CharlesCoffey17835PSET2

Charles Coffey

9/19/2020

R Markdown

Charles Coffey 17835 PSET 2 9.22.20

1.1

```
library(ggplot2)
library(matrixStats)

load("boas_hidalgo_2011.RData")

pctVV_mean = mean(br$pctVV)
pctVV_median = median(br$pctVV)
pctVV_max = max(br$pctVV)
pctVV_min = min(br$pctVV)

naive_estimator = mean(br[br$treat == 1,"pctVV"]) - mean(br[br$treat == 0,"pctVV"])

print(paste("Percent of Valid Votes Won Mean: ", toString(pctVV_mean)))
```

```
## [1] "Percent of Valid Votes Won Mean:  2.39268675306896"
```

```
print(paste("Percent of Valid Votes Won Median: ", toString(pctVV_median)))
```

```
## [1] "Percent of Valid Votes Won Median:  1.82"
```

```
print(paste("Percent of Valid Votes Won Max: ", toString(pctVV_max)))
```

```
## [1] "Percent of Valid Votes Won Max:  15.973690392295"
```

```
print(paste("Percent of Valid Votes Won Min: ", toString(pctVV_min)))
```

```
## [1] "Percent of Valid Votes Won Min:  0"
```

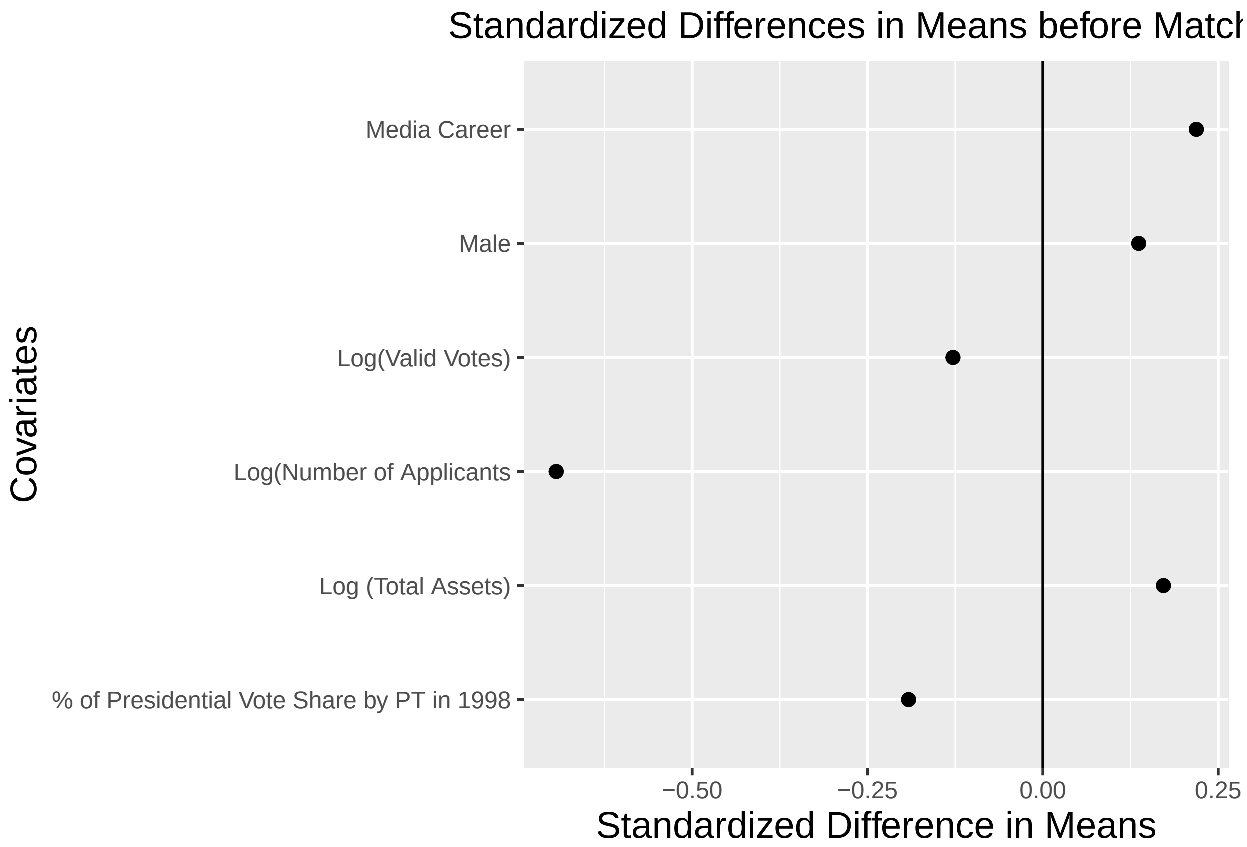
```
print(paste("Difference between Treated and Control Mean Percent of Valid Votes:", toString(naive_estima
```

```
## [1] "Difference between Treated and Control Mean Percent of Valid Votes: 0.45292954990335"
```

