

Hyperparameter Tuning

Team Rho

2025-04-28

```
#libraries
```

```
library(readxl)
library(caret)
library(tidyr)
library(dplyr)
library(corrplot)
library(rvest)
library(glmnet)
library(pls)
```

```
#reading data
```

```
# data_GTrends <- read_excel("C:/data_science/DSE63110M/week3_Exploratory Data Analysis/googleTrendsMH")
# acs_data <- load("C:\\data_science\\DSE63110M\\Week4_Pre-processing and Feature Engineering\\ACS_for")
# Replace with this code to allow quick access regardless of download location
data_GTrends <- read_excel("~/GitHub/DSE63110M_SP2025R2_Data-Science-Capstone/Data/googleTrendsMH.xlsx"
  sheet = "googleTrendsMH")
acs_data <- load("~/GitHub/DSE63110M_SP2025R2_Data-Science-Capstone/Data/ACS_for_MHGoogleTrends.Rdata")

acs_data <- ACS_data
ACS_data <- NULL
```

```
##CORRELATION MATRIX FOR acs_data
```

```
acs_correlation_matrix <- acs_data %>%
  select_if(is.numeric) %>%
  select(-prop_persons_below_poverty_threshold, -prop_veterans_disability) %>%
  cor()

print(acs_correlation_matrix)
```

```
##
##          year prop_families_below_poverty
## year          1.00000000          -0.1610309
## prop_families_below_poverty      -0.16103094           1.0000000
## prop_adults_without_health_insurance -0.35051348           0.1974453
## prop_unemployed_in_labor_force      -0.50071692           0.6113240
## prop_without_internet_access         0.31496819           0.3030755
## prop_adult_disability                0.04834553           0.5972604
##          prop_adults_without_health_insurance
## year          -0.3505135
```

```

## prop_families_below_poverty                0.1974453
## prop_adults_without_health_insurance        1.0000000
## prop_unemployed_in_labor_force              0.2889701
## prop_without_internet_access                -0.1226758
## prop_adult_disability                       0.1945398
##
##                                prop_unemployed_in_labor_force
## year                                -0.5007169
## prop_families_below_poverty            0.6113240
## prop_adults_without_health_insurance    0.2889701
## prop_unemployed_in_labor_force          1.0000000
## prop_without_internet_access            -0.1705119
## prop_adult_disability                   0.1723363
##
##                                prop_without_internet_access
## year                                0.3149682
## prop_families_below_poverty            0.3030755
## prop_adults_without_health_insurance    -0.1226758
## prop_unemployed_in_labor_force          -0.1705119
## prop_without_internet_access            1.0000000
## prop_adult_disability                   0.3494365
##
##                                prop_adult_disability
## year                                0.04834553
## prop_families_below_poverty            0.59726036
## prop_adults_without_health_insurance    0.19453980
## prop_unemployed_in_labor_force          0.17233629
## prop_without_internet_access            0.34943653
## prop_adult_disability                   1.00000000

```

updated code

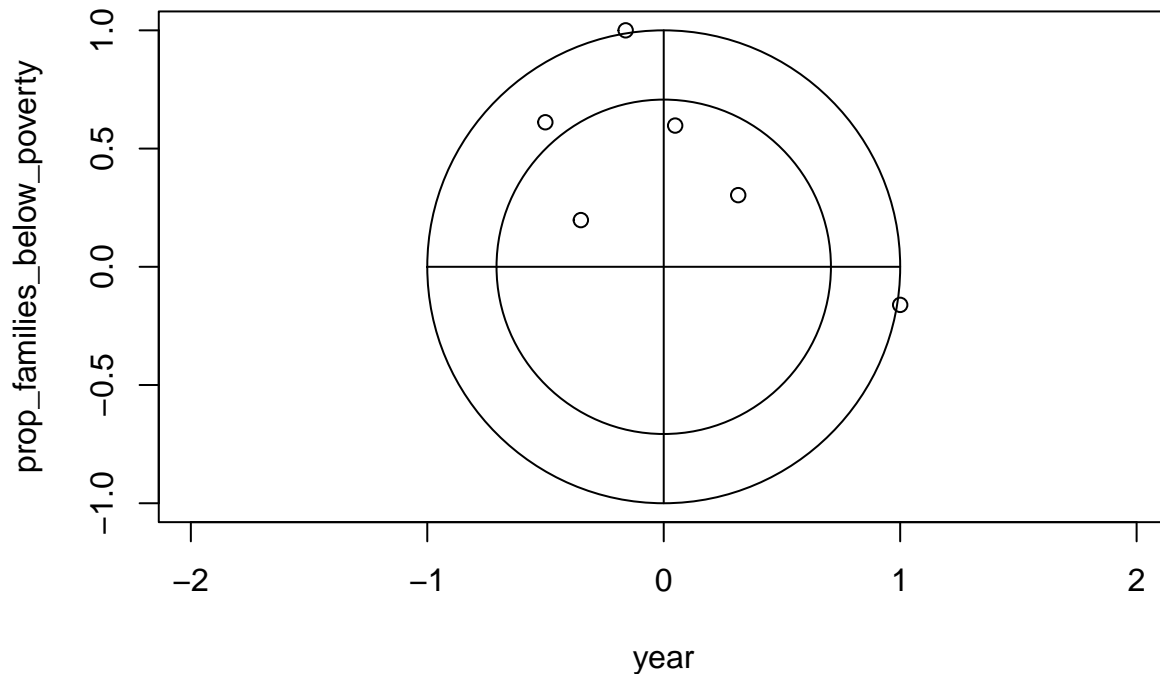
```

#presenting correlation matrix in graphic format

acs_correlation_matrix <- acs_data %>%
  select_if(is.numeric) %>%
  select(-prop_persons_below_poverty_threshold, -prop_veterans_disability) %>%
  cor() %>%
  corrplot( diag = F,
            tl.cex = 0.7,
            tl.col = "black",
            main = "acs_data correlation matrix",
            mar = c(0,0,1,0))

