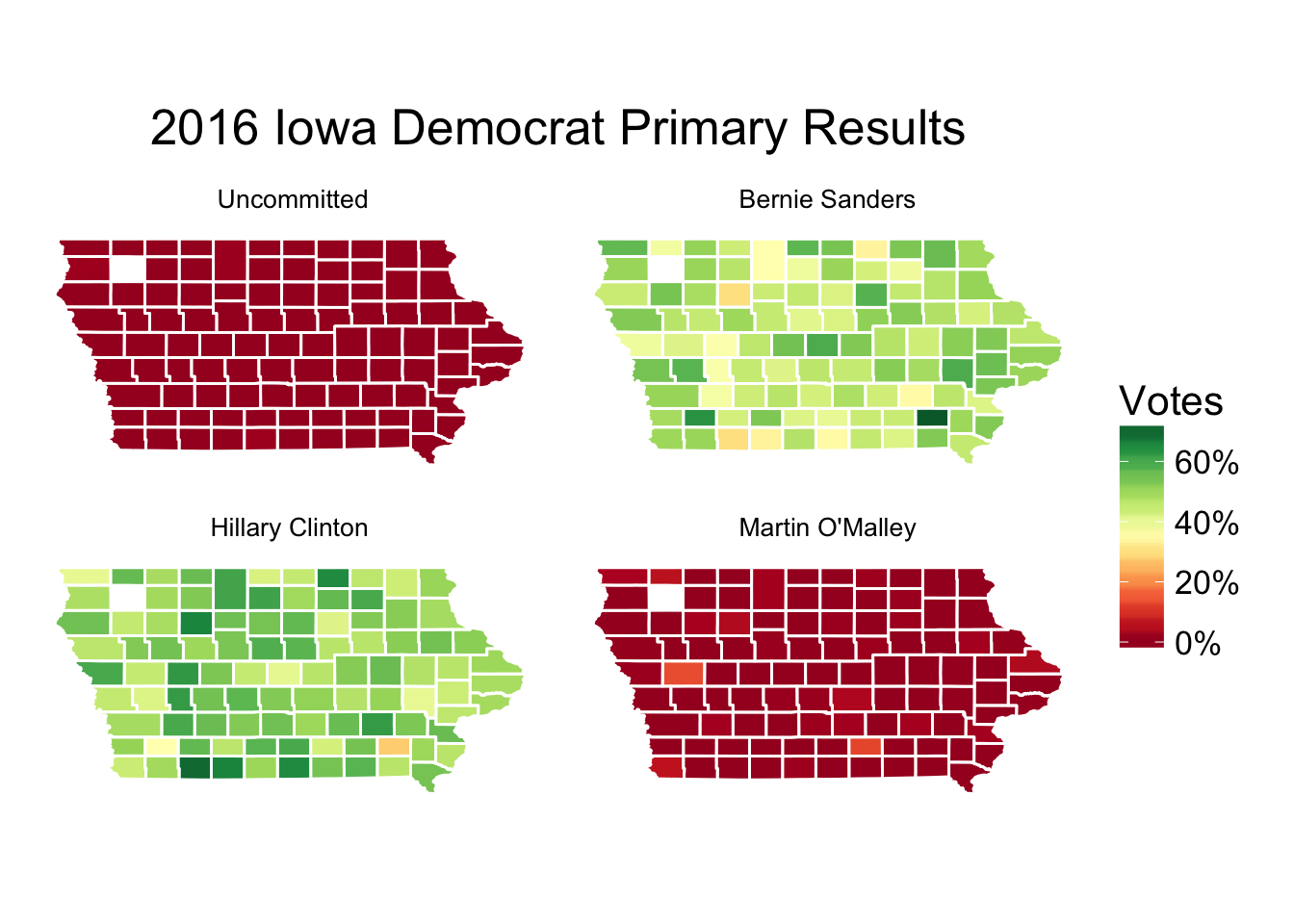
## OGR data source with driver: ESRI Shapefile   
## Source: "county\_shapefiles", layer: "cb\_2014\_us\_county\_500k"  
## with 3233 features  
## It has 9 fields

## Warning in readOGR(dsn = "county\_shapefiles", layer =  
## "cb\_2014\_us\_county\_500k"): Z-dimension discarded

counties@data$id <- rownames(counties@data)  
counties.points <- fortify(counties, region="id")  
counties.df <- join(counties.points, counties@data, by="id")  
  
stateFips <- state.fips$fips[state.fips$abb==stateAbbreviation]  
state <- primaryResults[primaryResults$state\_abbreviation==stateAbbreviation,]$state[[1]]  
stateCounties <- counties.df[counties.df$STATEFP==stateFips,]  
  
Plot\_Map <- function(stateCounties, primaryResults, stateAbbreviation, party)  
{  
stateResults <- merge(stateCounties, primaryResults[primaryResults$state\_abbreviation==stateAbbreviation & primaryResults$party==party,], by.x="NAME", by.y="county")  
stateResults <- stateResults[order(stateResults$order),]  
  
p <- ggplot(stateResults) +   
 aes(long,lat,group=group,fill=fraction\_votes) +   
 geom\_polygon() +  
 geom\_path(color="white") +  
 facet\_wrap(~candidate) +  
 coord\_equal() +  
 scale\_fill\_gradientn(name="Votes",  
 colours=brewer.pal(11,"RdYlGn"),  
 labels=percent) +   
 theme\_light(base\_size=16) +  
 theme(strip.text.x = element\_text(size=10, colour="black"),  
 strip.background = element\_rect(colour="white", fill="white"),  
 axis.line=element\_blank(),axis.text.x=element\_blank(),  
 axis.text.y=element\_blank(),axis.ticks=element\_blank(),  
 axis.title.x=element\_blank(),axis.title.y=element\_blank(),  
 panel.background=element\_blank(),panel.border=element\_blank(),panel.grid.major=element\_blank(),  
 panel.grid.minor=element\_blank(),plot.background=element\_blank()) +  
 ggtitle(paste("2016", state, party, "Primary Results"))  
return(p)  
}  
  
plot\_R<-Plot\_Map(stateCounties, primaryResults, stateAbbreviation, "Republican")  
print(plot\_R)



plot\_D<-Plot\_Map(stateCounties, primaryResults, stateAbbreviation, "Democrat")  
print(plot\_D)

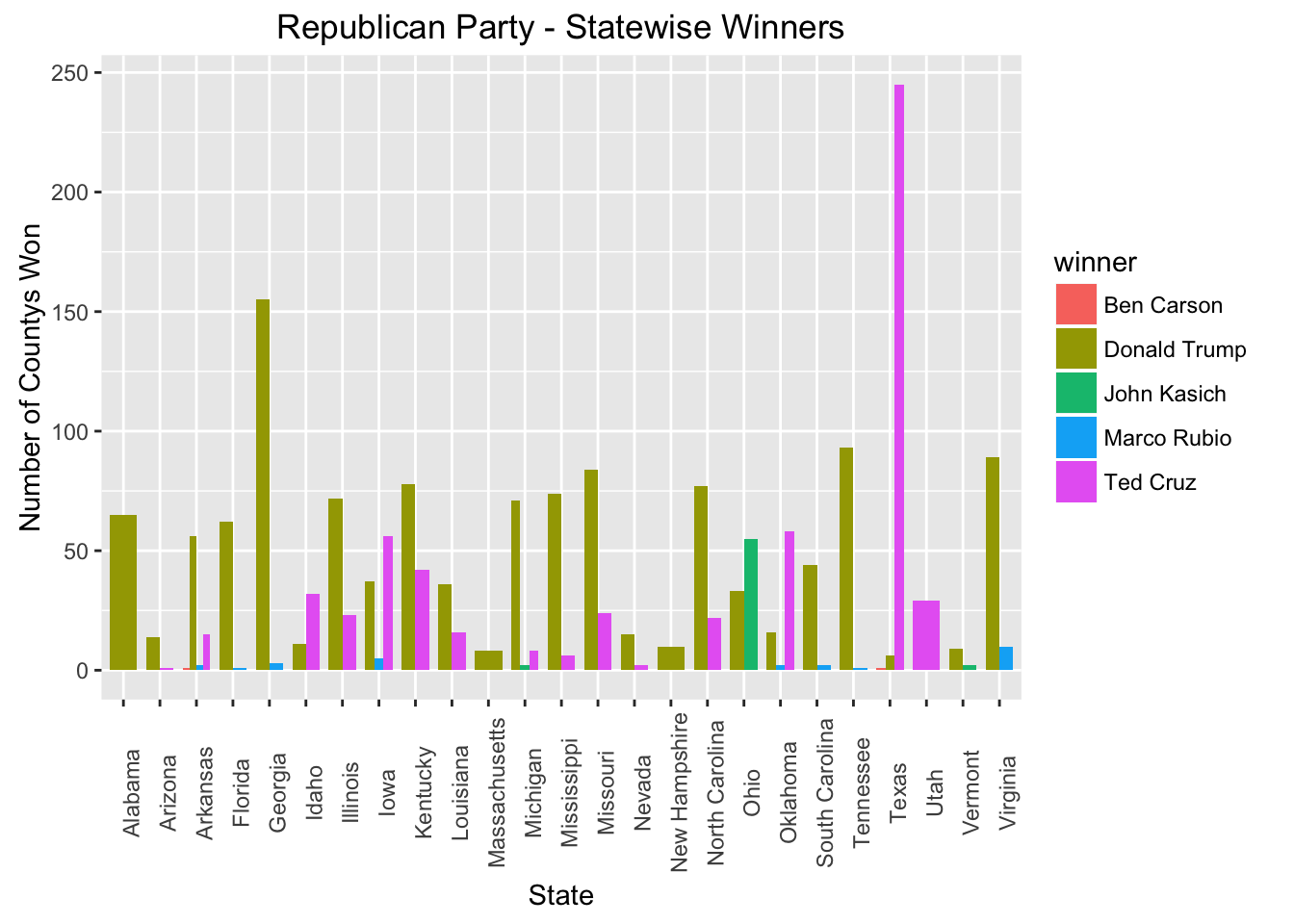


Number of counties won per state for Republicans

stateCountR<-sqldf('select state,winner,count(winner) as countyswon from countyWinnerR group by state,winner')

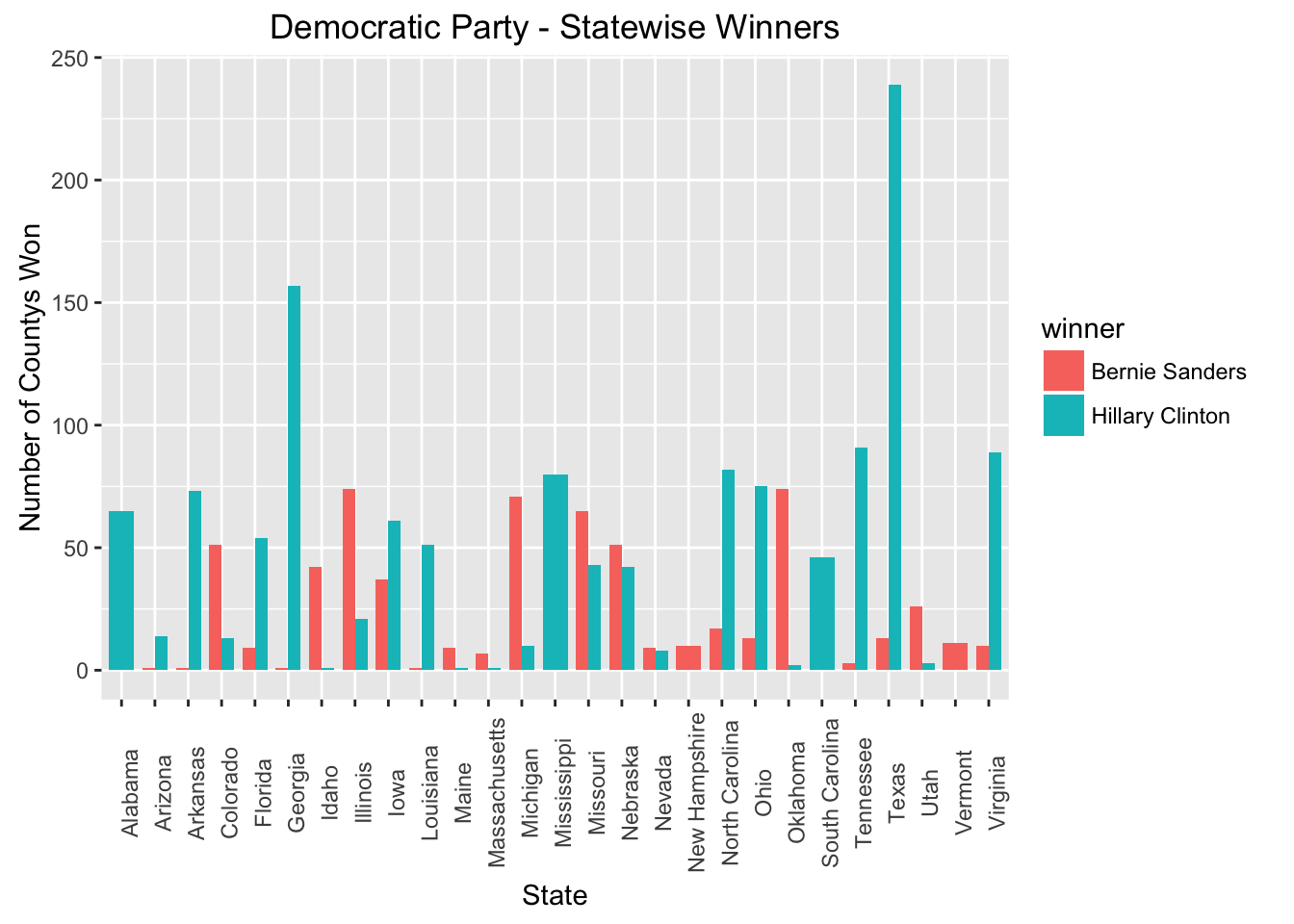
## Loading required package: tcltk

ggplot(data = stateCountR, aes(x = state , y = countyswon , group = winner, fill = winner)) + geom\_bar(stat = "identity", width = 0.75, position = "dodge") + ggtitle("Republican Party - Statewise Winners") + theme(axis.text.x = element\_text(angle = 90))+ ylab("Number of Countys Won") + xlab("State")



Number of counties won per state for Democrats

stateCountD<-sqldf('select state,winner,count(winner) as countyswon from countyWinnerD group by state,winner')  
ggplot(data = stateCountD, aes(x = state , y = countyswon , group = winner, fill = winner)) + geom\_bar(stat = "identity", width = 0.75, position = "dodge")+ ggtitle("Democratic Party - Statewise Winners") + theme(axis.text.x = element\_text(angle = 90)) + ylab("Number of Countys Won") + xlab("State")