1.2 We cannot just compare the outcomes of politicians who received radio licenses before an election with those who did not because there may be other variables at play that contribute to the disparity between politicians. Matching, while not perfect, may be useful because it allows us to estimate what may have happened had we not introduced a treatment. We will find the control units who most closely resemble our treated units and then use their measurable data as an estimate counterfactual to the treated units' measurable data. With this estimation, we are able to approximate the effect of the treatment. This method is not perfect, but can be useful as long as we understand and acknowledge our assumptions.

1.3

```r
vars = names(br)[3:length(names(br))] # vector of covariate names

br_match_var = br[,vars] # subset of covariates to match against

std = colSds(as.matrix(br[,3:ncol(br)]))

treated_means = colMeans(br[br$treat == 1,3:ncol(br)]) #find the mean covariate values for treated unit
control_means = colMeans(br[br$treat == 0,3:ncol(br)]) #find mean covariate values for contorl units

mean_differences_std = data.frame((treated_means-control_means)/std) #standardized mean differences

names(mean_differences_std) = "mean_diff"
mean_differences_std$var <- c("Log(Valid Votes)", "% of Presidential Vote Share by PT in 1998", "Male",

#plot standardized mean differences
ggplot(mean_differences_std, aes(y=mean_diff, x=var)) + geom_hline(yintercept=0) + geom_point(size=2) +
```



The plot shows that there are some large disparities between treated and non-treated units in the dataset. We find that based on our chosen covariates, our treated candidates are very different from our control candidates. This may imply that performing causal inference could be challenging because there are such large disparities among our treated and control groups. However, this naive estimator may be problematic because we are factoring in the controls that that are very different from our treated values. If we were to estimate only using the controls that are more similar to our treated values, then we may see smaller disparities.

1.4

```r
#intitialization of mahalonobis function
MDist <- function(X, idx1, idx2){
  x1 = X[idx1,]
  x2 = X[idx2,]

  vcov = var(X)

  dist = sqrt(t(x1-x2) %*% solve(vcov) %*% (x1-x2))

  return(dist)
}
```