```

acs_data correlation matrix



#removing correlated features

```
acs_data_clean <- acs_data %>%
  select(-prop_persons_below_poverty_threshold, -prop_veterans_disability)
```

convert state names into abbreviation to match state in data_GTrends

```
acs_data_clean$state <- toupper(state.abb[match(tolower(acs_data_clean$state), tolower(state.name))])
```

testing GitHub

#data transformations ct variables

#creating response variable => state_mentalhealth_utili = state_psych_care / population_est

#state_mentalhealth_utili <- data_GTrends\$state_psych_care / data_GTrends\$population_est

```
data_GTrends <- data_GTrends %>%
  mutate(state_mentalhealth_utili = state_psych_care/population_est,
         anxiety_prop = anxiety_ct/ population_est,
         trauma_stress_prop = trauma_stress_ct/population_est,
         adhd_prop = adhd_ct/population_est,
         bipolar_prop = bipolar_ct/population_est,
         depression_prop = depression_ct/population_est)
```

#data_GTrends <- data_GTrends %>%

#select(-state_psych_care, -anxiety_ct, -trauma_stress_ct, -adhd_ct, -bipolar_ct, -depression_ct) all

```
#joining both datasets acs_data and data_GTrends
```

```
GTrends_acs_joined <- inner_join(data_GTrends, acs_data_clean, by = c("year", "state"))
```

```
#testing correlation
```

```
correlation_matrix <- GTrends_acs_joined %>%
```

```
  select_if(is.numeric) %>%
```

```
  select(-fips, -population_est, -private_psych_care, -total_util, -outpatient_util, -mean_anxiety, -res,
        -total_util) %>%
```

```
  cor()
```

```
print(correlation_matrix)
```

```
##              year  anxiety_ct  trauma_stress_ct
## year          1.00000000  0.230563501    0.13366856
## anxiety_ct     0.23056350  1.000000000    0.92240079
## trauma_stress_ct 0.13366856  0.922400795    1.00000000
## adhd_ct        0.01851770  0.847645702    0.87161036
## bipolar_ct     -0.13690754  0.653131435    0.75571956
## depression_ct   0.06120702  0.873780027    0.94087338
## comm_psych_care 0.05264059  0.793626073    0.89977194
## state_psych_care 0.05220254  0.800842275    0.90248691
## mean_adhd      0.75682637  0.192811841    0.08958471
## mean_ptsd       0.62228218  0.090669189    0.04475684
## mean_bipolar   -0.09097469 -0.085128361    -0.08423315
## mean_depression -0.02390143  0.009319898    -0.02136263
## mean_mental_hospital 0.27777930  0.319455125    0.28112091
## mean_psychiatrists_near_me 0.18697534  0.063526502    0.09919989
## mean_psychologist_near_me 0.64878930  0.404062943    0.38356349
## anxiety_prop    0.25256530  0.575638687    0.40794338
## adhd_prop       0.02582844  0.540119606    0.44884626
## bipolar_prop    -0.27713846  0.402247684    0.39406527
## prop_families_below_poverty -0.31411265 -0.065951520    -0.02266406
## prop_adults_without_health_insurance -0.35036488 -0.120820100    -0.08943951
## prop_unemployed_in_labor_force -0.54031845 -0.047006409    0.07676369
## prop_without_internet_access 0.31423583  0.011777977    -0.03506000
## prop_adult_disability 0.07154859 -0.089418168    -0.12802032
##              adhd_ct  bipolar_ct  depression_ct
## year          0.018517704 -0.13690754  0.06120702
## anxiety_ct     0.847645702  0.65313144  0.87378003
## trauma_stress_ct 0.871610355  0.75571956  0.94087338
## adhd_ct        1.000000000  0.83440163  0.90823233
## bipolar_ct     0.834401629  1.00000000  0.88673220
## depression_ct   0.908232333  0.88673220  1.00000000
## comm_psych_care 0.874225711  0.87090215  0.95667411
## state_psych_care 0.884006979  0.87166405  0.95701158
## mean_adhd      -0.007745775 -0.10866030  0.02253769
## mean_ptsd       -0.124707857 -0.22821302  -0.08131642
## mean_bipolar    -0.082850695 -0.03030126  -0.08659302
## mean_depression -0.026389005 -0.09361394  -0.02884011
## mean_mental_hospital 0.220054198  0.21655455  0.28147786
```