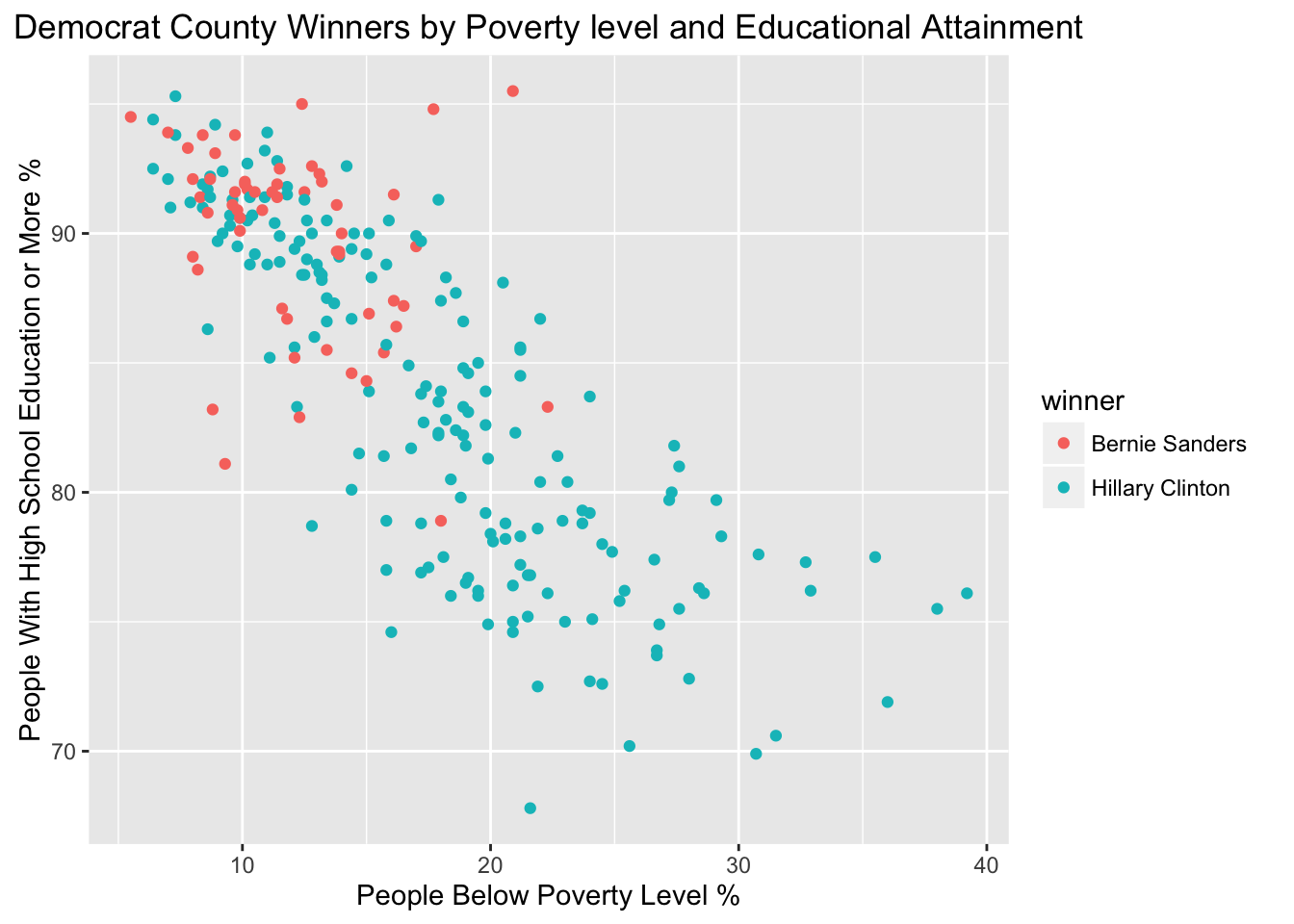


County wins plotted against county demographics

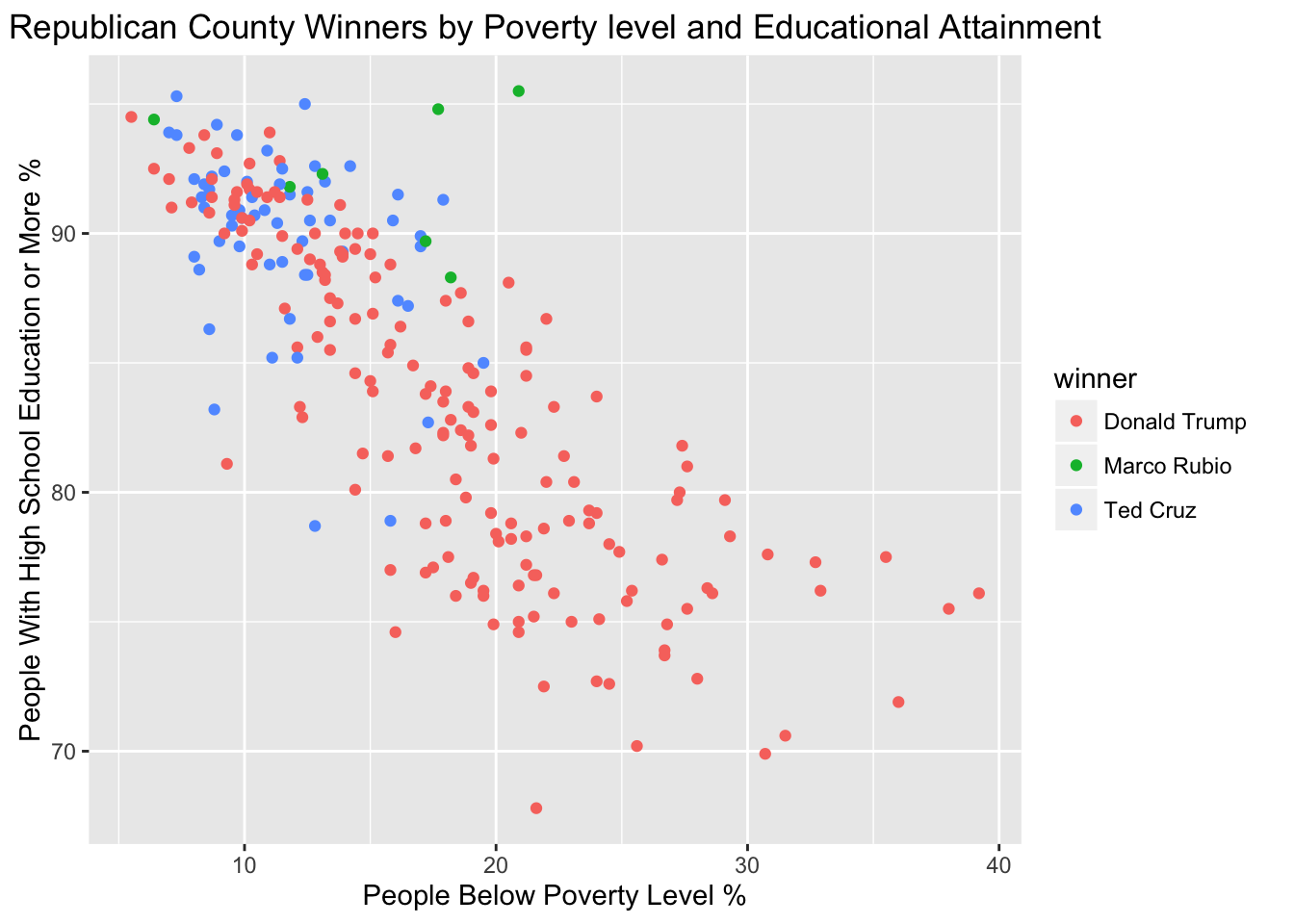
Demographics <- county.demo %>%  
 filter(state\_abbreviation %in% c("IA", "NV", "SC", "AL", "NH")) %>%   
 select(state\_abbreviation = state\_abbreviation, county = area\_name,   
 median\_income = INC110213, per\_capita\_income = INC910213, hispanic = RHI725214, poverty=PVY020213, white= RHI825214, college = EDU685213, highschool = EDU635213, density = POP060210, female = SEX255214, black=RHI225214) %>%   
 mutate(county = gsub(" County", "", county))  
  
countyWinnerD\_Demographics <- merge(countyWinnerD, Demographics, by = c("state\_abbreviation", "county"))  
countyWinnerR\_Demographics <- merge(countyWinnerR, Demographics, by = c("state\_abbreviation", "county"))

Effect of Education and Population below poverty level

ggplot(data=countyWinnerD\_Demographics, aes(x = poverty , y = highschool,color = winner)) + geom\_point()+ ggtitle("Democrat County Winners by Poverty level and Educational Attainment") + xlab("People Below Poverty Level %") + ylab("People With High School Education or More %")

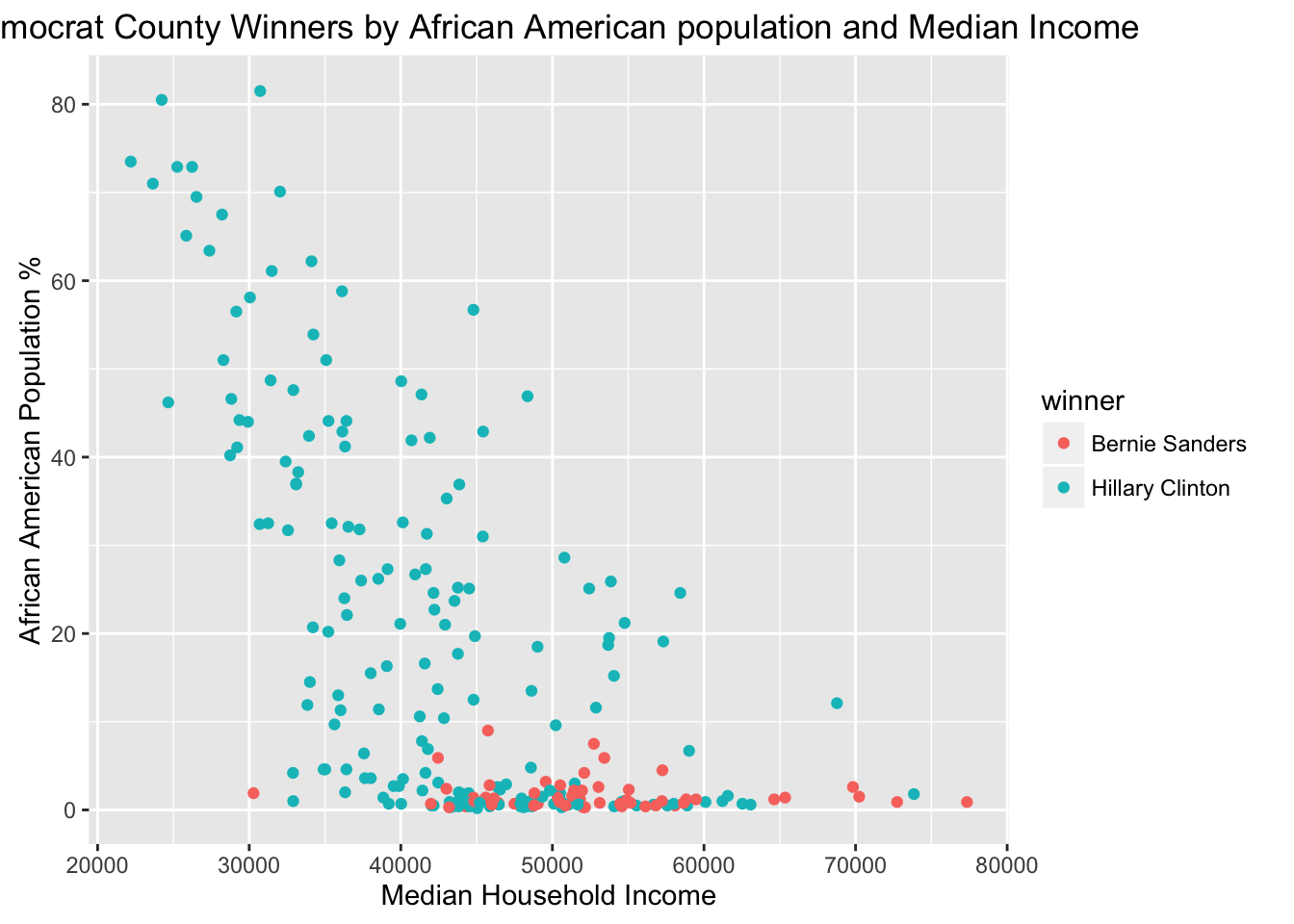


ggplot(data=countyWinnerR\_Demographics, aes(x = poverty , y = highschool,color = winner)) + geom\_point()+ ggtitle("Republican County Winners by Poverty level and Educational Attainment") + xlab("People Below Poverty Level %") + ylab("People With High School Education or More %")

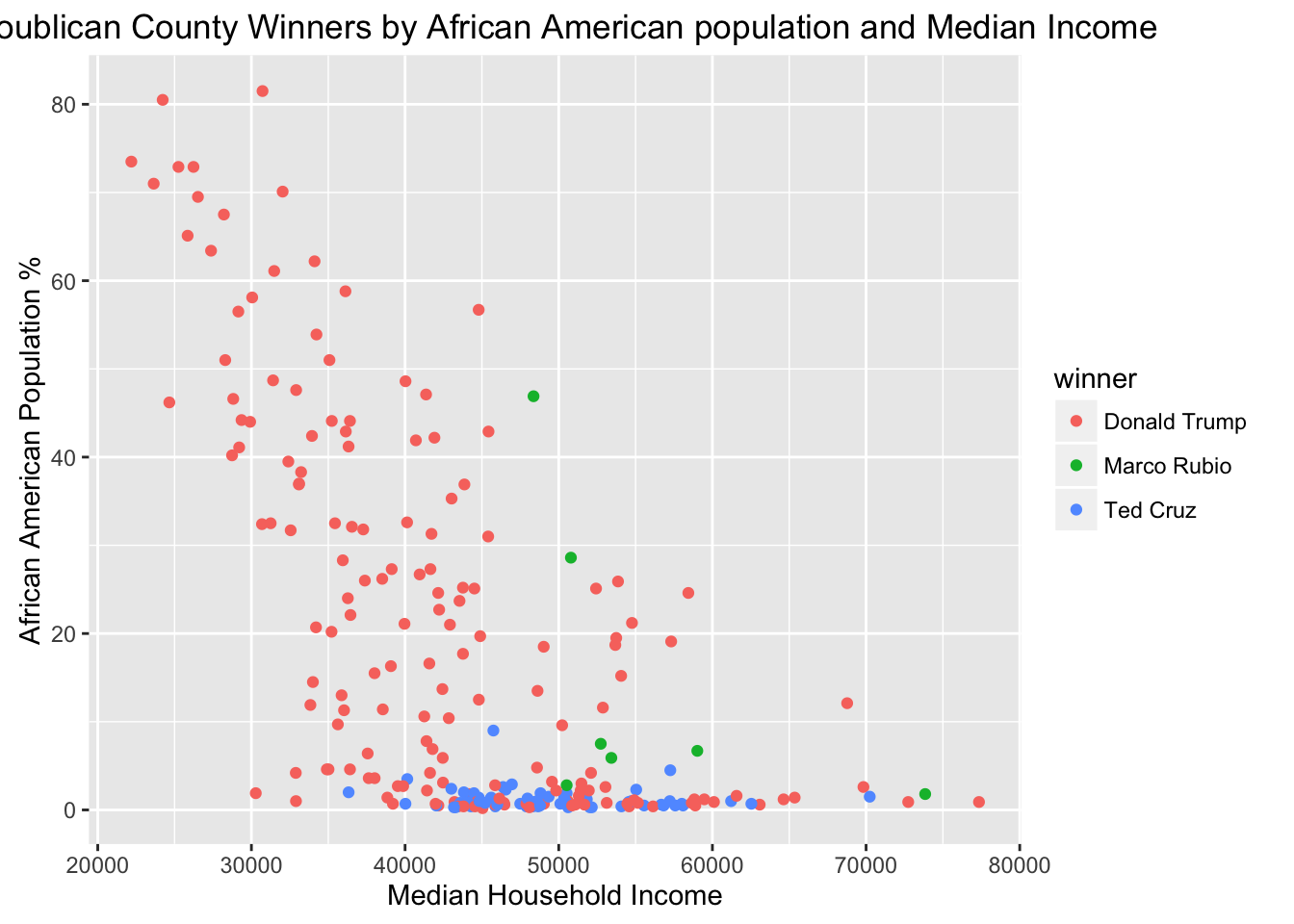


Effect of Median Househod Income and Race

ggplot(data=countyWinnerD\_Demographics, aes(x = median\_income , y = black,color = winner)) + geom\_point()+ ggtitle("Democrat County Winners by African American population and Median Income") + xlab("Median Household Income") + ylab("African American Population %")