1.5 My model finds that the Average Treatment effect on the Treated Units is Vote Share increasing by 0.215.

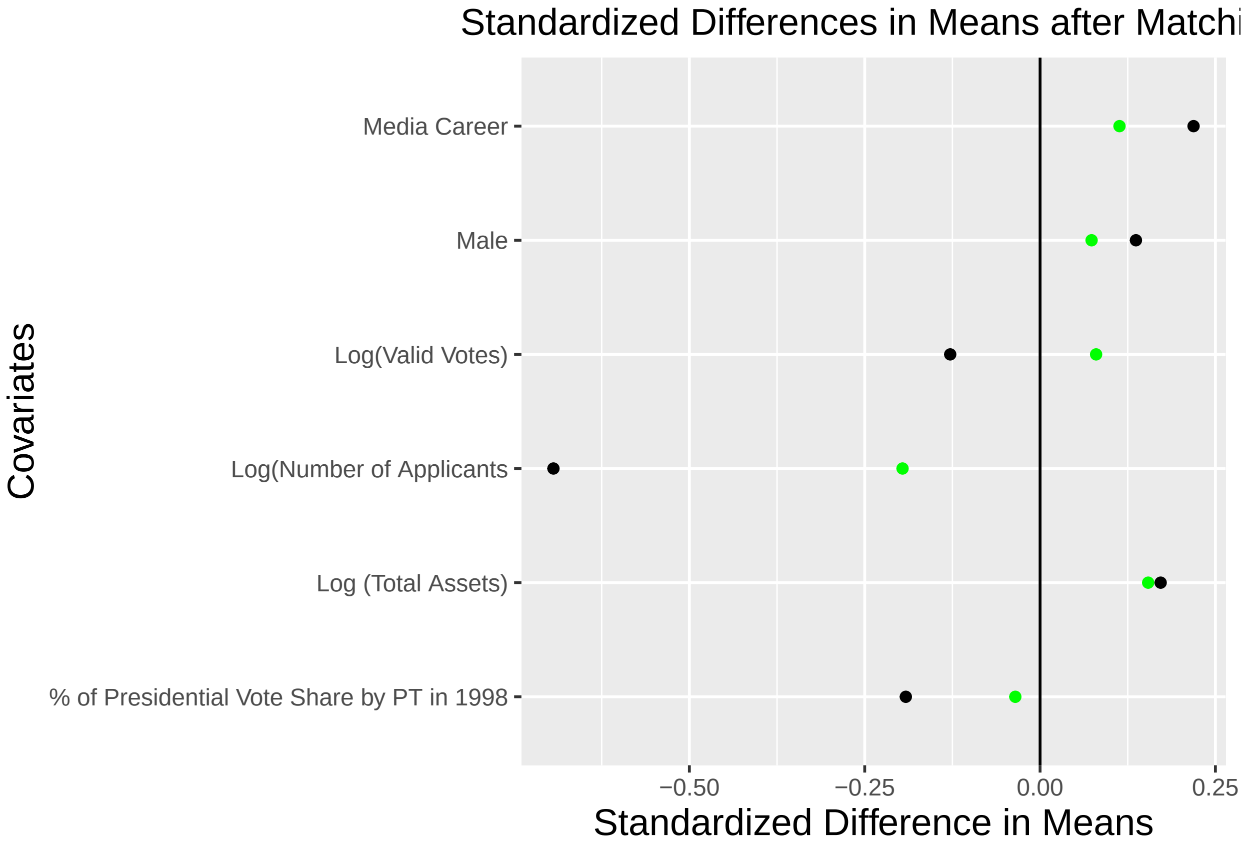
```
treated_unit_idxs = which(br$treat == 1) #indices of treated units
control_unit_idxs = which(br$treat == 0) #indices of control units

att.comb = NULL #treated and matched controls differences vector
used_control_matches = NULL #matched controls indices

for (i in treated_unit_idxs){
  #loop through treated values

  dist_M = NULL #Mahalanobis distances for each control

  for (j in control_unit_idxs){
    #get mahalonobis distnace values between one treated and all controls
    dist = MDist(as.matrix(br_match_var), i, j)
    dist_M =  append(dist_M, dist)
  }

  # distance measures for each of the controls
  control_matches = data.frame(cbind(control_unit_idxs, dist_M))

  #top two best matches
  best_matches = control_matches[order(dist_M), ][1:2, ]

  used_control_matches = append(used_control_matches, best_matches$control_unit_idxs)

  att = br[i,"pctVV"] - mean(br[best_matches$control_unit_idxs, "pctVV"])

  att.comb = append(att.comb, att)

}

#average treatment effect of radio licenses on vote share
ATT = mean(att.comb)
print(paste("The Average Treatment Effect on the Treated Units is:", toString(ATT)))
```

```
## [1] "The Average Treatment Effect on the Treated Units is: 0.215775850598631"
```

This output implies that having the radio license on average increases a candidates vote share by 0.21%. Considering that the mean vote share among all candidates is 2.39%, the ATT shows us that, on average, having a radio license shows not very large, but non-neligible increase in vote share.