## mean_psychiatrists_near_me	0.086212620	0.06521304	0.09221333
## mean_psychologist_near_me	0.316683082	0.20732437	0.35169600
## anxiety_prop	0.306023903	0.03211950	0.27306557
## adhd_prop	0.557691198	0.19368296	0.36224924
## bipolar_prop	0.458390120	0.36562312	0.36378200
## prop_families_below_poverty	0.091452450	0.21421452	0.06093810
## prop_adults_without_health_insurance	0.001121328	0.24369742	0.03448441
## prop_unemployed_in_labor_force	0.124358517	0.28278587	0.13217179
## prop_without_internet_access	0.010097643	-0.11859483	-0.03027184
## prop_adult_disability	-0.041620397	-0.11618594	-0.11834226
##	comm_psych_care	state_psych_care	
## year	0.05264059	0.05220254	
## anxiety_ct	0.79362607	0.80084228	
## trauma_stress_ct	0.89977194	0.90248691	
## adhd_ct	0.87422571	0.88400698	
## bipolar_ct	0.87090215	0.87166405	
## depression_ct	0.95667411	0.95701158	
## comm_psych_care	1.00000000	0.99936080	
## state_psych_care	0.99936080	1.00000000	
## mean_adhd	0.01154550	0.01301038	
## mean_ptsd	-0.09505592	-0.09409334	
## mean_bipolar	-0.06243299	-0.06269307	
## mean_depression	-0.04094749	-0.04237320	
## mean_mental_hospital	0.24373032	0.24415647	
## mean_psychiatrists_near_me	0.13571311	0.13354197	
## mean_psychologist_near_me	0.36100819	0.35825438	
## anxiety_prop	0.18813746	0.20049138	
## adhd_prop	0.28982510	0.30527377	
## bipolar_prop	0.30483675	0.31814831	
## prop_families_below_poverty	0.06341390	0.06303851	
## prop_adults_without_health_insurance	0.02920460	0.02820942	
## prop_unemployed_in_labor_force	0.16815934	0.16554652	
## prop_without_internet_access	-0.03609294	-0.03484673	
## prop_adult_disability	-0.15530682	-0.14673191	
##	mean_adhd	mean_ptsd	mean_bipolar
## year	0.756826372	0.62228218	-0.090974692
## anxiety_ct	0.192811841	0.09066919	-0.085128361
## trauma_stress_ct	0.089584712	0.04475684	-0.084233146
## adhd_ct	-0.007745775	-0.12470786	-0.082850695
## bipolar_ct	-0.108660303	-0.22821302	-0.030301260
## depression_ct	0.022537693	-0.08131642	-0.086593022
## comm_psych_care	0.011545502	-0.09505592	-0.062432992
## state_psych_care	0.013010379	-0.09409334	-0.062693072
## mean_adhd	1.000000000	0.42495384	0.179510680
## mean_ptsd	0.424953840	1.000000000	0.193509244
## mean_bipolar	0.179510680	0.19350924	1.000000000
## mean_depression	-0.245750075	0.41128942	0.308755245
## mean_mental_hospital	0.287677009	0.09702821	0.232486981
## mean_psychiatrists_near_me	0.042769431	0.05674090	-0.005280538
## mean_psychologist_near_me	0.415735545	0.23433255	-0.080183845
## anxiety_prop	0.222753634	0.30520691	-0.005956554
## adhd_prop	0.028590323	0.09085592	-0.010212770
## bipolar_prop	-0.159049076	-0.04663275	0.157398435
## prop_families_below_poverty	-0.208577621	-0.20391856	0.293106346

```

## prop_adults_without_health_insurance -0.186412427 -0.24473889 0.233057761
## prop_unemployed_in_labor_force -0.327758496 -0.43653037 0.157300589
## prop_without_internet_access -0.126520915 0.33393361 -0.090016482
## prop_adult_disability 0.109982033 0.10629585 0.222236769
## mean_depression mean_mental_hospital
## year -0.023901425 0.27777930
## anxiety_ct 0.009319898 0.31945513
## trauma_stress_ct -0.021362629 0.28112091
## adhd_ct -0.026389005 0.22005420
## bipolar_ct -0.093613944 0.21655455
## depression_ct -0.028840113 0.28147786
## comm_psych_care -0.040947486 0.24373032
## state_psych_care -0.042373199 0.24415647
## mean_adhd -0.245750075 0.28767701
## mean_ptsd 0.411289416 0.09702821
## mean_bipolar 0.308755245 0.23248698
## mean_depression 1.000000000 -0.10548867
## mean_mental_hospital -0.105488666 1.000000000
## mean_psychiatrists_near_me 0.001374564 0.15614239
## mean_psychologist_near_me -0.098056483 0.41633384
## anxiety_prop 0.050429764 0.02664347
## adhd_prop 0.069487449 -0.06288825
## bipolar_prop 0.026384149 -0.09485722
## prop_families_below_poverty -0.077146712 0.21535926
## prop_adults_without_health_insurance -0.062380502 -0.02688604
## prop_unemployed_in_labor_force -0.348426242 0.10886182
## prop_without_internet_access 0.385215253 0.07508085
## prop_adult_disability -0.081676556 0.16483923
## mean_psychiatrists_near_me
## year 0.186975337
## anxiety_ct 0.063526502
## trauma_stress_ct 0.099199887
## adhd_ct 0.086212620
## bipolar_ct 0.065213036
## depression_ct 0.092213328
## comm_psych_care 0.135713106
## state_psych_care 0.133541968
## mean_adhd 0.042769431
## mean_ptsd 0.056740904
## mean_bipolar -0.005280538
## mean_depression 0.001374564
## mean_mental_hospital 0.156142388
## mean_psychiatrists_near_me 1.000000000
## mean_psychologist_near_me 0.466711912
## anxiety_prop -0.104990533
## adhd_prop -0.105489672
## bipolar_prop -0.156142069
## prop_families_below_poverty -0.185544042
## prop_adults_without_health_insurance -0.257450224
## prop_unemployed_in_labor_force -0.020698183
## prop_without_internet_access 0.051130358
## prop_adult_disability -0.239770625
## mean_psychologist_near_me anxiety_prop
## year 0.64878930 0.252565296