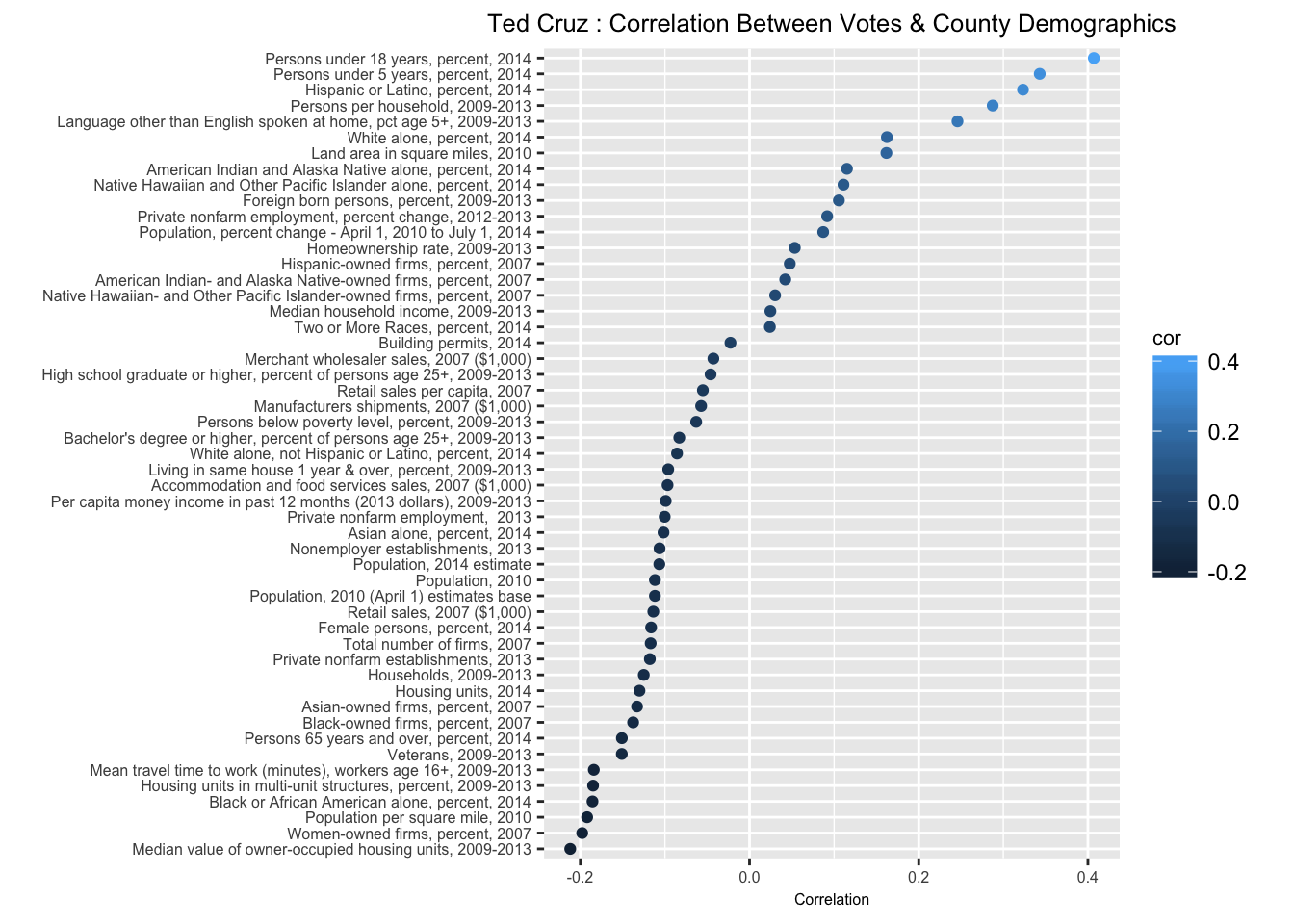
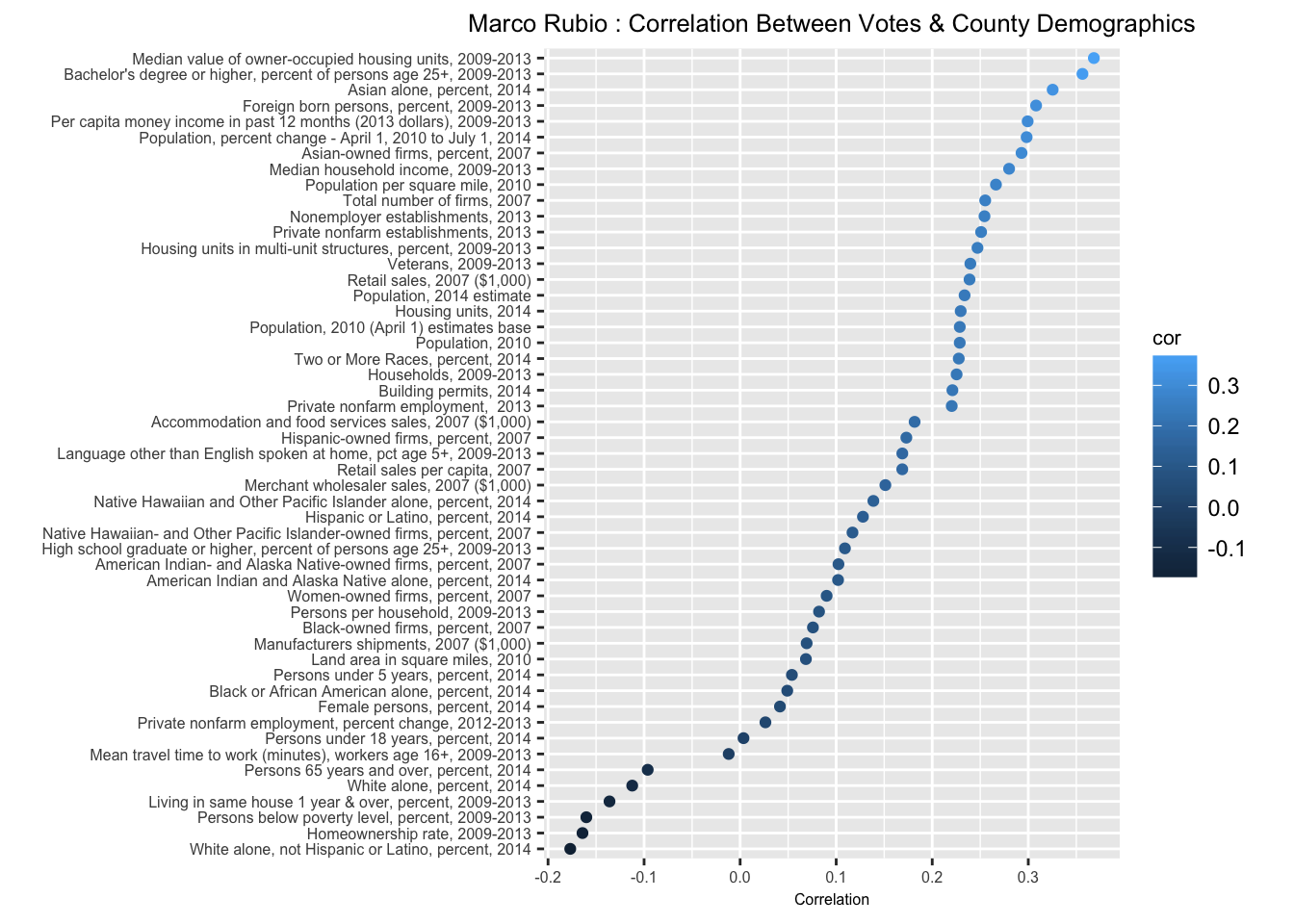
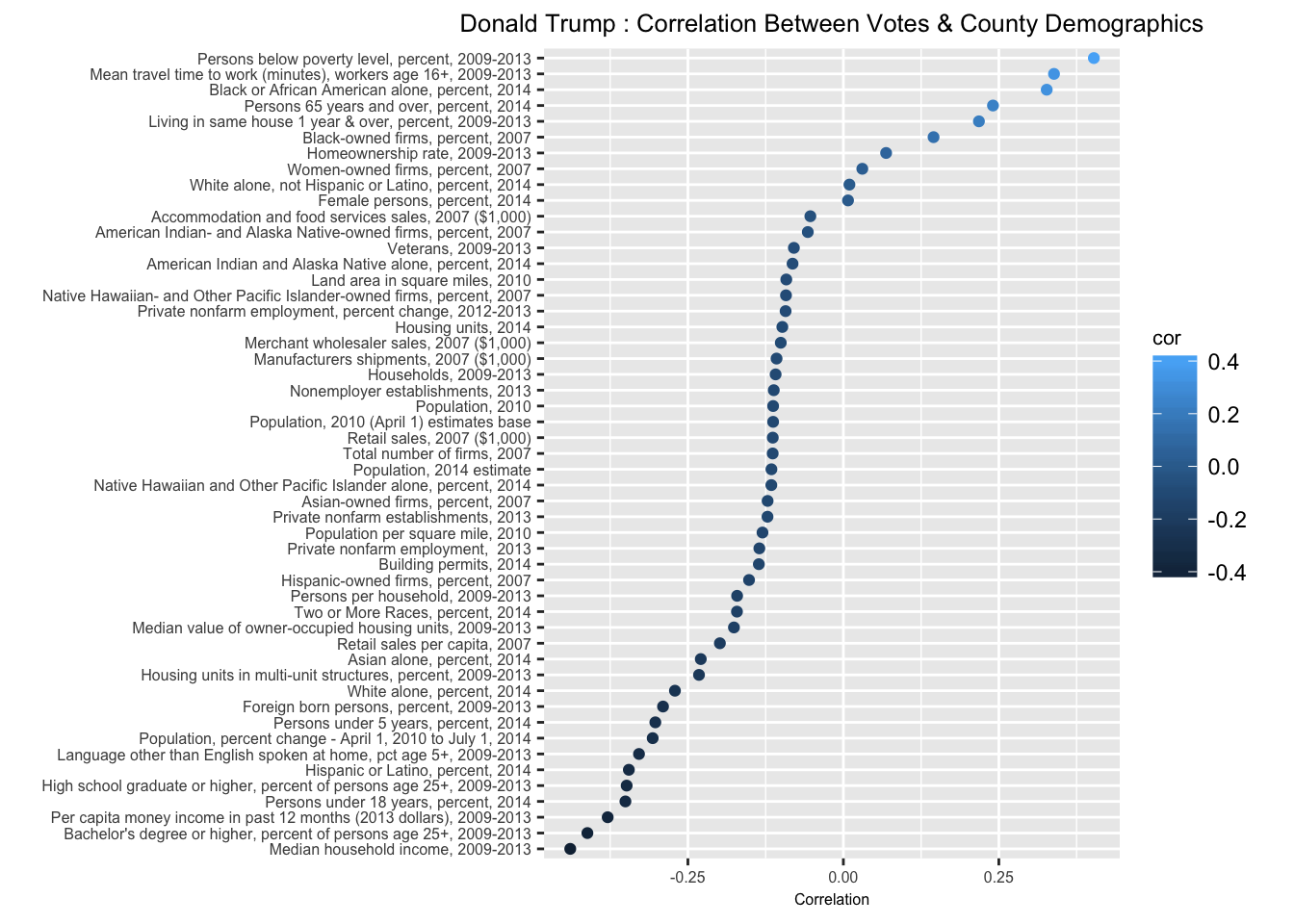
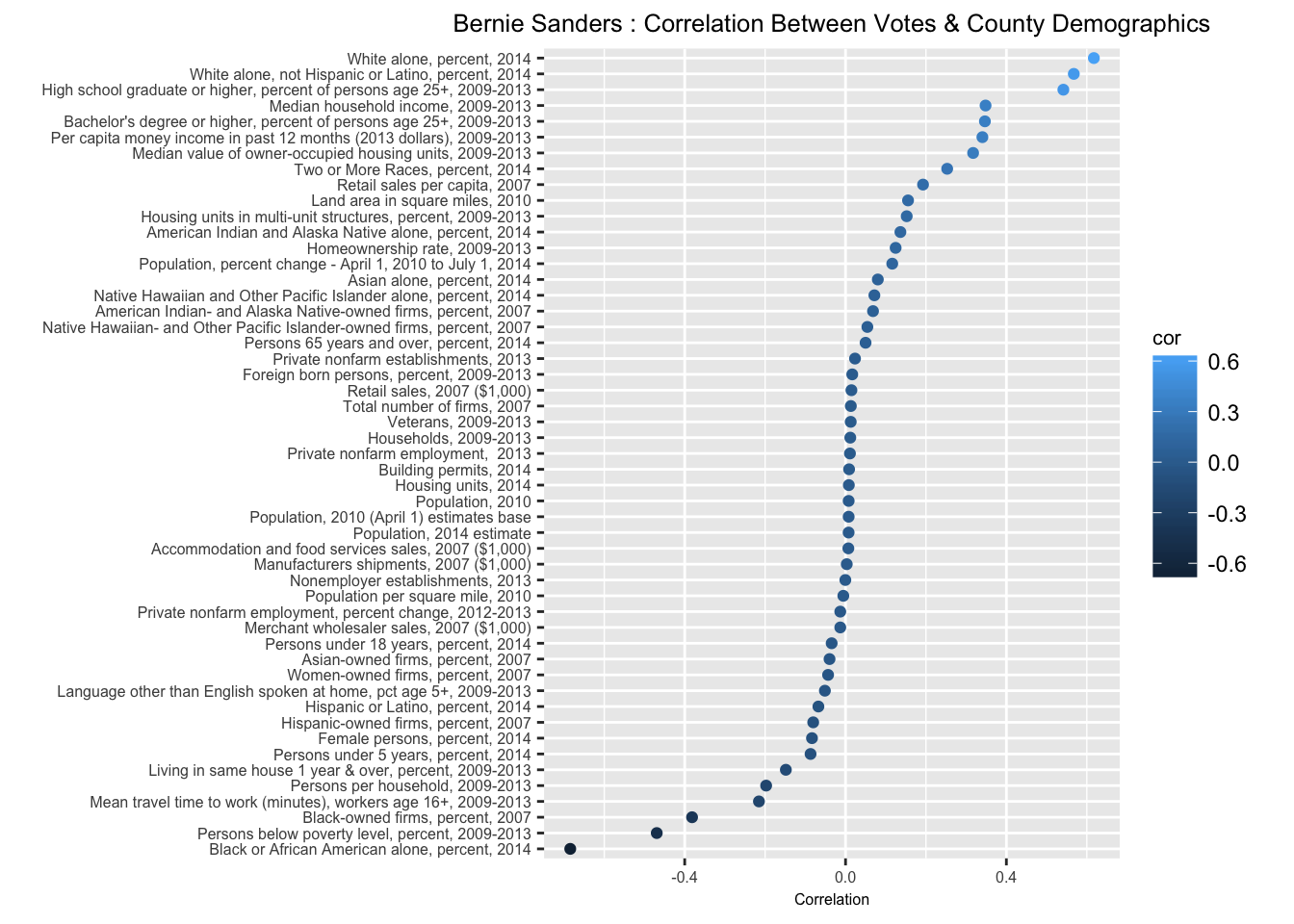
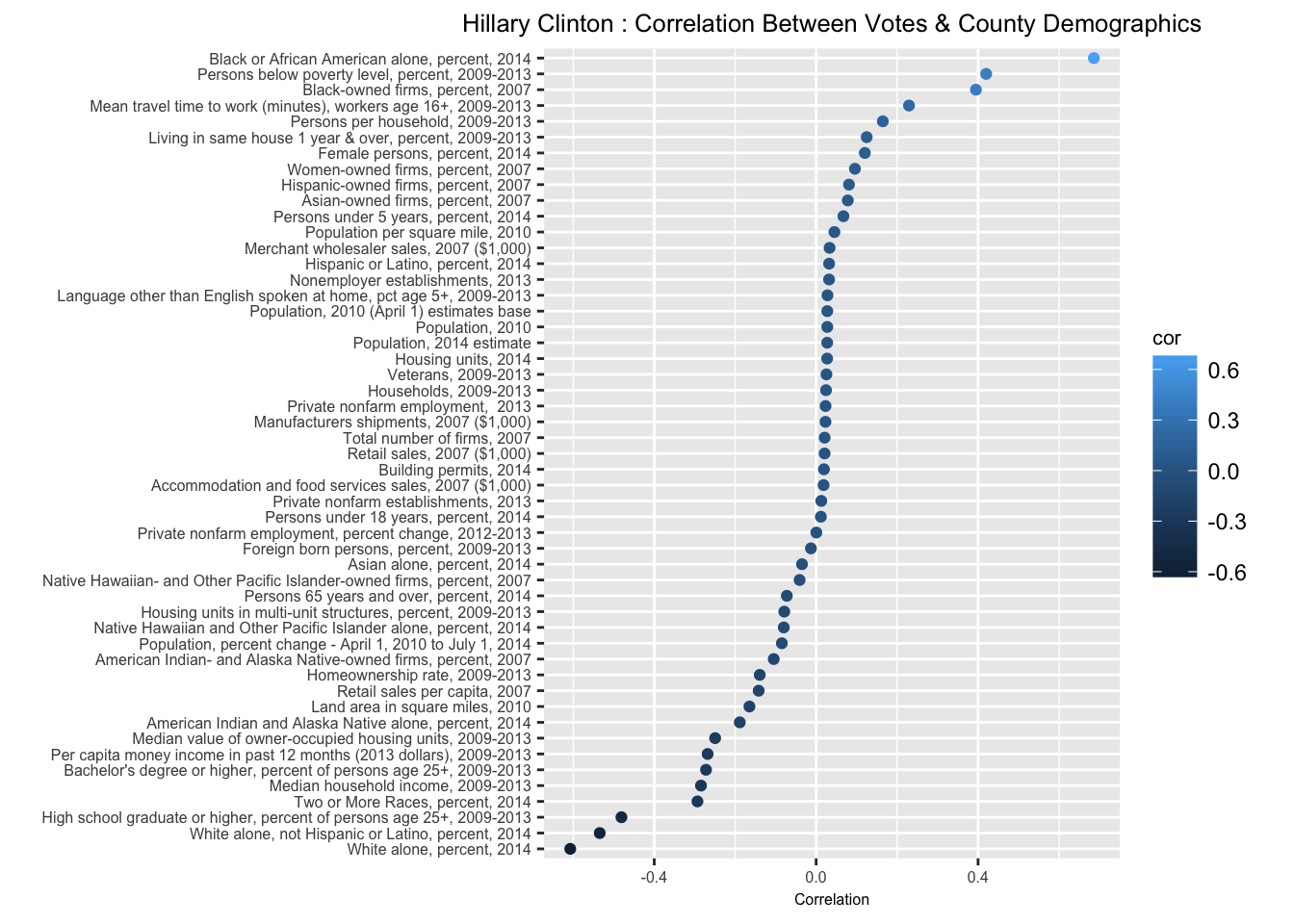


ggplot(data=countyWinnerR\_Demographics, aes(x = median\_income , y = black,color = winner)) + geom\_point()+ ggtitle("Reoublican County Winners by African American population and Median Income") + xlab("Median Household Income") + ylab("African American Population %")



Corelations between demographics and number of votes for Individual Candidates

data <- merge(results,county.demo, by="fips")  
  
candidateList <- c("Hillary Clinton","Bernie Sanders","Donald Trump","Marco Rubio","Ted Cruz")  
   
for(candidate in candidateList)   
{  
 candidate\_cor <- data.frame("var"=dictionary$description, "name"=dictionary$column\_name)  
 candidate\_cor$cor <- c(cor(data[data$candidate==candidate,]$fraction\_votes,  
 data[data$candidate==candidate, dictionary$column\_name]))  
 candidate\_cor <- candidate\_cor[order(candidate\_cor$cor),]  
 candidate\_cor$var <- factor(candidate\_cor$var, levels=candidate\_cor$var)  
 p <- ggplot(candidate\_cor, aes(x=var, y=cor, color= cor)) +   
 geom\_point(stat="identity") +  
 coord\_flip() +   
 ylab("Correlation") + xlab("") +  
 theme(axis.text = element\_text(size=6),  
 axis.title = element\_text(size=6),  
 title = element\_text(size=8)) +  
 ggtitle(paste0(candidate, " : Correlation Between Votes & County Demographics"))  
 print(p)  
}