1.6

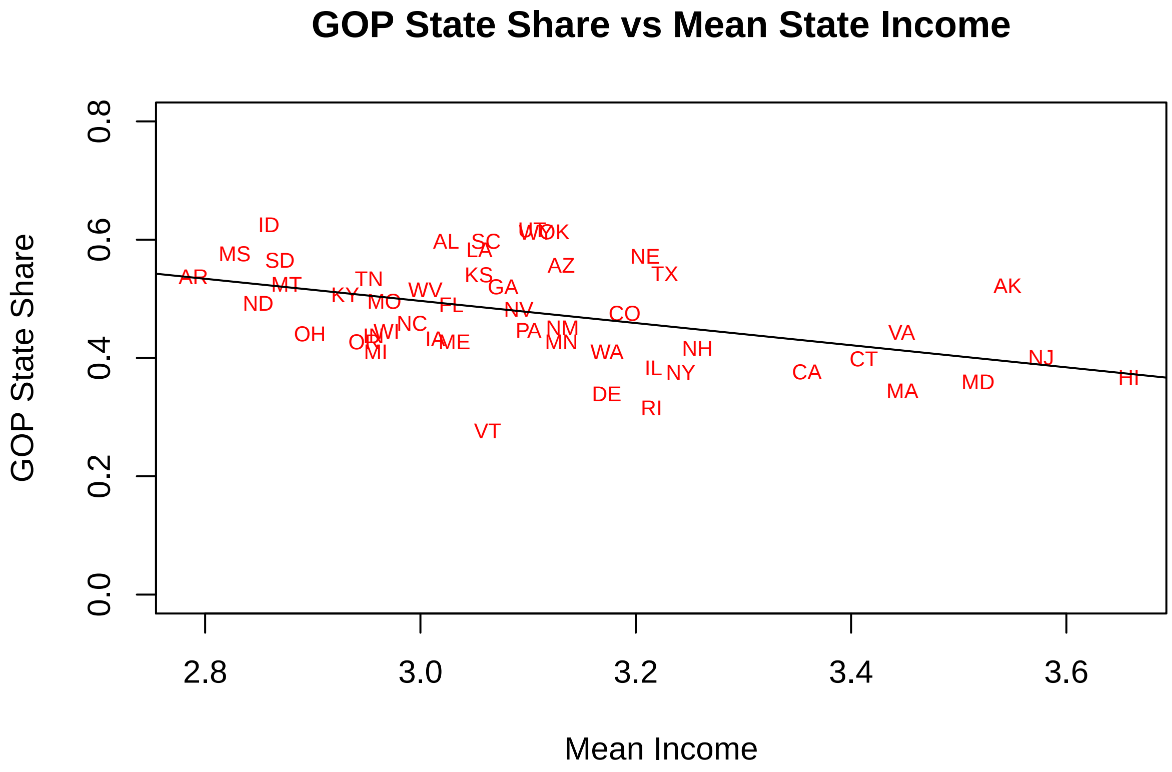
```
used_control_matches_unique = unique(used_control_matches)
not_used_controls = control_unit_idxs[!control_unit_idxs %in% used_control_matches_unique ]

br_subset = br[-not_used_controls,] #this subset contains just the treated values and controls used for

std_match = colSds(as.matrix(br_subset[,3:ncol(br_subset)])) #std of treated and matched controls

control_means_match = colMeans(br_subset[br_subset$treat == 0,3:ncol(br_subset)]) #means of all matched

mean_differences_std_match = data.frame((treated_means-control_means_match)/std_match) #standardized me
names(mean_differences_std_match) = "mean_diff_matched"

mean_differences_combined = cbind(mean_differences_std, mean_differences_std_match)

#plot old differences before and after matching
ggplot(mean_differences_combined, aes(x=var, y = mean_diff)) + geom_point(aes(y=mean_diff_matched), col
```