```

## anxiety_ct	0.40406294	0.575638687
## trauma_stress_ct	0.38356349	0.407943378
## adhd_ct	0.31668308	0.306023903
## bipolar_ct	0.20732437	0.032119498
## depression_ct	0.35169600	0.273065574
## comm_psych_care	0.36100819	0.188137462
## state_psych_care	0.35825438	0.200491380
## mean_adhd	0.41573555	0.222753634
## mean_ptsd	0.23433255	0.305206913
## mean_bipolar	-0.08018385	-0.005956554
## mean_depression	-0.09805648	0.050429764
## mean_mental_hospital	0.41633384	0.026643466
## mean_psychiatrists_near_me	0.46671191	-0.104990533
## mean_psychologist_near_me	1.00000000	0.018713136
## anxiety_prop	0.01871314	1.000000000
## adhd_prop	-0.02192663	0.772593545
## bipolar_prop	-0.20102389	0.592973858
## prop_families_below_poverty	-0.16397365	-0.139411004
## prop_adults_without_health_insurance	-0.20618180	-0.202330161
## prop_unemployed_in_labor_force	-0.18536934	-0.244392365
## prop_without_internet_access	0.15990322	0.090420463
## prop_adult_disability	-0.08569762	0.099264075
##	adhd_prop	bipolar_prop
## year	0.02582844	-0.27713846
## anxiety_ct	0.54011961	0.40224768
## trauma_stress_ct	0.44884626	0.39406527
## adhd_ct	0.55769120	0.45839012
## bipolar_ct	0.19368296	0.36562312
## depression_ct	0.36224924	0.36378200
## comm_psych_care	0.28982510	0.30483675
## state_psych_care	0.30527377	0.31814831
## mean_adhd	0.02859032	-0.15904908
## mean_ptsd	0.09085592	-0.04663275
## mean_bipolar	-0.01021277	0.15739843
## mean_depression	0.06948745	0.02638415
## mean_mental_hospital	-0.06288825	-0.09485722
## mean_psychiatrists_near_me	-0.10548967	-0.15614207
## mean_psychologist_near_me	-0.02192663	-0.20102389
## anxiety_prop	0.77259354	0.59297386
## adhd_prop	1.00000000	0.73676449
## bipolar_prop	0.73676449	1.00000000
## prop_families_below_poverty	0.06474605	0.24288704
## prop_adults_without_health_insurance	-0.10333794	0.15947980
## prop_unemployed_in_labor_force	-0.06381305	0.17824936
## prop_without_internet_access	0.10675502	-0.09079816
## prop_adult_disability	0.20587109	0.24830497
##	prop_families_below_poverty	
## year		-0.31411265
## anxiety_ct		-0.06595152
## trauma_stress_ct		-0.02266406
## adhd_ct		0.09145245
## bipolar_ct		0.21421452
## depression_ct		0.06093810
## comm_psych_care		0.06341390

```

## state_psych_care 0.06303851
## mean_adhd -0.20857762
## mean_ptsd -0.20391856
## mean_bipolar 0.29310635
## mean_depression -0.07714671
## mean_mental_hospital 0.21535926
## mean_psychiatrists_near_me -0.18554404
## mean_psychologist_near_me -0.16397365
## anxiety_prop -0.13941100
## adhd_prop 0.06474605
## bipolar_prop 0.24288704
## prop_families_below_poverty 1.00000000
## prop_adults_without_health_insurance 0.60329043
## prop_unemployed_in_labor_force 0.52364772
## prop_without_internet_access 0.12312374
## prop_adult_disability 0.65543780
## prop_adults_without_health_insurance
## year -0.350364883
## anxiety_ct -0.120820100
## trauma_stress_ct -0.089439512
## adhd_ct 0.001121328
## bipolar_ct 0.243697423
## depression_ct 0.034484408
## comm_psych_care 0.029204600
## state_psych_care 0.028209419
## mean_adhd -0.186412427
## mean_ptsd -0.244738889
## mean_bipolar 0.233057761
## mean_depression -0.062380502
## mean_mental_hospital -0.026886042
## mean_psychiatrists_near_me -0.257450224
## mean_psychologist_near_me -0.206181798
## anxiety_prop -0.202330161
## adhd_prop -0.103337943
## bipolar_prop 0.159479797
## prop_families_below_poverty 0.603290434
## prop_adults_without_health_insurance 1.000000000
## prop_unemployed_in_labor_force 0.409465887
## prop_without_internet_access -0.106556672
## prop_adult_disability 0.289928013
## prop_unemployed_in_labor_force
## year -0.54031845
## anxiety_ct -0.04700641
## trauma_stress_ct 0.07676369
## adhd_ct 0.12435852
## bipolar_ct 0.28278587
## depression_ct 0.13217179
## comm_psych_care 0.16815934
## state_psych_care 0.16554652
## mean_adhd -0.32775850
## mean_ptsd -0.43653037
## mean_bipolar 0.15730059
## mean_depression -0.34842624
## mean_mental_hospital 0.10886182