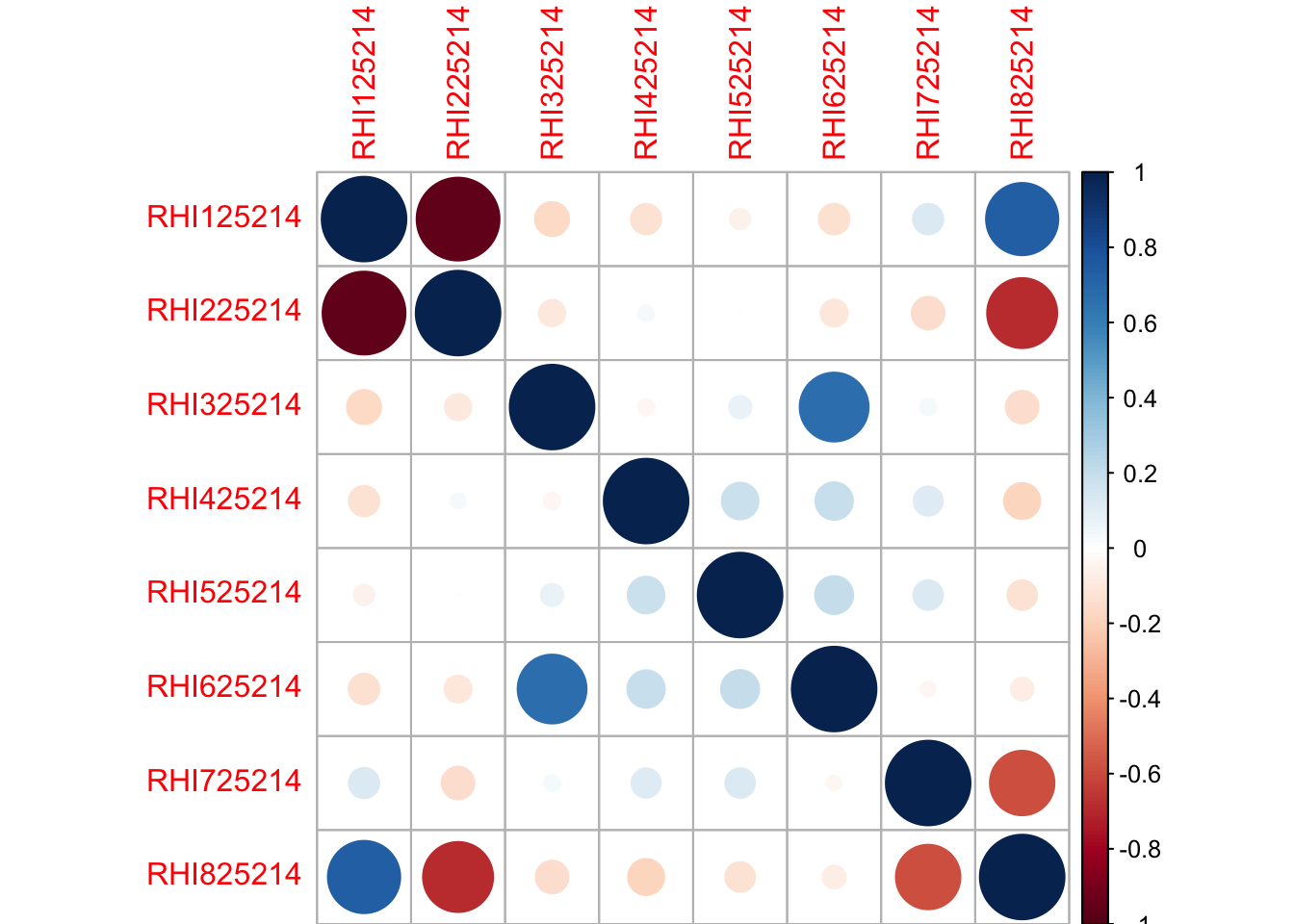


Correlation between Demographics

merged<-read.csv("merged.csv")  
  
Republican.votes<-merged %>%  
 filter(party=="Republican")  
  
Democrats.votes<-merged %>%  
 filter(party=="Democrat")  
  
# Finding correlation between Race variables  
#--------------------------------------------------------------------------  
library(corrplot)

## Warning: package 'corrplot' was built under R version 3.2.5

library(caret)  
raceR<-Republican.votes[, c(20:27)]  
raceR.corr<-cor(raceR)  
corrplot(raceR.corr)

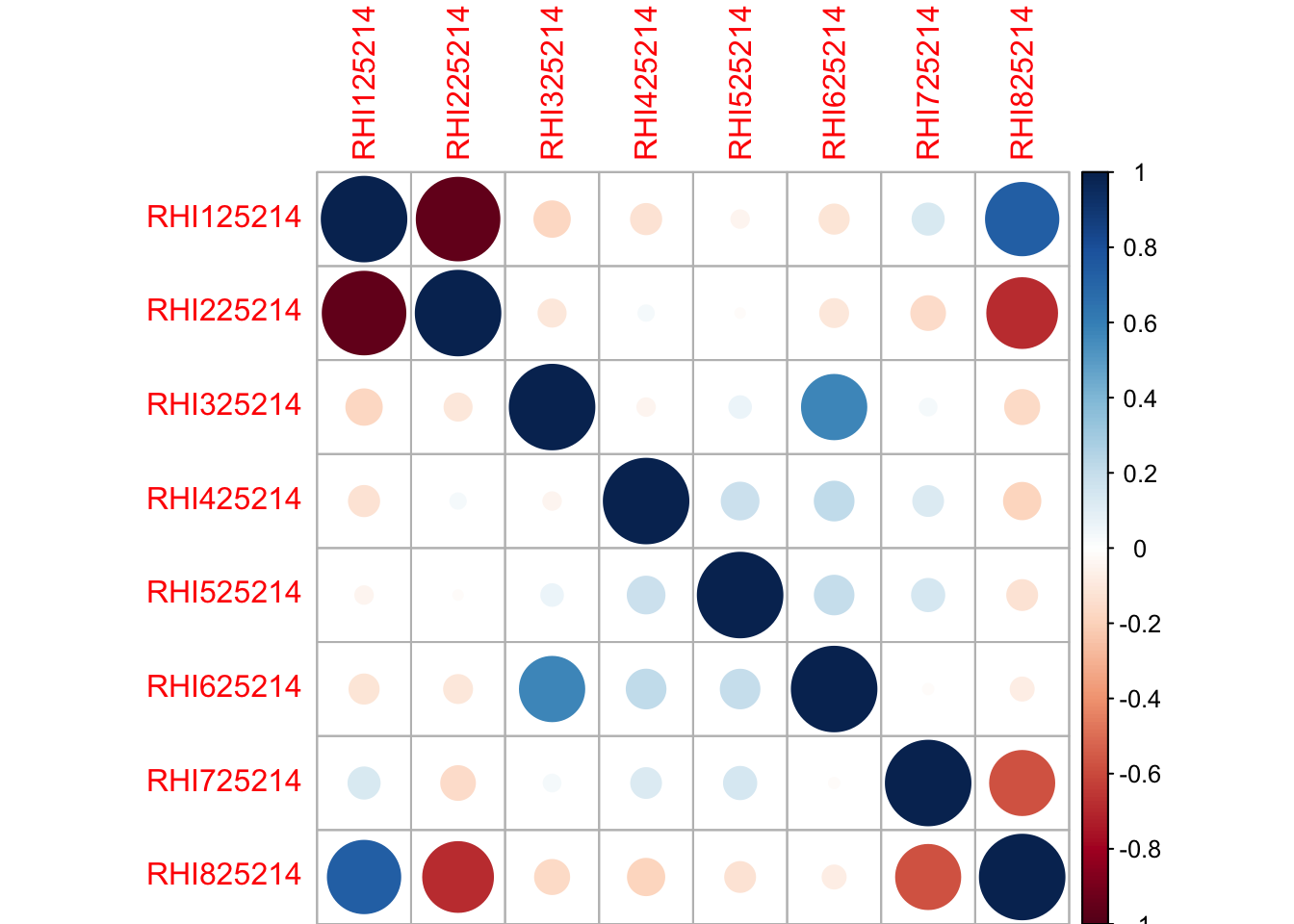


findCorrelation(raceR.corr, cutoff = .50, verbose = TRUE, names = TRUE, exact = TRUE)

## Compare row 8 and column 1 with corr 0.73   
## Means: 0.361 vs 0.204 so flagging column 8   
## Compare row 1 and column 2 with corr 0.959   
## Means: 0.262 vs 0.164 so flagging column 1   
## Compare row 6 and column 3 with corr 0.664   
## Means: 0.239 vs 0.128 so flagging column 6   
## All correlations <= 0.5

## [1] "RHI825214" "RHI125214" "RHI625214"

raceD<-Democrats.votes[, c(20:27)]  
raceD.corr<-cor(raceD)  
corrplot(raceD.corr)

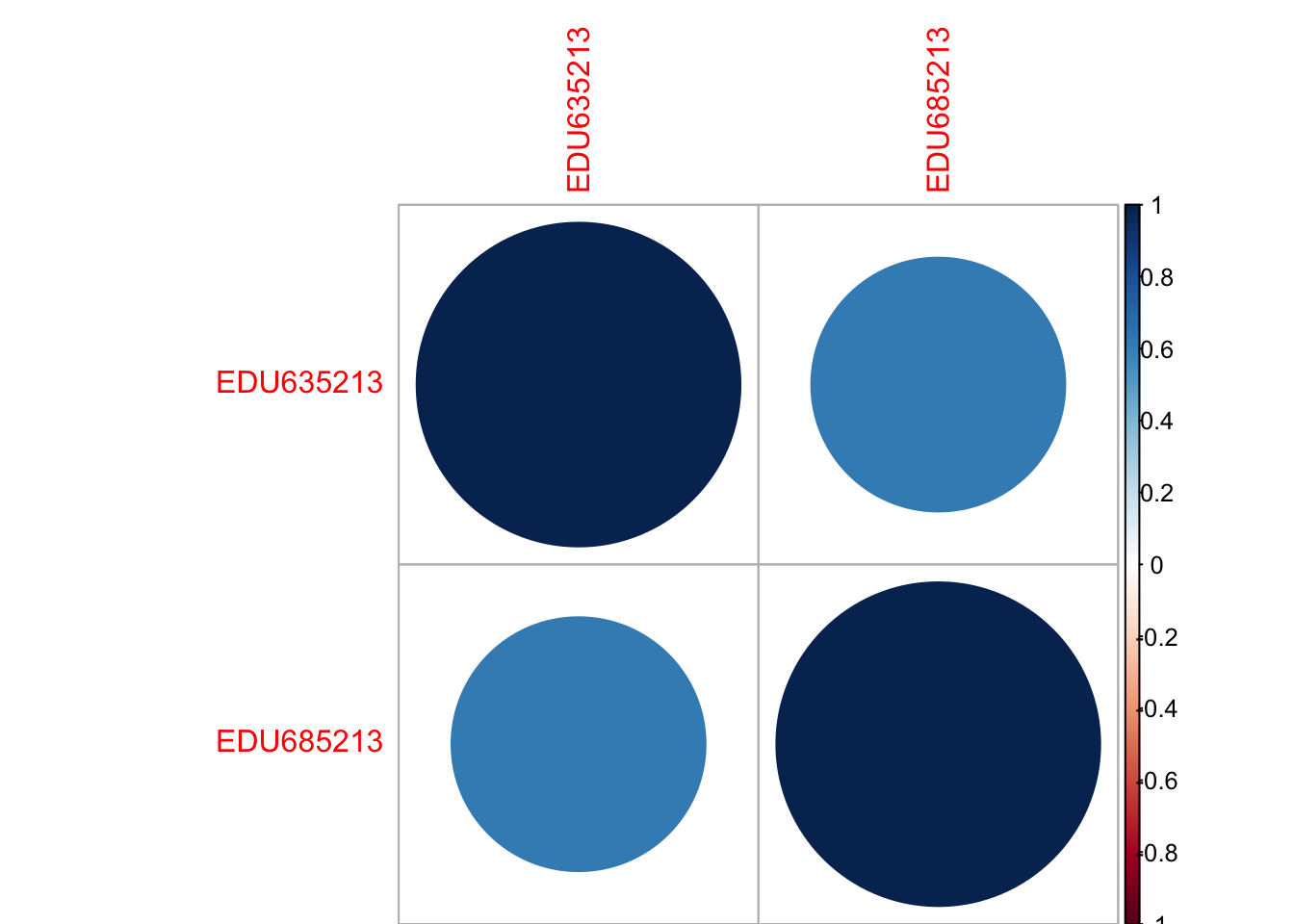


findCorrelation(raceD.corr, cutoff = .50, verbose = TRUE, names = TRUE, exact = TRUE)

## Compare row 8 and column 1 with corr 0.732   
## Means: 0.362 vs 0.203 so flagging column 8   
## Compare row 1 and column 2 with corr 0.954   
## Means: 0.259 vs 0.161 so flagging column 1   
## Compare row 6 and column 3 with corr 0.579   
## Means: 0.224 vs 0.13 so flagging column 6   
## All correlations <= 0.5

## [1] "RHI825214" "RHI125214" "RHI625214"

#----------------------------------------------------------------------------  
# Finding correlation between Education Variables  
educationR<-Republican.votes[, c(31:32)]  
educationR.corr<-cor(educationR)  
corrplot(educationR.corr)

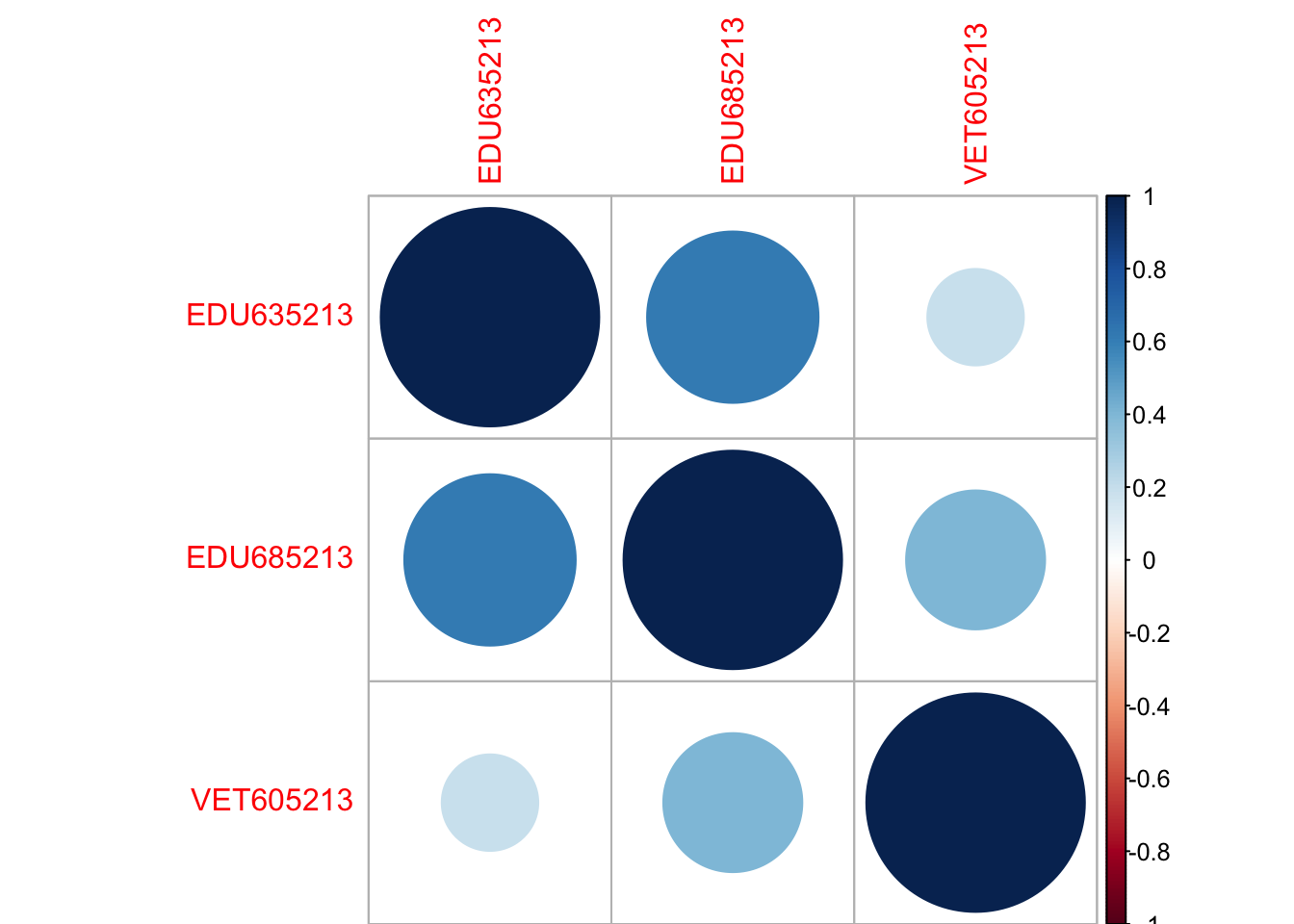


findCorrelation(educationR.corr, cutoff = .50, verbose = TRUE, names = TRUE, exact = TRUE)