This data shows that matching improved the balance between treated and control units. The post-matching values (green dots) for each covariate is closer to 0 than the pre-matching values (black dots). This indicates the the balance between treated and matched units improved overall. This balance could be improved even further by leveling out different covariate variables or standardizing them amonng candidates. This could perhaps manifest by all candidates having the same amount of social media activity or all politicians having similar total assets. The balance could also be affected by choosing different covariates; choosing other factors as being the most important. Another way to pontially balance it further would be to use only the best matches as counterfactuals and not the best two. I believe that we should trust the matching estimator more because we can see that disparities between treated units and matched controls are greatly decreased after matching. With less disparities, we see that the units are more similar and this then allows us to more confidently draw comparisons between them. This is more like comparing apples to apples instead of apples to oranges.

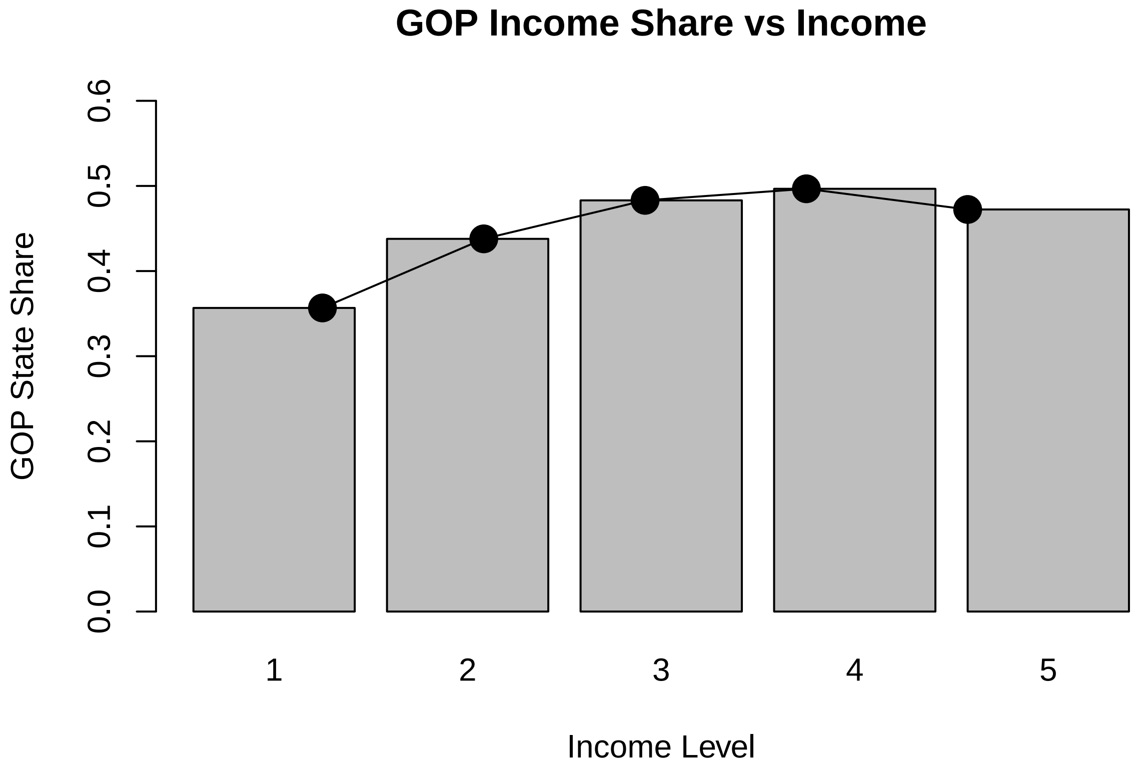
2.1

```r
cces_2012 <- read.csv("cces_2012.csv")

#calculate mean GOP vote share per state
#this could be done in one step by just setting the function to mean
gop_state_vote = tapply(cces_2012$vote_gop, INDEX = cces_2012$state_abb, FUN = sum)
gop_state_n = tapply(cces_2012$vote_gop, INDEX = cces_2012$state_abb, FUN = length)
gop_state_share = gop_state_vote/gop_state_n

#calculate mean income per state
income_state = tapply(cces_2012$income, INDEX = cces_2012$state_abb, FUN=mean)


plot(income_state, gop_state_share, ylim = c(0, 0.8), "n",
     main = "GOP State Share vs Mean State Income",
     ylab = "GOP State Share",
     xlab = "Mean Income")

text(income_state, gop_state_share, names(gop_state_share), col="red", cex = 0.65)

abline(lm(gop_state_share ~ income_state))
```