```



```

## mean_psychiatrists_near_me -0.02069818
## mean_psychologist_near_me -0.18536934
## anxiety_prop -0.24439237
## adhd_prop -0.06381305
## bipolar_prop 0.17824936
## prop_families_below_poverty 0.52364772
## prop_adults_without_health_insurance 0.40946589
## prop_unemployed_in_labor_force 1.00000000
## prop_without_internet_access -0.34452758
## prop_adult_disability 0.06756309
## prop_without_internet_access
## year 0.31423583
## anxiety_ct 0.01177798
## trauma_stress_ct -0.03506000
## adhd_ct 0.01009764
## bipolar_ct -0.11859483
## depression_ct -0.03027184
## comm_psych_care -0.03609294
## state_psych_care -0.03484673
## mean_adhd -0.12652092
## mean_ptsd 0.33393361
## mean_bipolar -0.09001648
## mean_depression 0.38521525
## mean_mental_hospital 0.07508085
## mean_psychiatrists_near_me 0.05113036
## mean_psychologist_near_me 0.15990322
## anxiety_prop 0.09042046
## adhd_prop 0.10675502
## bipolar_prop -0.09079816
## prop_families_below_poverty 0.12312374
## prop_adults_without_health_insurance -0.10655667
## prop_unemployed_in_labor_force -0.34452758
## prop_without_internet_access 1.00000000
## prop_adult_disability 0.30396009
## prop_adult_disability
## year 0.07154859
## anxiety_ct -0.08941817
## trauma_stress_ct -0.12802032
## adhd_ct -0.04162040
## bipolar_ct -0.11618594
## depression_ct -0.11834226
## comm_psych_care -0.15530682
## state_psych_care -0.14673191
## mean_adhd 0.10998203
## mean_ptsd 0.10629585
## mean_bipolar 0.22223677
## mean_depression -0.08167656
## mean_mental_hospital 0.16483923
## mean_psychiatrists_near_me -0.23977062
## mean_psychologist_near_me -0.08569762
## anxiety_prop 0.09926407
## adhd_prop 0.20587109
## bipolar_prop 0.24830497
## prop_families_below_poverty 0.65543780

```

```
## prop_adults_without_health_insurance      0.28992801
## prop_unemployed_in_labor_force            0.06756309
## prop_without_internet_access              0.30396009
## prop_adult_disability                     1.00000000
```

high correlation variables

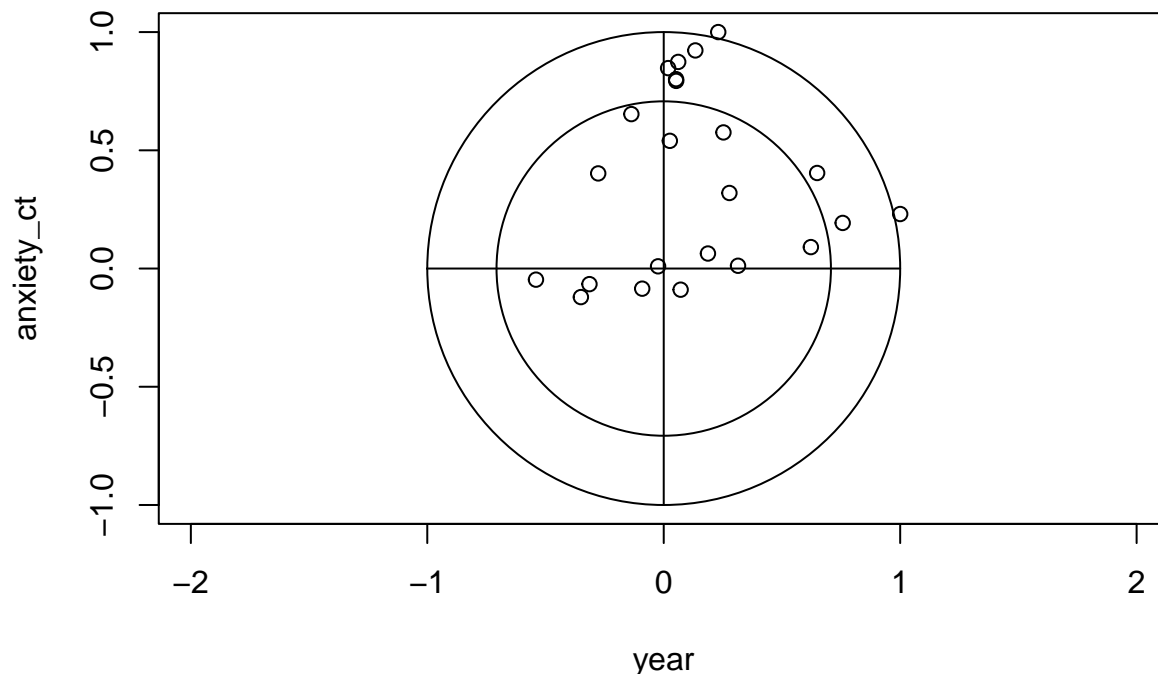
1. private, reside and comm_psych_care, 2.inpatient_util vs outpatient_util (i already have state_mentalhealth_util) 3.mean_therapist_near_me vs mean_psychiatrist and mean_psychologist
- 4.mean_alltrend vs mean_adhd, mean_ptsd, mean_anxiety, mean_mentalhospital.
2. mean_anxiety vs year, mean_adhd & ptsd 6.outpatient_util vs total_util, adhd, bipolar & depression
- 7.total_util 8.depression prob vs adhd. ptsd, bipolar and trauma_stress_prop 9.trauma_stress_prop vs adhd, anxiety_prop and state_mentalhealth_util 10.state_mentalhealth_util vs adhd, ptsd, bipolar