## Compare row 1 and column 2 with corr 0.615   
## Means: 0.615 vs 0.615 so flagging column 2

## [1] "EDU685213"

#----------------------------------------------------------------------------  
# Finding correlation between Education Variables and Veteran variable  
educationVR<-Republican.votes[, c(31:33)]  
educationVR.corr<-cor(educationVR)  
corrplot(educationVR.corr)



findCorrelation(educationVR.corr, cutoff = .60, verbose = TRUE, names = TRUE, exact = TRUE)

## Compare row 2 and column 1 with corr 0.615   
## Means: 0.51 vs 0.405 so flagging column 2   
## All correlations <= 0.6

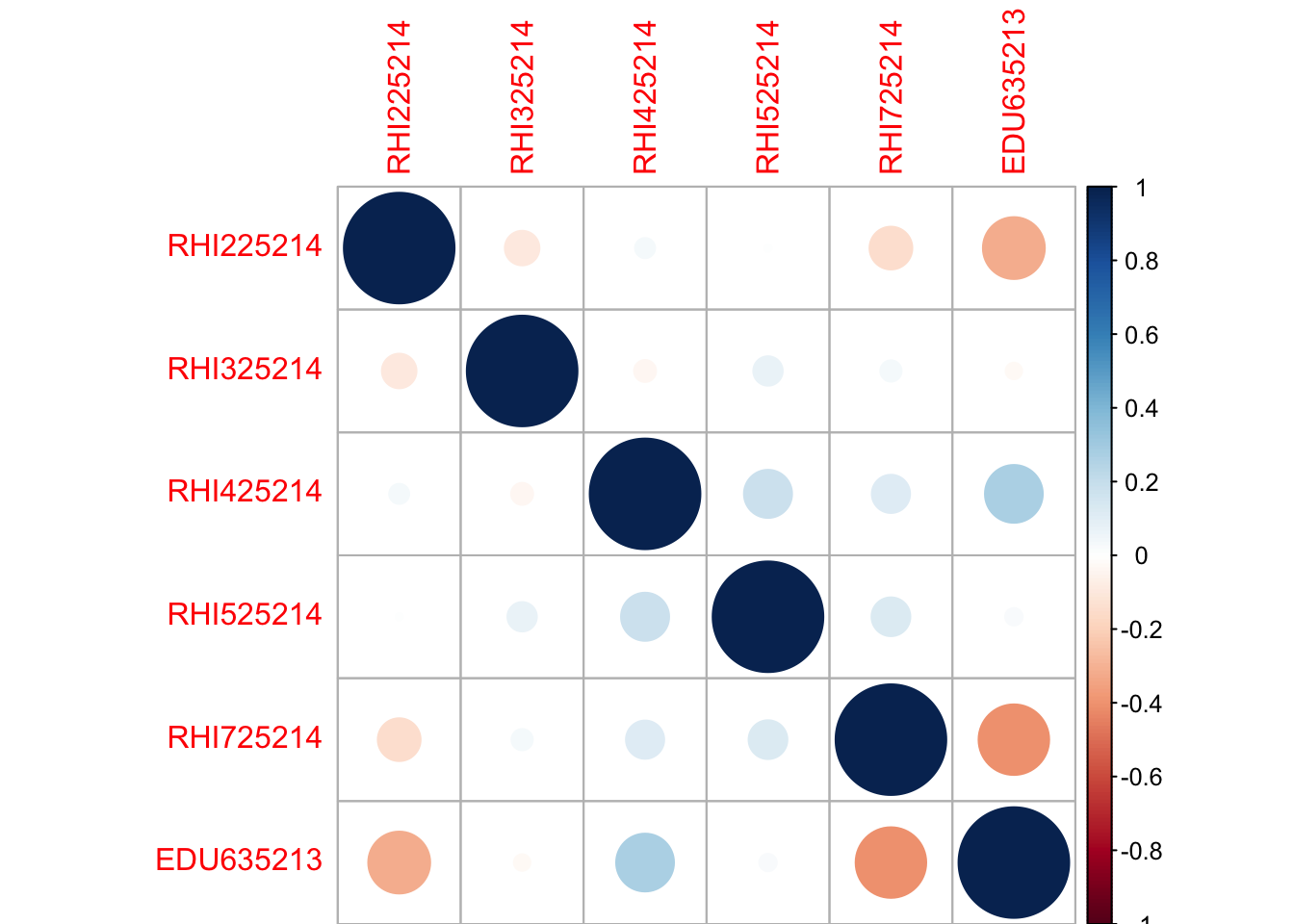
## [1] "EDU685213"

#----------------------------------------------------------------------------------------  
findCorrelation(educationR.corr, cutoff = .50, verbose = TRUE, names = TRUE, exact = TRUE)

## Compare row 1 and column 2 with corr 0.615   
## Means: 0.615 vs 0.615 so flagging column 2

## [1] "EDU685213"

#Finding correlation between Race and Education variables  
eduraceR<-Republican.votes[,c(21,22,23,24,26,31)]  
eduraceR.corr<-cor(eduraceR)  
corrplot(eduraceR.corr)

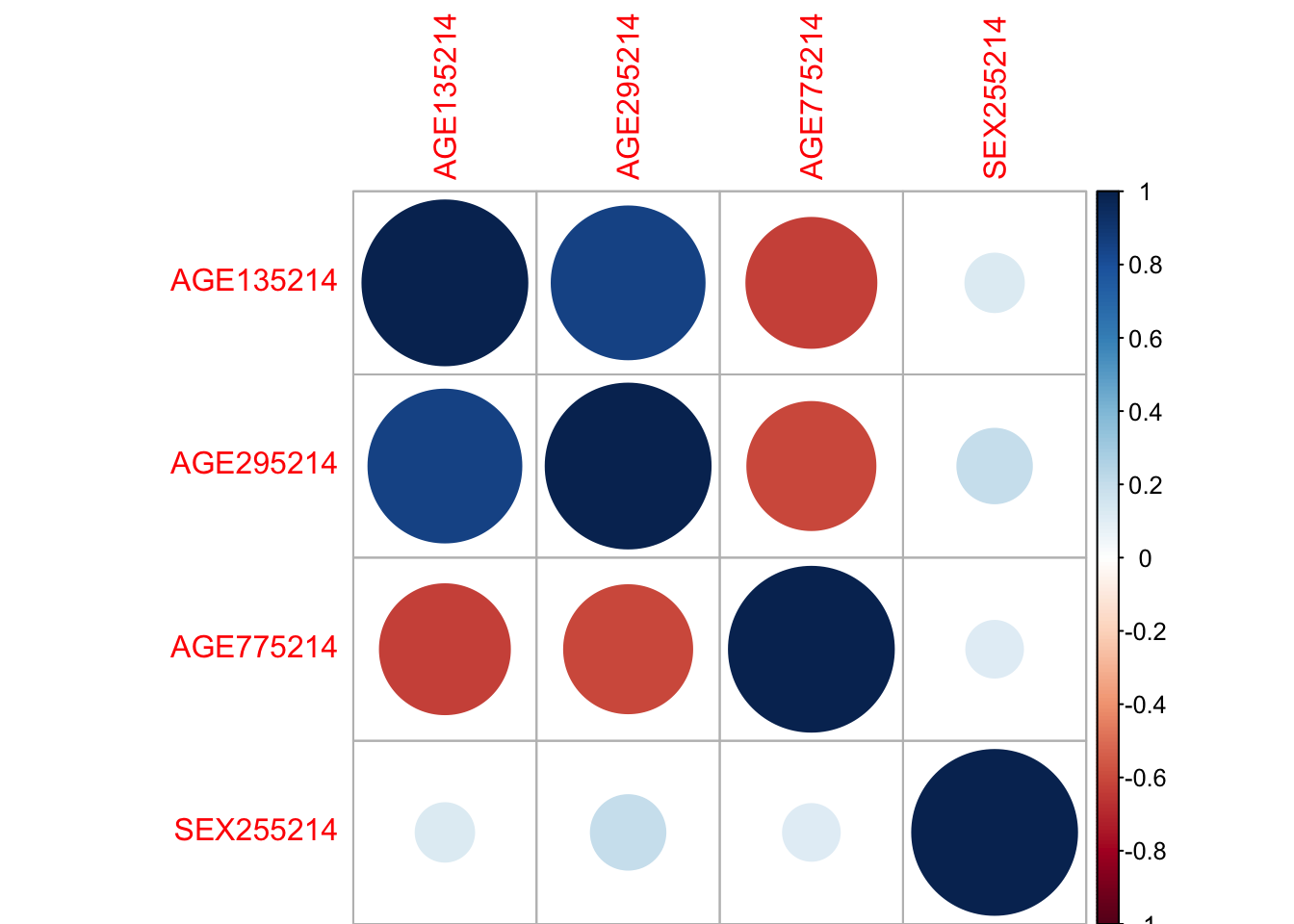


findCorrelation(eduraceR.corr, cutoff = .60, verbose = TRUE, names = TRUE, exact = TRUE)

## All correlations <= 0.6

## character(0)

#---------------------------------------------------------------------------  
# Finding correlation between age and sex variables  
agesexR<-Republican.votes[, c(16:19)]  
agesexR.corr<-cor(agesexR)  
corrplot(agesexR.corr)

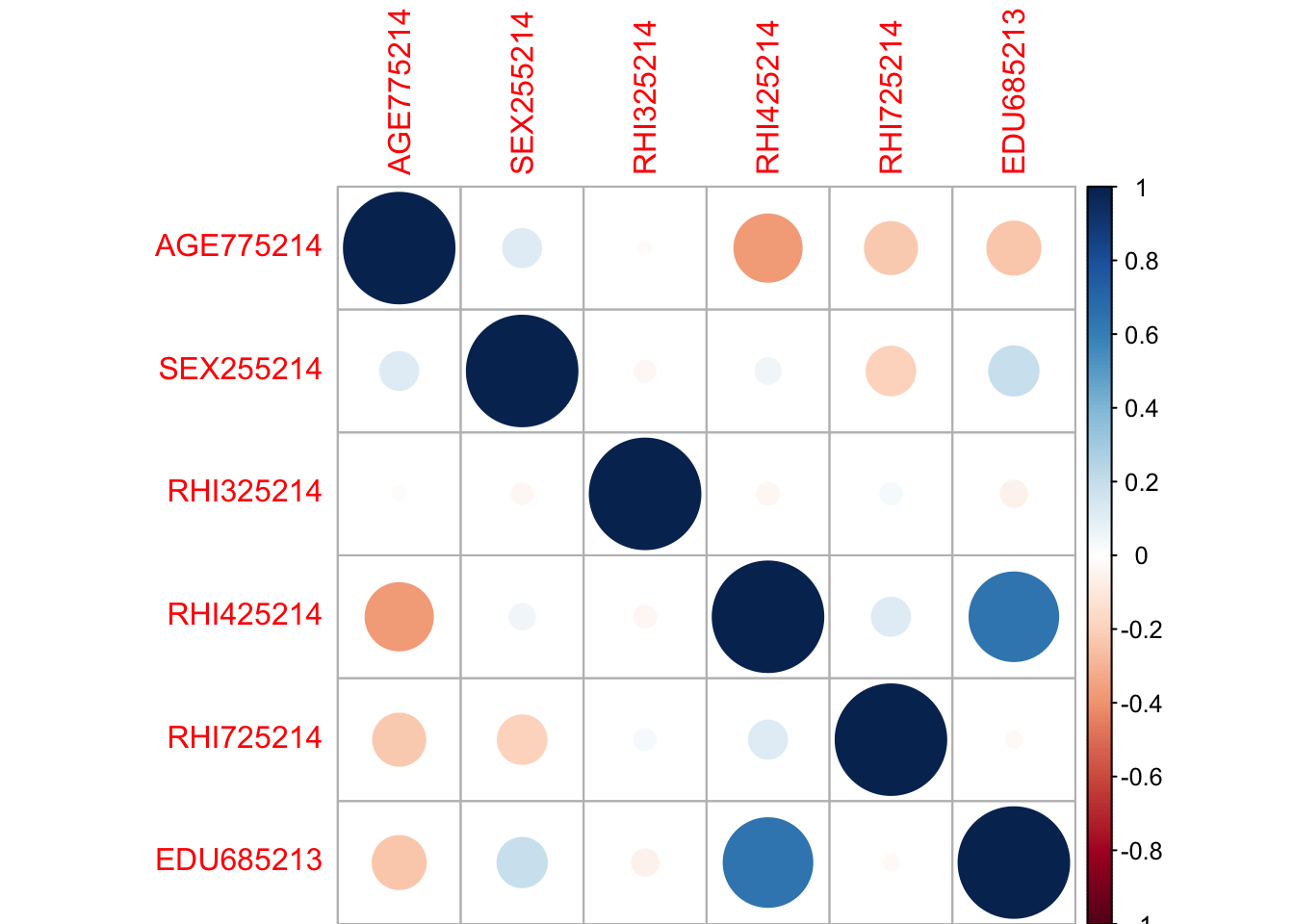


findCorrelation(agesexR.corr, cutoff = .60, verbose = TRUE, names = TRUE, exact = TRUE)

## Compare row 2 and column 1 with corr 0.86   
## Means: 0.556 vs 0.384 so flagging column 2   
## Compare row 1 and column 3 with corr 0.621   
## Means: 0.374 vs 0.248 so flagging column 1   
## All correlations <= 0.6

## [1] "AGE295214" "AGE135214"

#-----------------------------------------------------------------------------  
#Finding correlation between Age, Sex(Females only), Education and Race Variables  
ASER<-Republican.votes[, c(18,19,22,23,26,32)]  
ASER.corr<-cor(ASER)  
corrplot(ASER.corr)

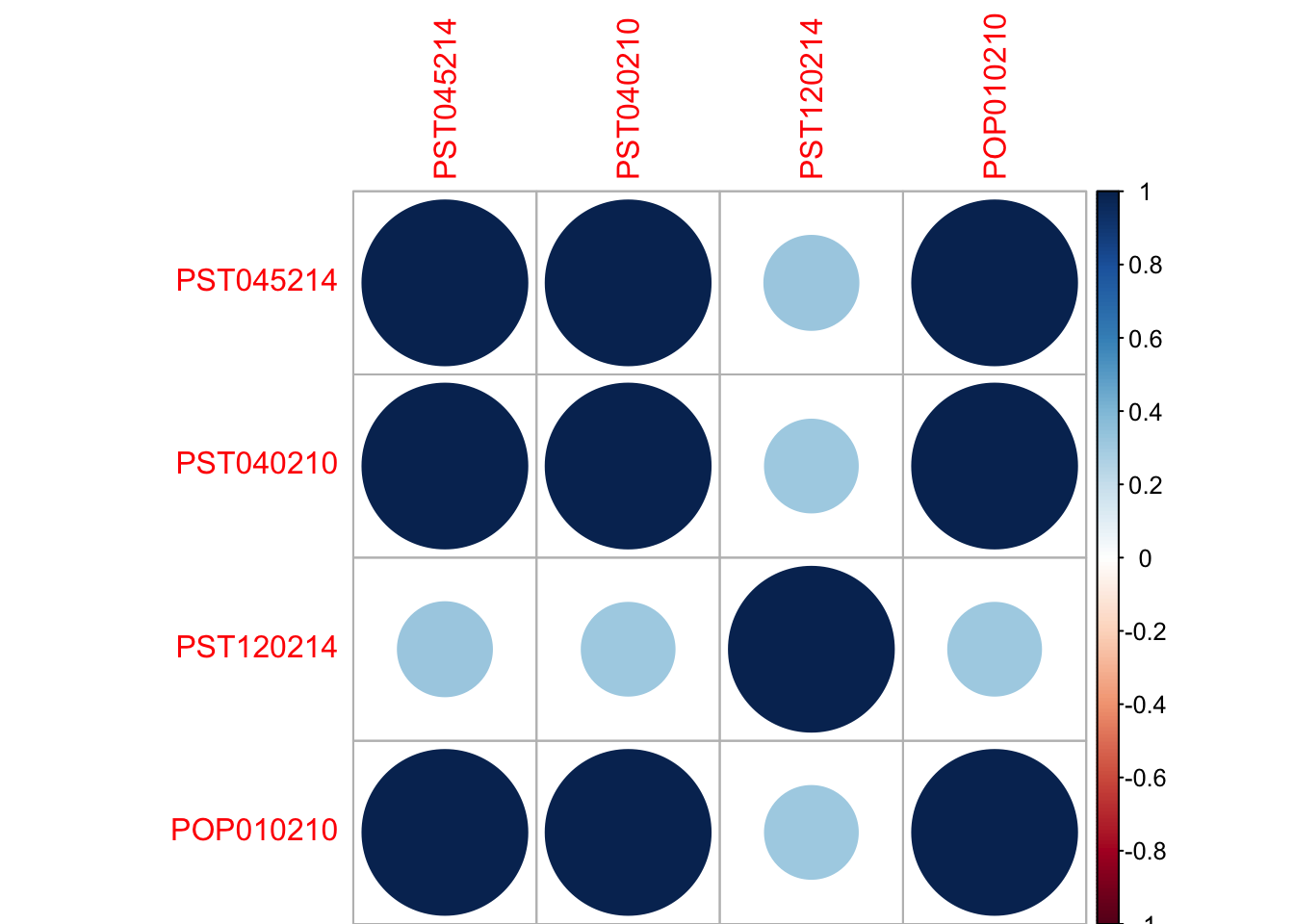


findCorrelation(ASER.corr, cutoff = .60, verbose = TRUE, names = TRUE, exact = TRUE)