2.2

```r
#calculate GOP vote share across incomes
gop_income_vote = tapply(cces_2012$vote_gop, INDEX = cces_2012$income, FUN = sum)
gop_income_n = tapply(cces_2012$vote_gop, INDEX = cces_2012$income, FUN = length)
gop_income_share = gop_income_vote/gop_income_n

barplot(gop_income_share, names.arg= names(gop_income_share), ylim = c(0, 0.6),
        main = "GOP Income Share vs Income",
        xlab = "Income Level",
        ylab = "GOP State Share")

points(c(1:length(gop_income_share)), gop_income_share, pch = 16, cex = 2, ylim = c(0, 0.6))
lines(c(1:length(gop_income_share)), gop_income_share)
```

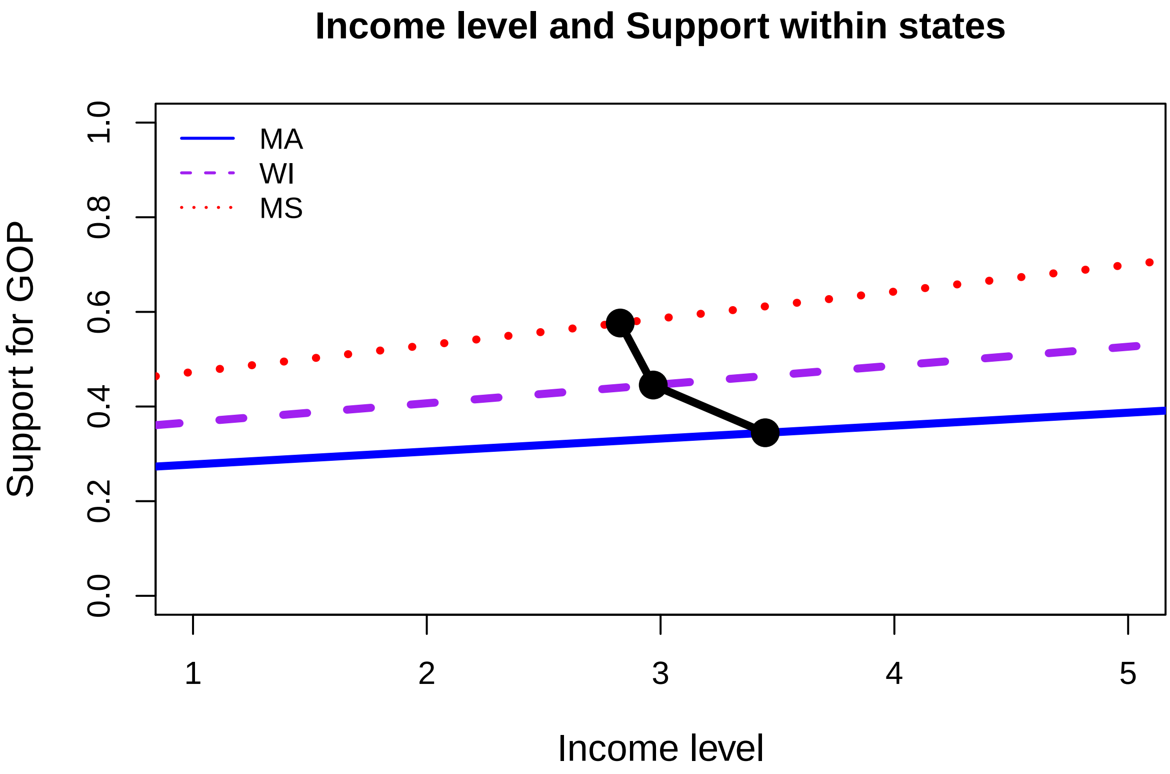


The bar plot shows that Voter Income when plotted against GOP Vote Share has a positive slope. This means that generally, as Voter Income increases, so does GOP Vote Share. This information is interesting when compared to the scatter plot. The scatter plot shows that when Average State Income is plotted against GOP Vote Share, the slope is negative. This indicates that throughout the states, the GOP generally has less vote share in those with higher average incomes. Those states with lower average incomes tend to have higher GOP vote shares. This shows that the voters for the GOP tend to have higher incomes while the the states voting for the GOP actually have lower average incomes.

2.3

```
#creat subset data for each state
MA_subset = cces_2012[cces_2012$state_abb == "MA", c("vote_gop", "income")]
WI_subset = cces_2012[cces_2012$state_abb == "WI", c("vote_gop", "income")]
MS_subset = cces_2012[cces_2012$state_abb == "MS", c("vote_gop", "income")]


plot(0,
     type = 'n',
     xlim = c(1, 5),
     ylim = c(0, 1),
     main = "Income level and Support within states",
     xlab = "Income level",
     ylab = "Support for GOP",
     cex.lab = 1.2)
```

```r
abline(lm(MA_subset), col="blue", lty = 1, lwd = 4)
abline(lm(WI_subset), col=" purple", lty = 2, lwd = 4)
abline(lm(MS_subset), col="red", lty = 3, lwd = 4)

legend("topleft",
       legend = c("MA", "WI", "MS"),