#correlation matrix

```
GTrends_acs_joined %>%
  select_if(is.numeric) %>%
  select(-fips, -population_est, -private_psych_care, -total_util, -outpatient_util, -mean_anxiety, -res,
         -total_util) %>%
  cor() %>%

corrplot(diag = F,
         tl.cex = 0.7,
         tl.col = "black",
         main = "Correlation Matrix of GTrends_acs_joined",
         mar = c(0, 0, 1, 0))
```

Correlation Matrix of GTrends_acs_joined

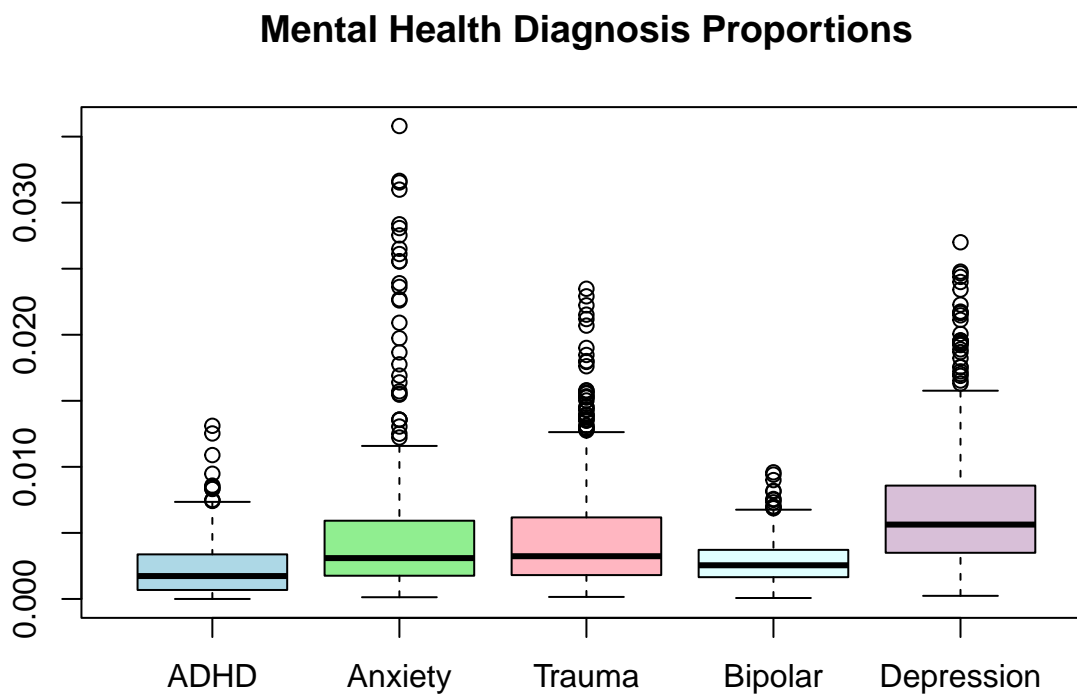


```

data <- data_GTrends
data$adhd_prop= data$adhd_ct/data$population_est
data$anxiety_prop = data$anxiety_ct/data$population_est
data$bipolar_prop = data$bipolar_ct/data$population_est
data$depression_prop = data$depression_ct/data$population_est
data$trauma_prop = data$trauma_stress_ct/data$population_est
data$state_util = data$state_psych_care/data$population_est
data$private_util = data$private_psych_care/data$population_est
data$diff_util = data$total_util-(data$state_util + data$private_util)

boxplot(data[c("adhd_prop", "anxiety_prop", "trauma_prop", "bipolar_prop", "depression_prop")],
  main = "Mental Health Diagnosis Proportions",
  names = c("ADHD", "Anxiety", "Trauma", "Bipolar", "Depression"),
  col = c("lightblue", "lightgreen", "lightpink", "lightcyan", "thistle" ))

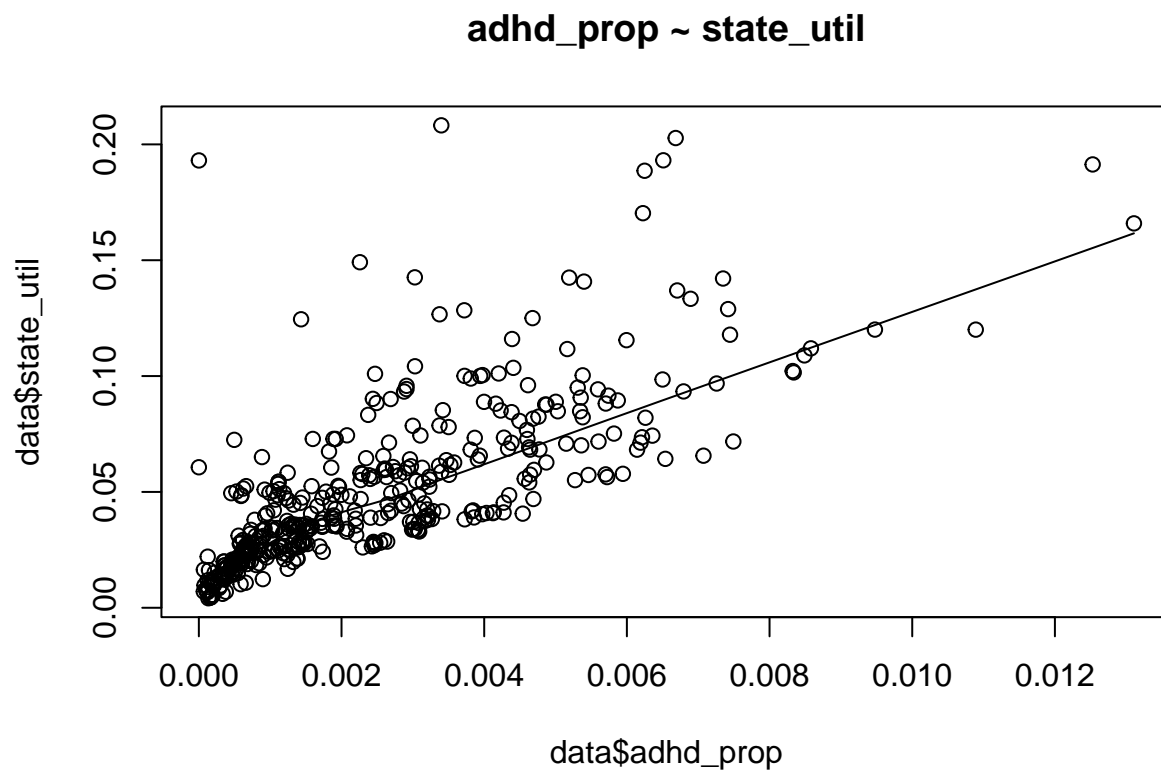
```