## Compare row 4 and column 6 with corr 0.643   
## Means: 0.245 vs 0.142 so flagging column 4   
## All correlations <= 0.6

## [1] "RHI425214"

#------------------------------------------------------------------------------  
#Finding correlation between population variables  
popR<-Republican.votes[, c(12:15)]  
pop.corr<-cor(popR)  
corrplot(pop.corr)

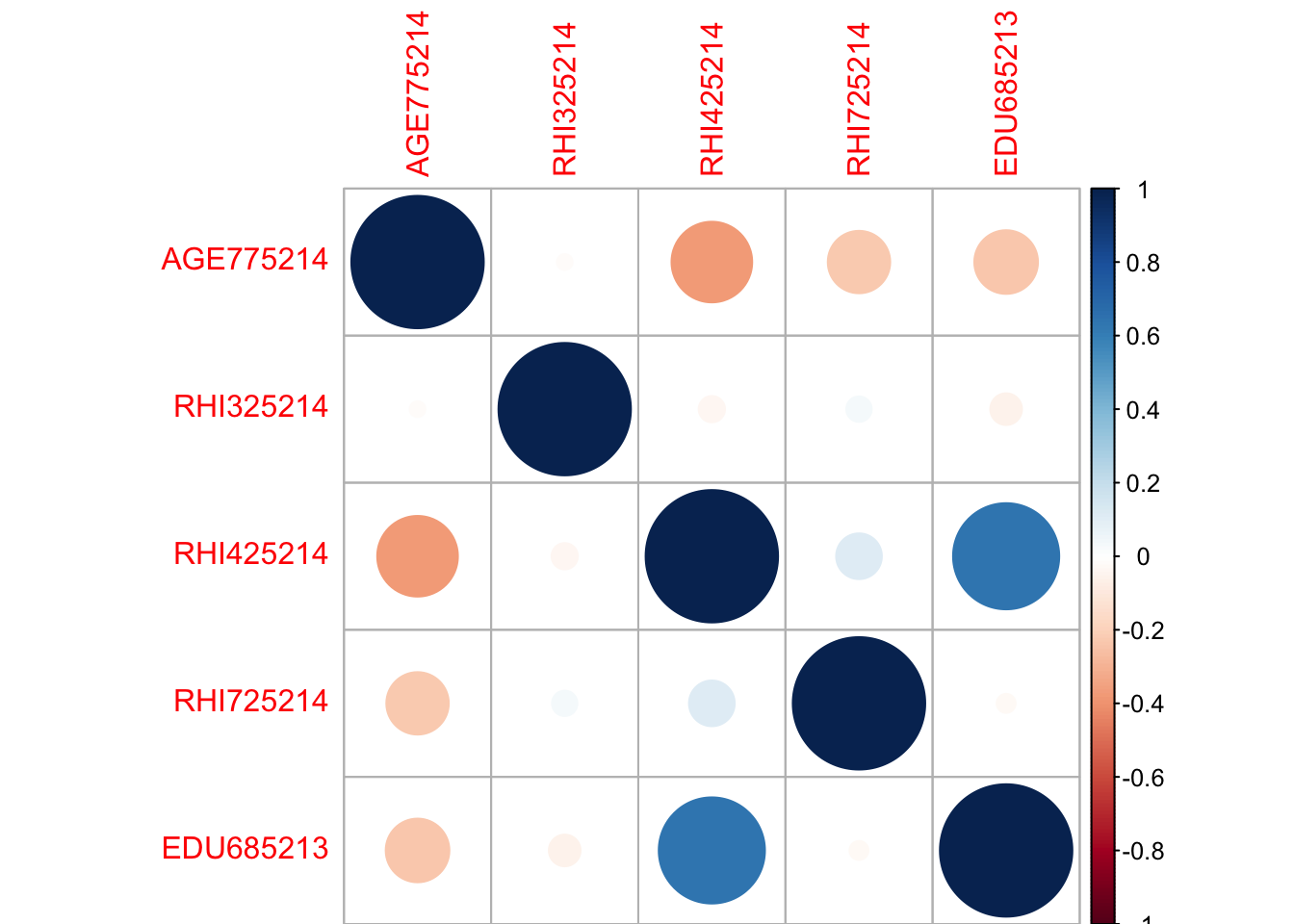


findCorrelation(pop.corr, cutoff = .60, verbose = TRUE, names = TRUE, exact = TRUE)

## Compare row 1 and column 2 with corr 0.999   
## Means: 0.775 vs 0.622 so flagging column 1   
## Compare row 2 and column 4 with corr 1   
## Means: 0.658 vs 0.488 so flagging column 2   
## All correlations <= 0.6

## [1] "PST045214" "PST040210"

#-------------------------------------------------------------------------------  
#Finding correlation between Age, Education and Race Variables  
AER<-Republican.votes[, c(18,22,23,26,32)]  
AER.corr<-cor(AER)  
corrplot(AER.corr)

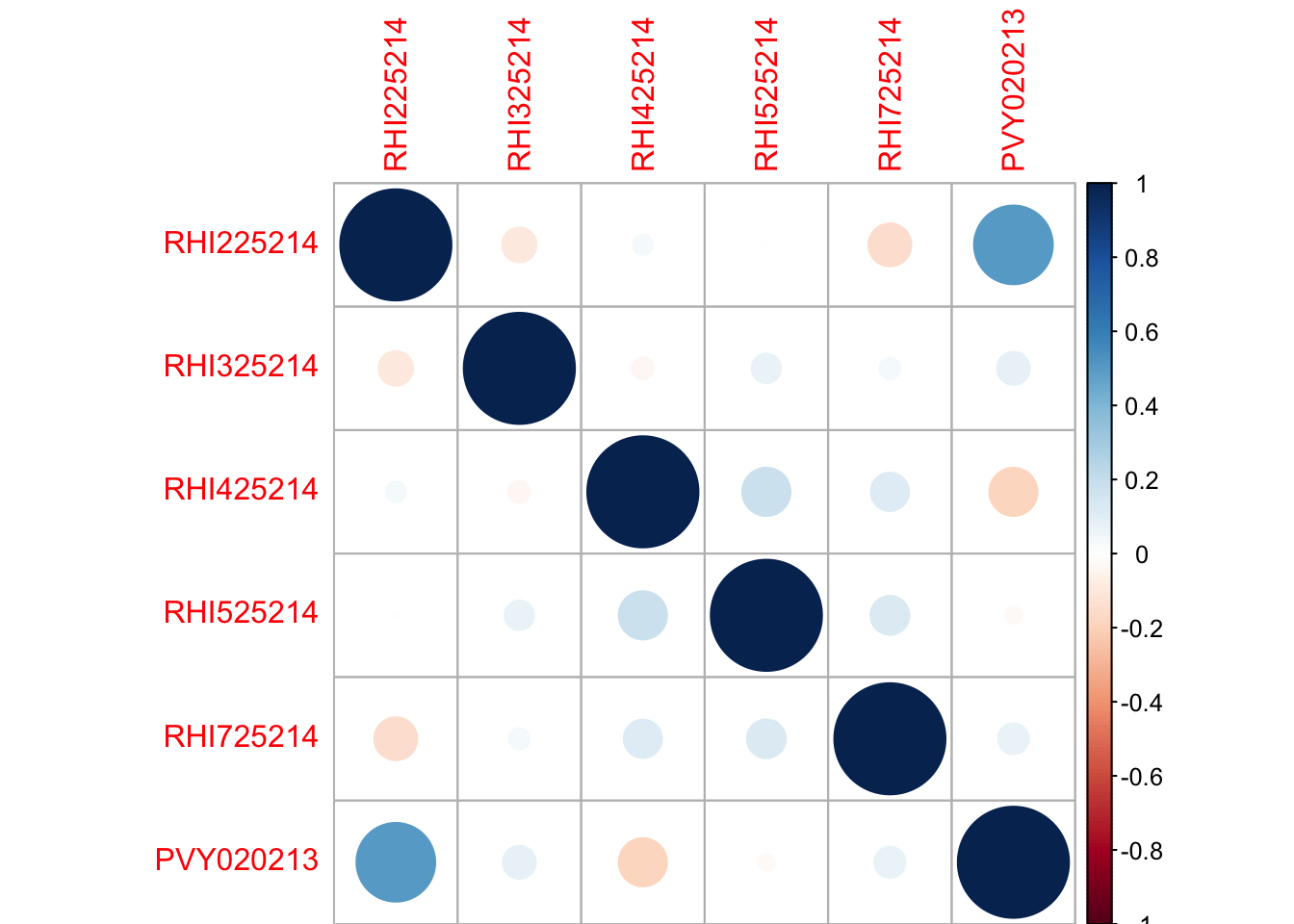


findCorrelation(AER.corr, cutoff = .60, verbose = TRUE, names = TRUE, exact = TRUE)

## Compare row 3 and column 5 with corr 0.643   
## Means: 0.293 vs 0.16 so flagging column 3   
## All correlations <= 0.6

## [1] "RHI425214"

#---------------------------------------------------------------------------------  
#Finding correlation between Race and Poverty level  
  
RP<-Republican.votes[, c(21,22,23,24,26,43)]  
RP.corr<-cor(RP)  
corrplot(RP.corr)

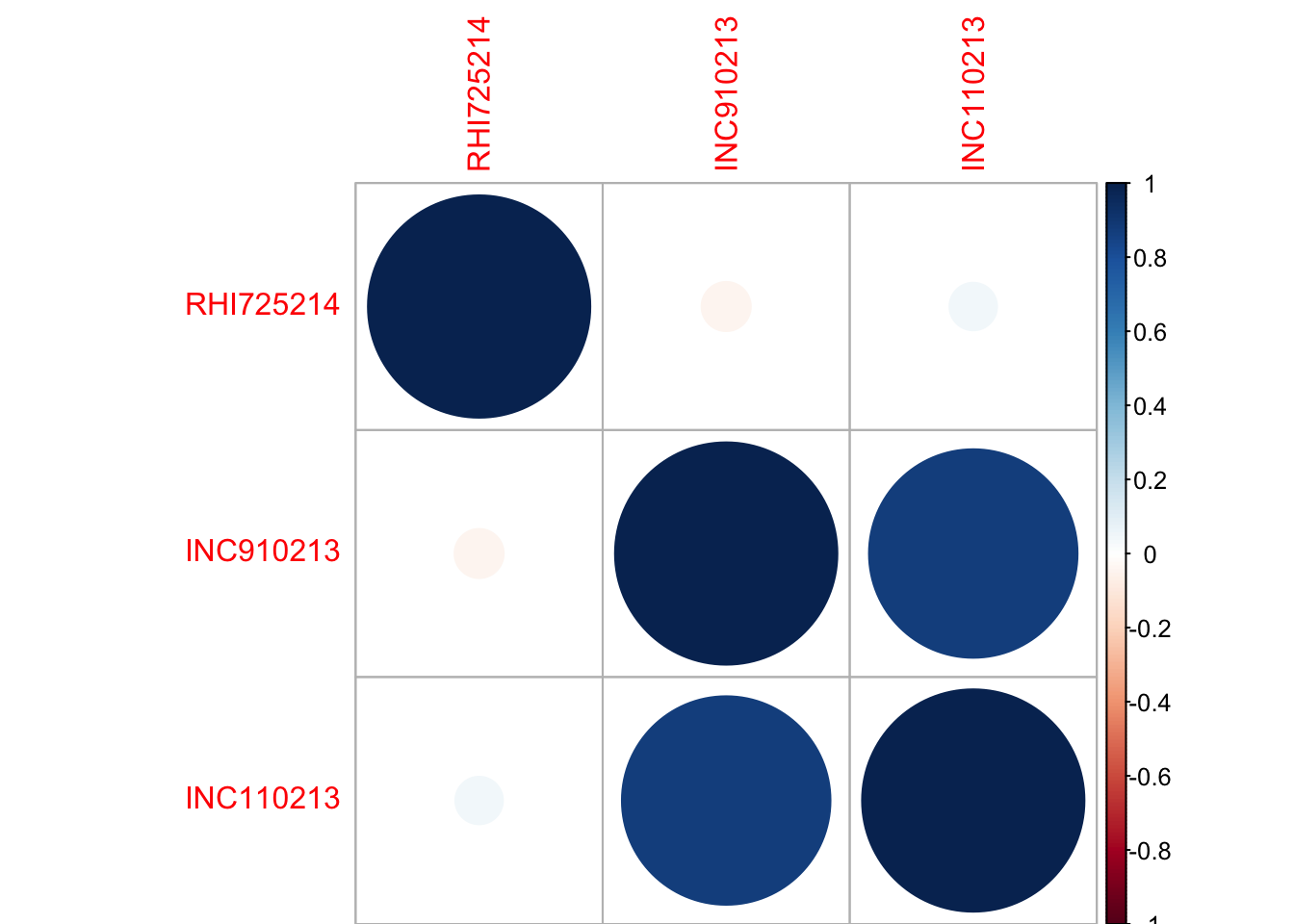


findCorrelation(RP.corr, cutoff = .60, verbose = TRUE, names = TRUE, exact = TRUE)

## All correlations <= 0.6

## character(0)

#----------------------------------------------------------------------------------  
#Finding correlation between Race and Income variables  
eduraceR<-Republican.votes[,c(26,41,42)]  
eduraceR.corr<-cor(eduraceR)  
corrplot(eduraceR.corr)

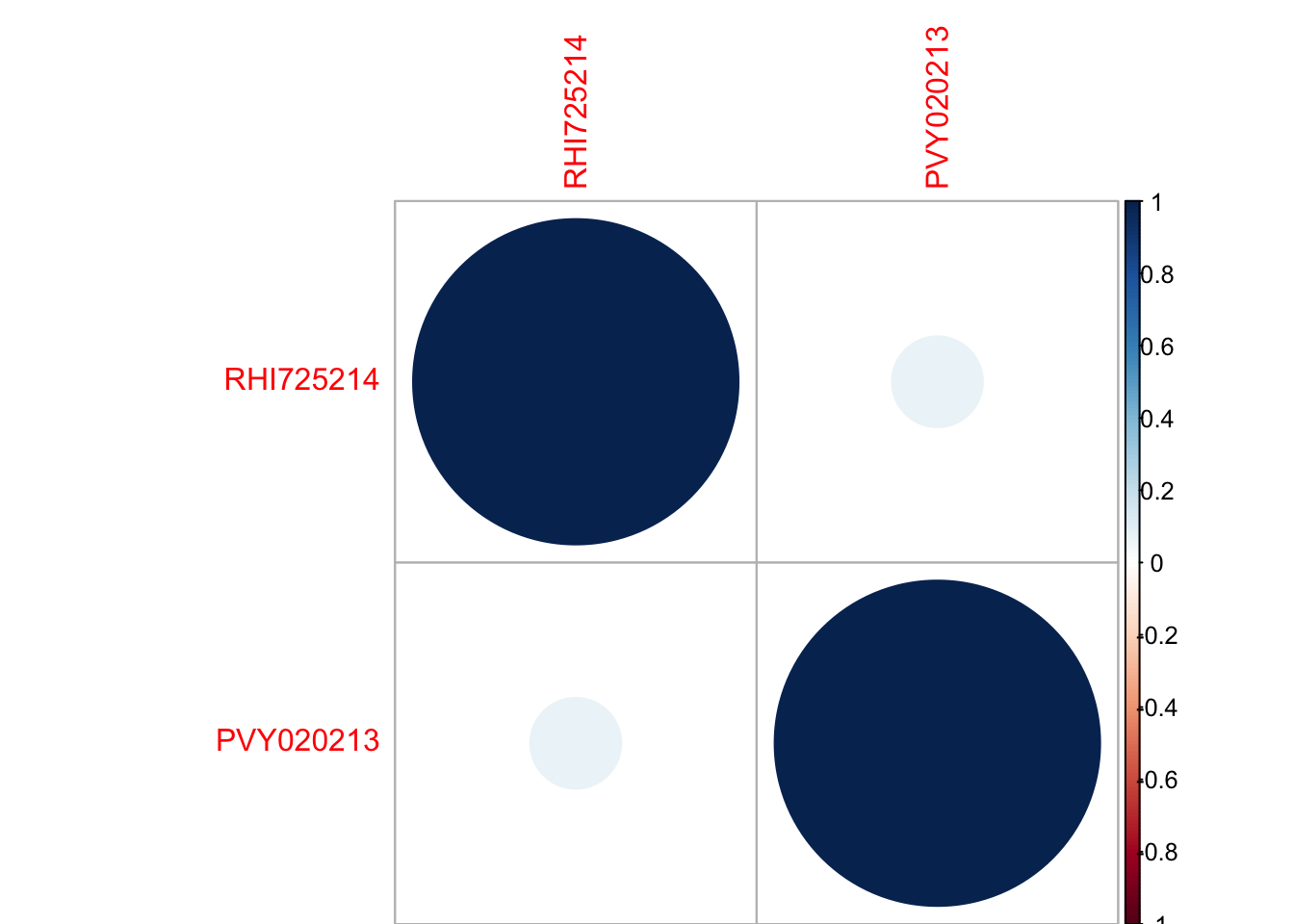


findCorrelation(eduraceR.corr, cutoff = .60, verbose = TRUE, names = TRUE, exact = TRUE)