       lwd = c(1.5,1.5,1.5),
       lty = 1:3,
       cex = 0.9,
       col = c("blue", "purple", "red"),
       bty = "n")

#find average gop vote share and income values for each state
state_average_values = data.frame()
state_average_values = rbind(state_average_values, colMeans(MA_subset), colMeans(WI_subset), colMeans(MS
names(state_average_values) = c("gop_vote_mean", "income_mean")

points(state_average_values[,2], state_average_values[,1], pch = 16, cex = 2)

lines(state_average_values[,2], state_average_values[,1], lwd = 4)
```



2.4

Based on this figure, we find that in MS, the poorest state, the GOP has the highest number of voter share as compared to the richer states of WI and MA. MS also has the steepest meaning thatthe GOP has the greatest increase in voter share per unit change in income level in MS. MA has the shallowest slope, this indicates that it is the most resistant to voting for the GOP after inceases in income level. The slopes of all of these lines shows us that as income level increases, GOP vote share increases in all of these states. This perhaps suggests that this kind of trend transcends across richer and poorer states.

We find the correlation coefficient, or slope, for MS to be bigger. This indicates the GOP vote share has larger increases per unit increase in income level in MS vs MA. The coefficients tell us that MS, the poorer state, is more prone to vote for the GOP as individuals attain more income.

```
MA_corr = cor(MA_subset$income, MA_subset$vote_gop)
MS_corr = cor(MS_subset$income, MS_subset$vote_gop)

print(paste("MA indv. income to indv. support for GOP candidate correlation coefficient: ", toString(MA_
```

```
## [1] "MA indv. income to indv. support for GOP candidate correlation coefficient:  0.0731169989448954
```

```
print(paste("MS indv. income to indv. support for GOP candidate correlation coefficient: ", toString(MS_
```

```
## [1] "MS indv. income to indv. support for GOP candidate correlation coefficient:  0.148840331949988"
```