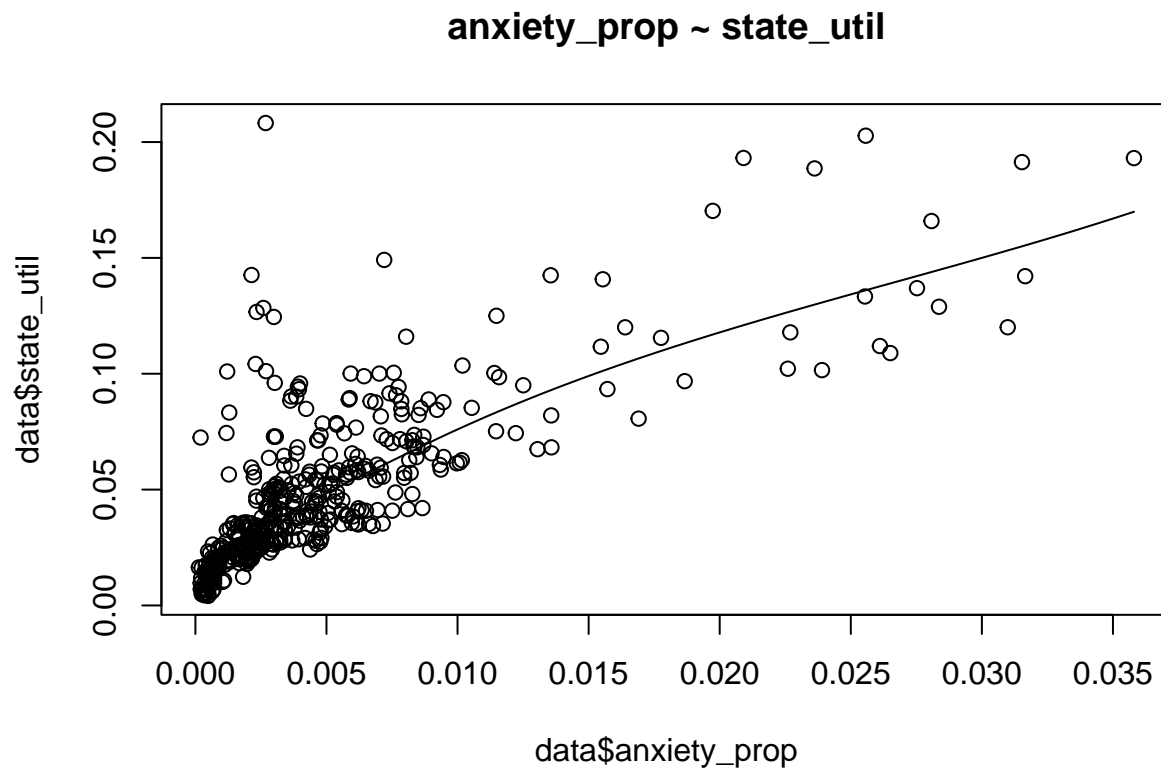
```

par(mfrow=c(1,1)) # divide graph area in 2 columns
scatter.smooth(x=data$adhd_prop, y=data$state_util, main="adhd_prop ~ state_util")

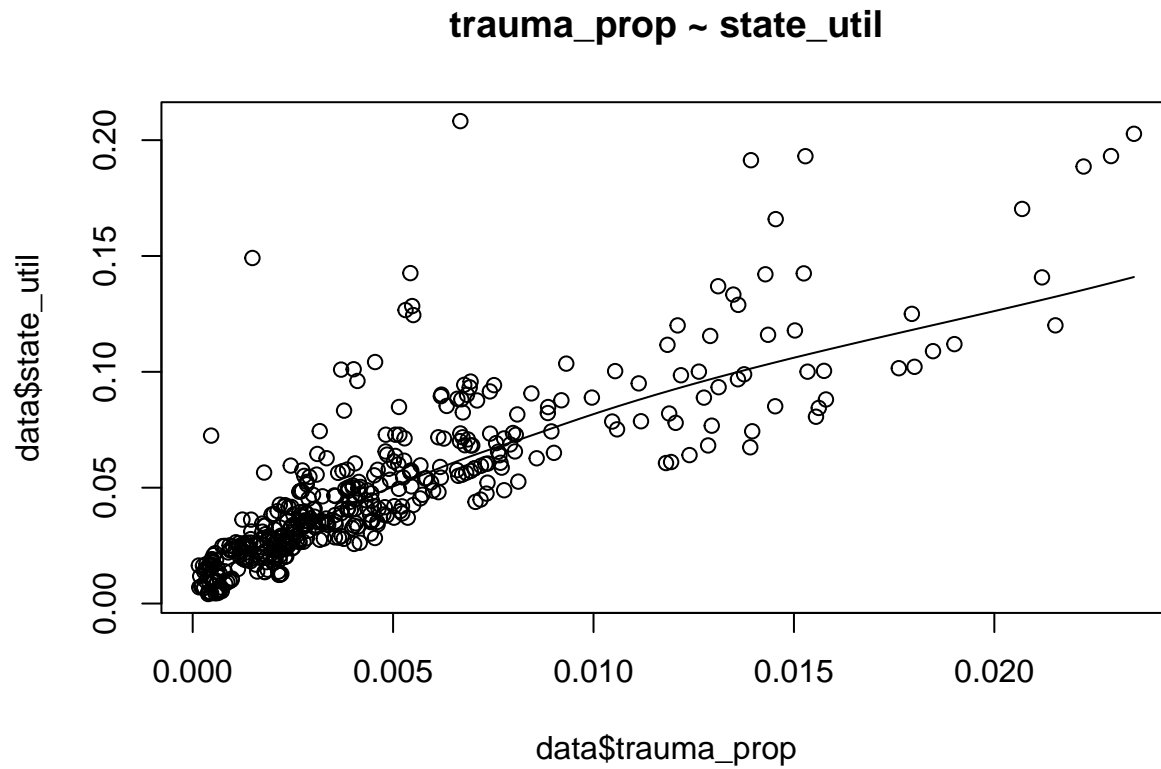
```