## Compare row 2 and column 3 with corr 0.879   
## Means: 0.464 vs 0.256 so flagging column 2   
## All correlations <= 0.6

## [1] "INC910213"

#-------------------------------------------------------------------------------------  
#Finding correlation between Hispanic and Poverty variables  
eduraceR<-Republican.votes[,c(26,43)]  
eduraceR.corr<-cor(eduraceR)  
corrplot(eduraceR.corr)

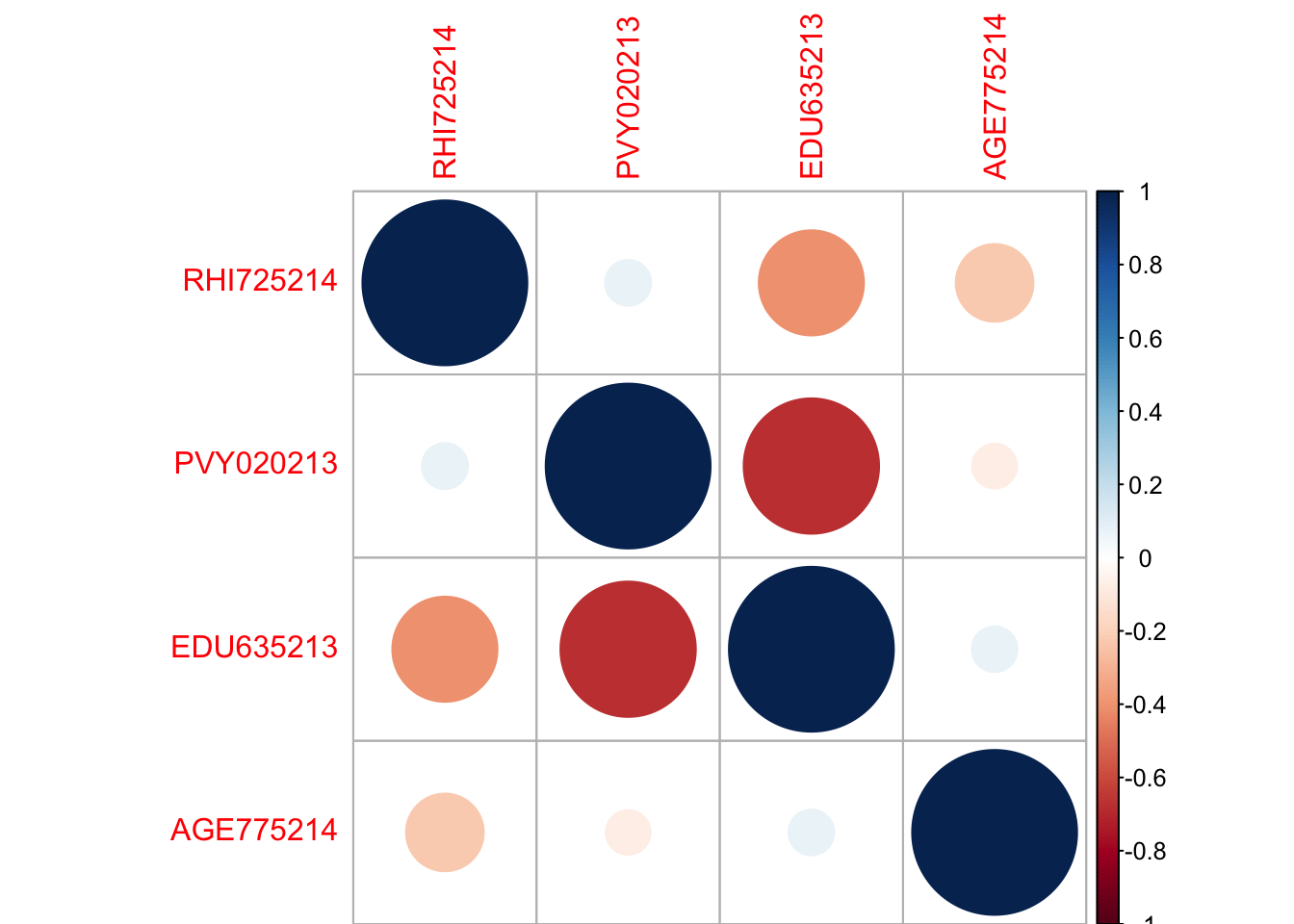


findCorrelation(eduraceR.corr, cutoff = .60, verbose = TRUE, names = TRUE, exact = TRUE)

## All correlations <= 0.6

## character(0)

#----------------------------------------------------------------------------------------  
#Finding correlation between Hispanic and Poverty, and age 65+ variables, education (Highschool)  
eduraceR<-Republican.votes[,c(26,43,31,18)]  
eduraceR.corr<-cor(eduraceR)  
corrplot(eduraceR.corr)

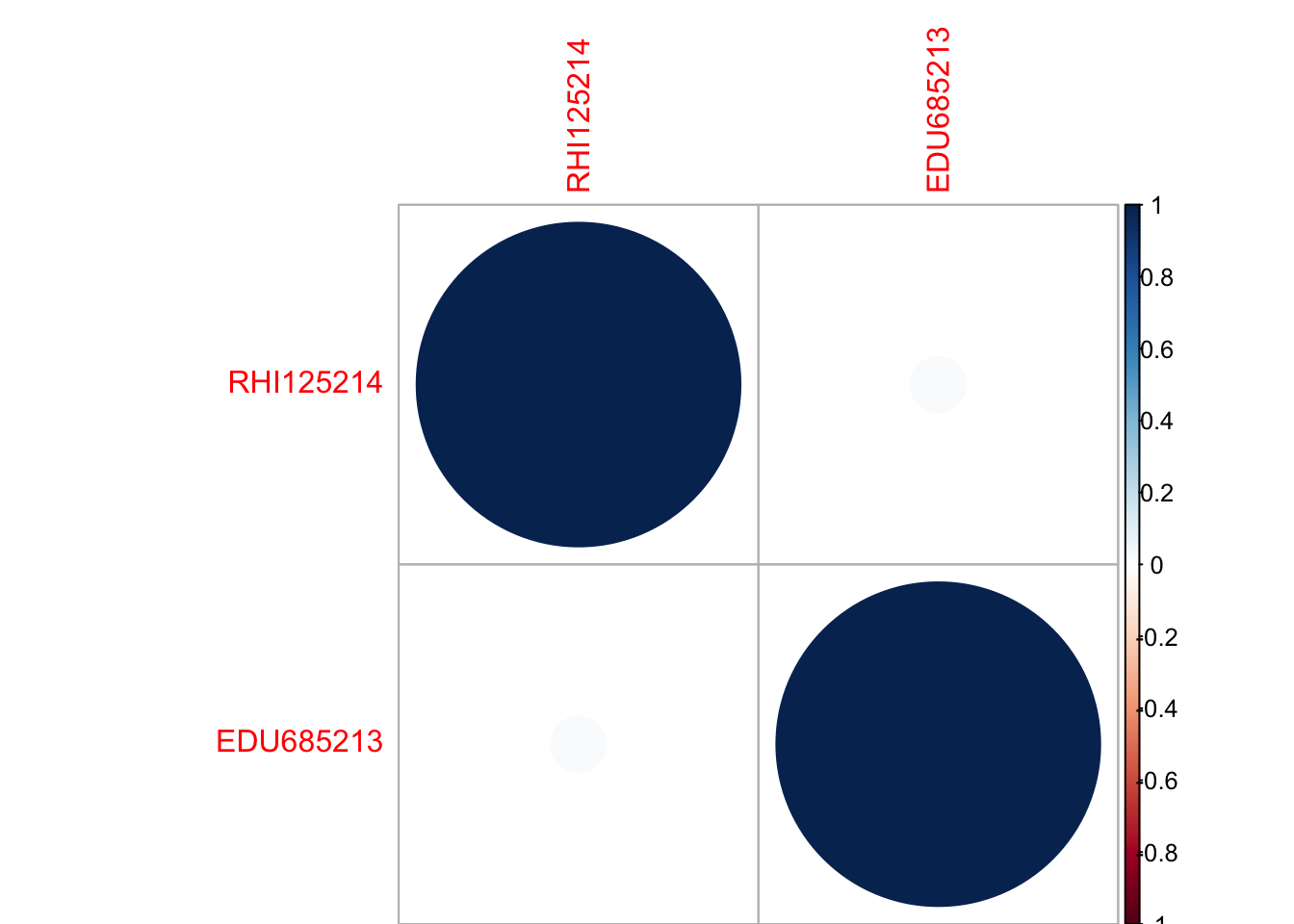


findCorrelation(eduraceR.corr, cutoff = .70, verbose = TRUE, names = TRUE, exact = TRUE)

## All correlations <= 0.7

## character(0)

#----------------------------------------------------------------------------------------  
#Finding correlation between white and education (Bachelor)  
eduraceR<-Republican.votes[,c(20,32)]  
eduraceR.corr<-cor(eduraceR)  
corrplot(eduraceR.corr)



findCorrelation(eduraceR.corr, cutoff = .70, verbose = TRUE, names = TRUE, exact = TRUE)

## All correlations <= 0.7

## character(0)

#----------------------------------------------------------------------------------------

Republican Predictive Models

Function\_Prediction <- function(testdata, model){  
 testdata$Predicted <- predict(model, testdata)  
 misClassificError\_glm\_2 <- mean(testdata$Predicted != testdata$winner)  
 return(1-misClassificError\_glm\_2)  
}  
#library(kernlab)  
#Model 1   
#Inputs: White alone, White alone(pop above 18yrs), African American, Persons below Poverty Level, High School Graduates, (25 yrs+), Bachelor???s degree, (25 yrs+), Under 18yrs, Per capita money income in past 12 months, Median household income   
LR\_TEST1 <- Function\_Prediction(testR, (multinom(winner ~ RHI825214+RHI225214+RHI125214+PVY020213+EDU685213+EDU635213+AGE295214+INC910213+INC110213, data=trainR)))

## # weights: 55 (40 variable)  
## initial value 2422.204058   
## iter 10 value 1337.333940  
## iter 20 value 1109.725701  
## iter 30 value 1088.089339  
## iter 40 value 908.902823  
## iter 50 value 882.476653  
## iter 60 value 880.893499  
## iter 70 value 880.181876  
## iter 80 value 880.051400  
## iter 90 value 880.022833  
## final value 880.022511   
## converged

SVM\_RAD\_TEST1 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+PVY020213+EDU685213+EDU635213+AGE295214+INC910213+INC110213, data=trainR,kernel= "rbfdot")))  
  
SVM\_VD\_TEST1 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+PVY020213+EDU685213+EDU635213+AGE295214+INC910213+INC110213, data=trainR,kernel= "vanilladot")))

## Setting default kernel parameters

SVM\_LP\_TEST1 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+PVY020213+EDU685213+EDU635213+AGE295214+INC910213+INC110213, data=trainR,kernel= "laplacedot")))  
  
SVM\_BD\_TEST1 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+PVY020213+EDU685213+EDU635213+AGE295214+INC910213+INC110213, data=trainR,kernel= "besseldot")))

## Setting default kernel parameters

Model1\_Accuracies <- data.frame(Algorithms = c("Logistic Regression", "SVM Radial","SVM Linear","SVM Laplacian", "SVM Bessel"), Model\_1= c(LR\_TEST1, SVM\_RAD\_TEST1, SVM\_VD\_TEST1, SVM\_LP\_TEST1, SVM\_BD\_TEST1))  
  
#Model 2  
#Inputs: White alone, African American, Persons below Poverty Level, High School Graduates, (25 yrs+), Bachelor???s degree, (25 yrs+), Under 18yrs, 65 yrs+, Per capita money income in past 12 months, Median household income, Population per Square mile   
LR\_TEST2 <- Function\_Prediction(testR, (multinom(winner ~ RHI825214+RHI225214+PVY020213+EDU685213+EDU635213+AGE775214+AGE295214+INC910213+INC110213+POP060210, data=trainR)))

## # weights: 60 (44 variable)  
## initial value 2422.204058   
## iter 10 value 1343.847308  
## iter 20 value 1039.932708  
## iter 30 value 1032.586525  
## iter 40 value 960.399362  
## iter 50 value 870.445800  
## iter 60 value 866.781172  
## iter 70 value 866.277158  
## iter 80 value 865.979188  
## iter 90 value 865.883637  
## iter 100 value 865.877431  
## final value 865.877431   
## stopped after 100 iterations

SVM\_RAD\_TEST2 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+PVY020213+EDU685213+EDU635213+AGE775214+AGE295214+INC910213+INC110213+POP060210, data=trainR,kernel= "rbfdot")))  
  
SVM\_VD\_TEST2 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+PVY020213+EDU685213+EDU635213+AGE775214+AGE295214+INC910213+INC110213+POP060210, data=trainR,kernel= "vanilladot")))

## Setting default kernel parameters

SVM\_LP\_TEST2 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+PVY020213+EDU685213+EDU635213+AGE775214+AGE295214+INC910213+INC110213+POP060210, data=trainR,kernel= "laplacedot")))  
  
SVM\_BD\_TEST2 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+PVY020213+EDU685213+EDU635213+AGE775214+AGE295214+INC910213+INC110213+POP060210, data=trainR,kernel= "besseldot")))

## Setting default kernel parameters

Model2\_Accuracies <- data.frame(Algorithms = c("Logistic Regression", "SVM Radial","SVM Linear","SVM Laplacian", "SVM Bessel"), Model\_2= c(LR\_TEST2, SVM\_RAD\_TEST2, SVM\_VD\_TEST2, SVM\_LP\_TEST2, SVM\_BD\_TEST2))  
  
#Model 3  
#Inputs: White alone, African American+White alone(pop above 18yrs), Persons below Poverty Level,High School Graduates, (25 yrs+), Bachelor???s degree, (25 yrs+), 65 yrs+, Per capita money income in past 12 months   
LR\_TEST3 <- Function\_Prediction(testR, (multinom(winner ~ RHI825214+RHI125214+RHI225214+PVY020213+EDU685213+EDU635213+AGE775214+INC910213, data=trainR)))

## # weights: 50 (36 variable)  
## initial value 2422.204058   
## iter 10 value 1068.134606  
## iter 20 value 1045.663442  
## iter 30 value 992.213819  
## iter 40 value 911.305406  
## iter 50 value 908.958091  
## iter 60 value 908.413434  
## iter 70 value 908.228319  
## iter 80 value 908.179562  
## iter 90 value 908.163677  
## iter 100 value 908.049497  
## final value 908.049497   
## stopped after 100 iterations

SVM\_RAD\_TEST3 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI125214+RHI225214+PVY020213+EDU685213+EDU635213+AGE775214+INC910213, data=trainR,kernel= "rbfdot")))  
  
SVM\_VD\_TEST3 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI125214+RHI225214+PVY020213+EDU685213+EDU635213+AGE775214+INC910213, data=trainR,kernel= "vanilladot")))

## Setting default kernel parameters

SVM\_LP\_TEST3 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI125214+RHI225214+PVY020213+EDU685213+EDU635213+AGE775214+INC910213, data=trainR,kernel= "laplacedot")))  
  
SVM\_BD\_TEST3 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI125214+RHI225214+PVY020213+EDU685213+EDU635213+AGE775214+INC910213, data=trainR,kernel= "besseldot")))

## Setting default kernel parameters

Model3\_Accuracies <- data.frame(Algorithms = c("Logistic Regression", "SVM Radial","SVM Linear","SVM Laplacian", "SVM Bessel"), Model\_3= c(LR\_TEST3, SVM\_RAD\_TEST3, SVM\_VD\_TEST3, SVM\_LP\_TEST3, SVM\_BD\_TEST3))  
  
#Model 4  
#Inputs: Hispanic or Latino, Persons below Poverty Level, High School Graduates, (25 yrs+), Bachelor???s degree, (25 yrs+), Under 18 yrs+ Female persons  
LR\_TEST4 <- Function\_Prediction(testR, (multinom(winner ~ RHI725214+PVY020213+EDU685213+EDU635213+AGE295214+SEX255214+POP815213+POP645213, data=trainR)))

## # weights: 50 (36 variable)  
## initial value 2422.204058   
## iter 10 value 1007.356848  
## iter 20 value 990.144267  
## iter 30 value 916.331707  
## iter 40 value 879.873737  
## iter 50 value 876.356388  
## iter 60 value 875.872238  
## iter 70 value 875.695743  
## iter 80 value 875.680493  
## iter 90 value 875.678247  
## iter 100 value 875.677027  
## final value 875.677027   
## stopped after 100 iterations

SVM\_RAD\_TEST4 <- Function\_Prediction(testR, (ksvm(winner ~ RHI725214+PVY020213+EDU685213+EDU635213+AGE295214+SEX255214+POP815213+POP645213, data=trainR,kernel= "rbfdot")))  
  
SVM\_VD\_TEST4 <- Function\_Prediction(testR, (ksvm(winner ~ RHI725214+PVY020213+EDU685213+EDU635213+AGE295214+SEX255214+POP815213+POP645213, data=trainR,kernel= "vanilladot")))

## Setting default kernel parameters

SVM\_LP\_TEST4 <- Function\_Prediction(testR, (ksvm(winner ~ RHI725214+PVY020213+EDU685213+EDU635213+AGE295214+SEX255214+POP815213+POP645213, data=trainR,kernel= "laplacedot")))  
  
SVM\_BD\_TEST4 <- Function\_Prediction(testR, (ksvm(winner ~ RHI725214+PVY020213+EDU685213+EDU635213+AGE295214+SEX255214+POP815213+POP645213, data=trainR,kernel= "besseldot")))

## Setting default kernel parameters

Model4\_Accuracies <- data.frame(Algorithms = c("Logistic Regression", "SVM Radial","SVM Linear","SVM Laplacian", "SVM Bessel"), Model\_4= c(LR\_TEST4, SVM\_RAD\_TEST4, SVM\_VD\_TEST4, SVM\_LP\_TEST4, SVM\_BD\_TEST4))  
  
#Model 5  
#Inputs: White alone(pop above 18yrs), African American, White alone, Persons below Poverty Level, Bachelor???s degree, (25 yrs+), High School Graduates, (25 yrs+) ,Per capita money income in past 12 months, Median household income   
LR\_TEST5 <- Function\_Prediction(testR, (multinom(winner ~ RHI825214+RHI225214+RHI125214+PVY020213+EDU685213+EDU635213+INC910213+INC110213, data=trainR)))

## # weights: 50 (36 variable)  
## initial value 2422.204058   
## iter 10 value 1328.742304  
## iter 20 value 1137.271083  
## iter 30 value 1084.224664  
## iter 40 value 924.088792  
## iter 50 value 920.587633  
## iter 60 value 919.636690  
## iter 70 value 919.340523  
## iter 80 value 919.309616  
## final value 919.309063   
## converged

SVM\_RAD\_TEST5 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+PVY020213+EDU685213+EDU635213+INC910213+INC110213, data=trainR,kernel= "rbfdot")))  
  
SVM\_VD\_TEST5 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+PVY020213+EDU685213+EDU635213+INC910213+INC110213, data=trainR,kernel= "vanilladot")))

## Setting default kernel parameters

SVM\_LP\_TEST5 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+PVY020213+EDU685213+EDU635213+INC910213+INC110213, data=trainR,kernel= "laplacedot")))  
  
SVM\_BD\_TEST5 <- Function\_Prediction(testR, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+PVY020213+EDU685213+EDU635213+INC910213+INC110213, data=trainR,kernel= "besseldot")))

## Setting default kernel parameters

Model5\_Accuracies <- data.frame(Algorithms = c("Logistic Regression", "SVM Radial","SVM Linear","SVM Laplacian", "SVM Bessel"), Model\_5= c(LR\_TEST5, SVM\_RAD\_TEST5, SVM\_VD\_TEST5, SVM\_LP\_TEST5, SVM\_BD\_TEST5))  
  
Accuracy\_1 <- merge(Model1\_Accuracies, Model2\_Accuracies, by = c("Algorithms"))  
Accuracy\_2 <- merge(Model3\_Accuracies, Model4\_Accuracies, by = c("Algorithms"))  
Accuracy\_3 <- merge(Accuracy\_1, Accuracy\_2, by = c("Algorithms"))  
Accuracy <- merge(Accuracy\_3, Model5\_Accuracies, by = c("Algorithms"))

Republican Model Accuracies

Accuracy

## Algorithms Model\_1 Model\_2 Model\_3 Model\_4 Model\_5  
## 1 Logistic Regression 0.7686170 0.7872340 0.7686170 0.7845745 0.7632979  
## 2 SVM Bessel 0.7819149 0.7632979 0.7925532 0.7686170 0.7845745  
## 3 SVM Laplacian 0.7792553 0.8031915 0.7898936 0.7872340 0.7872340  
## 4 SVM Linear 0.7898936 0.8031915 0.7632979 0.7978723 0.7659574  
## 5 SVM Radial 0.7978723 0.8111702 0.7872340 0.8005319 0.7819149

Democrat Predictive Models

Function\_Prediction <- function(testdata, model){  
 testdata$Predicted <- predict(model, testdata)  
 misClassificError\_glm\_2 <- mean(testdata$Predicted != testdata$winner)  
 return(1-misClassificError\_glm\_2)  
}  
#library(kernlab)  
#Model 1   
#Inputs: White Alone, 65 yrs, Bachelor???s degree, (25 yrs+), Median household income   
LR\_TEST1 <- Function\_Prediction(testD, (multinom(winner ~ RHI825214+RHI125214+EDU685213+INC110213+AGE775214, data=trainD)))

## # weights: 7 (6 variable)  
## initial value 1069.526100   
## iter 10 value 779.466478  
## final value 779.381362   
## converged

SVM\_RAD\_TEST1 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213+INC110213+AGE775214, data=trainD,kernel= "rbfdot")))  
  
SVM\_VD\_TEST1 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213+INC110213+AGE775214, data=trainD,kernel= "vanilladot")))

## Setting default kernel parameters

SVM\_LP\_TEST1 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213+INC110213+AGE775214, data=trainD,kernel= "laplacedot")))  
  
SVM\_BD\_TEST1 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213+INC110213+AGE775214, data=trainD,kernel= "besseldot")))

## Setting default kernel parameters

Model1\_Accuracies <- data.frame(Algorithms = c("Logistic Regression", "SVM Radial","SVM Linear","SVM Laplacian", "SVM Bessel"), MOdel\_1= c(LR\_TEST1, SVM\_RAD\_TEST1, SVM\_VD\_TEST1, SVM\_LP\_TEST1, SVM\_BD\_TEST1))  
  
#Model 2  
#Inputs: White alone, Bachelor???s degree, (25 yrs+), White alone(Not Hispanic or Latino)   
LR\_TEST2 <- Function\_Prediction(testD, (multinom(winner ~ RHI825214+RHI125214+EDU685213, data=trainD)))

## # weights: 5 (4 variable)  
## initial value 1069.526100   
## iter 10 value 795.116182  
## final value 795.107071   
## converged

SVM\_RAD\_TEST2 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213, data=trainD,kernel= "rbfdot")))  
  
SVM\_VD\_TEST2 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213, data=trainD,kernel= "vanilladot")))

## Setting default kernel parameters

SVM\_LP\_TEST2 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213, data=trainD,kernel= "laplacedot")))  
  
SVM\_BD\_TEST2 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213, data=trainD,kernel= "besseldot")))

## Setting default kernel parameters

Model2\_Accuracies <- data.frame(Algorithms = c("Logistic Regression", "SVM Radial","SVM Linear","SVM Laplacian", "SVM Bessel"), Model\_2= c(LR\_TEST2, SVM\_RAD\_TEST2, SVM\_VD\_TEST2, SVM\_LP\_TEST2, SVM\_BD\_TEST2))  
  
#Model 3  
#Inputs: White alone, Bachelor???s degree, (25 yrs+), 65 yrs  
LR\_TEST3 <- Function\_Prediction(testD, (multinom(winner ~ RHI825214+RHI125214+EDU685213+AGE775214 , data=trainD)))

## # weights: 6 (5 variable)  
## initial value 1069.526100   
## iter 10 value 785.759367  
## final value 785.755345   
## converged

SVM\_RAD\_TEST3 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213+AGE775214, data=trainD,kernel= "rbfdot")))  
  
SVM\_VD\_TEST3 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213+AGE775214, data=trainD,kernel= "vanilladot")))

## Setting default kernel parameters

SVM\_LP\_TEST3 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213+AGE775214, data=trainD,kernel= "laplacedot")))  
  
SVM\_BD\_TEST3 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI125214+EDU685213+AGE775214, data=trainD,kernel= "besseldot")))

## Setting default kernel parameters

Model3\_Accuracies <- data.frame(Algorithms = c("Logistic Regression", "SVM Radial","SVM Linear","SVM Laplacian", "SVM Bessel"), Model\_3= c(LR\_TEST3, SVM\_RAD\_TEST3, SVM\_VD\_TEST3, SVM\_LP\_TEST3, SVM\_BD\_TEST3))  
  
#Model 4  
#Inputs: African American, White alone, 65 yrs + ,Bachelor???s degree, (25 yrs+), Median household income   
LR\_TEST4 <- Function\_Prediction(testD, (multinom(winner ~ RHI225214+RHI125214+EDU685213+AGE775214+EDU685213+INC110213, data=trainD)))

## # weights: 7 (6 variable)  
## initial value 1069.526100   
## iter 10 value 662.849683  
## final value 660.212516   
## converged

SVM\_RAD\_TEST4 <- Function\_Prediction(testD, (ksvm(winner ~ RHI225214+RHI125214+EDU685213+AGE775214+EDU685213+INC110213, data=trainD,kernel= "rbfdot")))  
  
SVM\_VD\_TEST4 <- Function\_Prediction(testD, (ksvm(winner ~ RHI225214+RHI125214+EDU685213+AGE775214+EDU685213+INC110213, data=trainD,kernel= "vanilladot")))

## Setting default kernel parameters

SVM\_LP\_TEST4 <- Function\_Prediction(testD, (ksvm(winner ~ RHI225214+RHI125214+EDU685213+AGE775214+EDU685213+INC110213, data=trainD,kernel= "laplacedot")))  
  
SVM\_BD\_TEST4 <- Function\_Prediction(testD, (ksvm(winner ~ RHI225214+RHI125214+EDU685213+AGE775214+EDU685213+INC110213, data=trainD,kernel= "besseldot")))

## Setting default kernel parameters

Model4\_Accuracies <- data.frame(Algorithms = c("Logistic Regression", "SVM Radial","SVM Linear","SVM Laplacian", "SVM Bessel"), Model\_4= c(LR\_TEST4, SVM\_RAD\_TEST4, SVM\_VD\_TEST4, SVM\_LP\_TEST4, SVM\_BD\_TEST4))  
  
#Model 5  
#Inputs:White alone, White alone(pop above 18yrs), African American+High School Graduates, (25 yrs+), Bachelor???s degree, (25 yrs+), Under 18yrs   
LR\_TEST5 <- Function\_Prediction(testD, (multinom(winner ~ RHI825214+RHI225214+RHI125214+EDU685213+EDU635213+AGE295214, data=trainD)))

## # weights: 8 (7 variable)  
## initial value 1069.526100   
## iter 10 value 641.105245  
## iter 20 value 629.068894  
## final value 629.068767   
## converged

SVM\_RAD\_TEST5 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+EDU685213+EDU635213+AGE295214, data=trainD,kernel= "rbfdot")))  
  
SVM\_VD\_TEST5 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+EDU685213+EDU635213+AGE295214, data=trainD,kernel= "vanilladot")))

## Setting default kernel parameters

SVM\_LP\_TEST5 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+EDU685213+EDU635213+AGE295214, data=trainD,kernel= "laplacedot")))  
  
SVM\_BD\_TEST5 <- Function\_Prediction(testD, (ksvm(winner ~ RHI825214+RHI225214+RHI125214+EDU685213+EDU635213+AGE295214, data=trainD,kernel= "besseldot")))

## Setting default kernel parameters

Model5\_Accuracies <- data.frame(Algorithms = c("Logistic Regression", "SVM Radial","SVM Linear","SVM Laplacian", "SVM Bessel"), Model\_5= c(LR\_TEST5, SVM\_RAD\_TEST5, SVM\_VD\_TEST5, SVM\_LP\_TEST5, SVM\_BD\_TEST5))  
  
Accuracy\_1 <- merge(Model1\_Accuracies, Model2\_Accuracies, by = c("Algorithms"))  
Accuracy\_2 <- merge(Model3\_Accuracies, Model4\_Accuracies, by = c("Algorithms"))  
Accuracy\_3 <- merge(Accuracy\_1, Accuracy\_2, by = c("Algorithms"))  
Accuracy <- merge(Accuracy\_3, Model5\_Accuracies, by = c("Algorithms"))

Democrat Model Accuracies

Accuracy

## Algorithms MOdel\_1 Model\_2 Model\_3 Model\_4 Model\_5  
## 1 Logistic Regression 0.7584416 0.7818182 0.7532468 0.8103896 0.8051948  
## 2 SVM Bessel 0.7792208 0.7766234 0.7714286 0.7948052 0.8311688  
## 3 SVM Laplacian 0.7922078 0.7766234 0.7740260 0.8000000 0.8415584  
## 4 SVM Linear 0.7688312 0.7636364 0.7480519 0.7896104 0.8103896  
## 5 SVM Radial 0.7948052 0.7818182 0.7792208 0.8129870 0.8415584

Evaluating Final Models

county.demo<-read.csv("county\_facts.csv")  
maryland\_Demo <- county.demo %>%  
 filter(state\_abbreviation=="MD")  
###Output For Maryland  
  
## Republic   
Final\_Republican\_Model <- (ksvm(winner ~ RHI825214+RHI225214+RHI125214+PVY020213+EDU685213+EDU635213+AGE295214+INC910213+INC110213, data=trainR,kernel= "rbfdot"))  
testR$Predicted <- predict(Final\_Republican\_Model, testR)  
confusionMatrix(testR$Predicted, testR$winner)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Ben Carson Donald Trump John Kasich Marco Rubio Ted Cruz  
## Ben Carson 0 0 0 0 0  
## Donald Trump 0 231 10 2 40  
## John Kasich 0 0 0 0 0  
## Marco Rubio 0 0 0 0 0  
## Ted Cruz 0 23 2 0 68  
##   
## Overall Statistics  
##   
## Accuracy : 0.7952   
## 95% CI : (0.7508, 0.8349)  
## No Information Rate : 0.6755   
## P-Value [Acc > NIR] : 1.737e-07   
##   
## Kappa : 0.513   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: Ben Carson Class: Donald Trump  
## Sensitivity NA 0.9094  
## Specificity 1 0.5738  
## Pos Pred Value NA 0.8163  
## Neg Pred Value NA 0.7527  
## Prevalence 0 0.6755  
## Detection Rate 0 0.6144  
## Detection Prevalence 0 0.7527  
## Balanced Accuracy NA 0.7416  
## Class: John Kasich Class: Marco Rubio Class: Ted Cruz  
## Sensitivity 0.00000 0.000000 0.6296  
## Specificity 1.00000 1.000000 0.9067  
## Pos Pred Value NaN NaN 0.7312  
## Neg Pred Value 0.96809 0.994681 0.8587  
## Prevalence 0.03191 0.005319 0.2872  
## Detection Rate 0.00000 0.000000 0.1809  
## Detection Prevalence 0.00000 0.000000 0.2473  
## Balanced Accuracy 0.50000 0.500000 0.7682

## Democratic  
Final\_Democratic\_Model <- (ksvm(winner ~ RHI825214+RHI225214+RHI125214+EDU685213+EDU635213+AGE295214,data=trainD,kernel= "rbfdot"))  
testD$Predicted <- predict(Final\_Democratic\_Model, testD)  
confusionMatrix(testD$Predicted, testD$winner)

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Bernie Sanders Hillary Clinton  
## Bernie Sanders 69 26  
## Hillary Clinton 38 252  
##   
## Accuracy : 0.8338   
## 95% CI : (0.7927, 0.8696)  
## No Information Rate : 0.7221   
## P-Value [Acc > NIR] : 1.901e-07   
##   
## Kappa : 0.571   
## Mcnemar's Test P-Value : 0.1691   
##   
## Sensitivity : 0.6449   
## Specificity : 0.9065   
## Pos Pred Value : 0.7263   
## Neg Pred Value : 0.8690   
## Prevalence : 0.2779   
## Detection Rate : 0.1792   
## Detection Prevalence : 0.2468   
## Balanced Accuracy : 0.7757   
##   
## 'Positive' Class : Bernie Sanders   
##

maryland\_Demo$Predicted\_Republican <- predict(Final\_Republican\_Model,maryland\_Demo)  
maryland\_Demo$Predicted\_Democratic <- predict(Final\_Democratic\_Model, maryland\_Demo)

Maryland Primaries Predictions

#Republicans  
summary(maryland\_Demo$Predicted\_Republican)

## Ben Carson Donald Trump John Kasich Marco Rubio Ted Cruz   
## 0 22 0 2 0

#Democrats  
summary(maryland\_Demo$Predicted\_Democratic)

## Bernie Sanders Hillary Clinton   
## 5 19

write.csv(maryland\_Demo,"Maryland\_Predicted\_Final.csv")