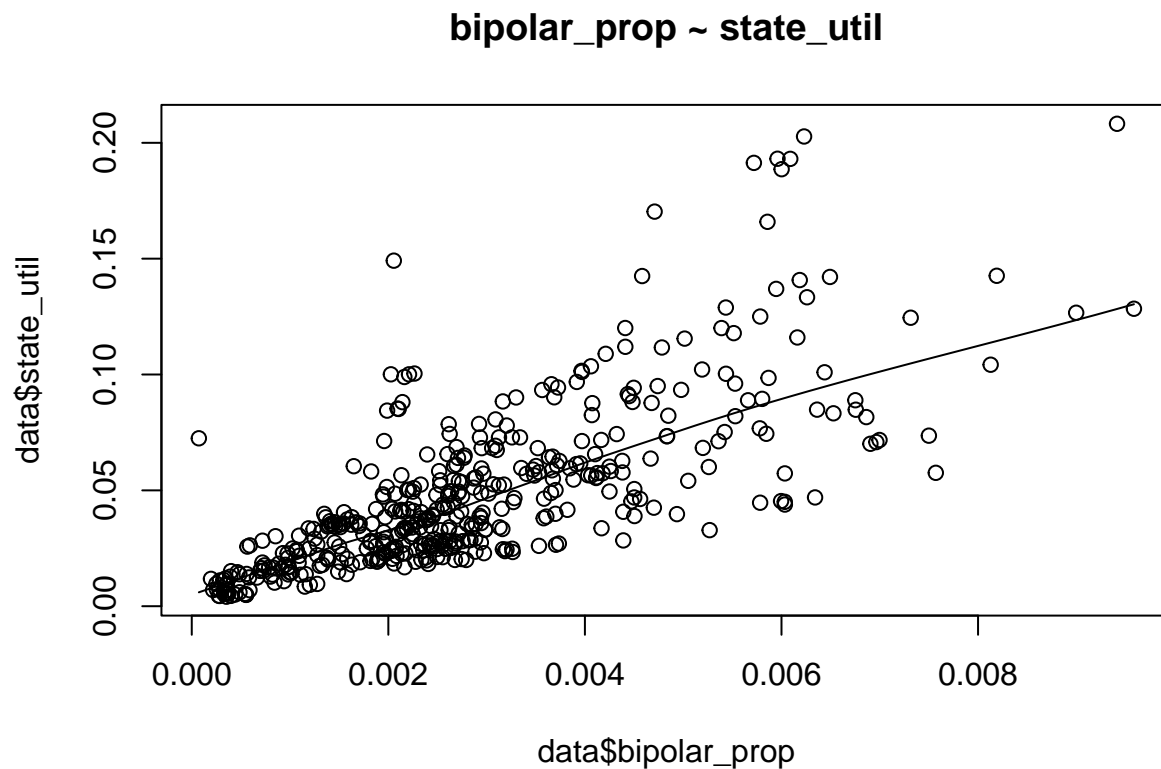
```
scatter.smooth(x=data$anxiety_prop, y=data$state_util, main="anxiety_prop ~ state_util")
```



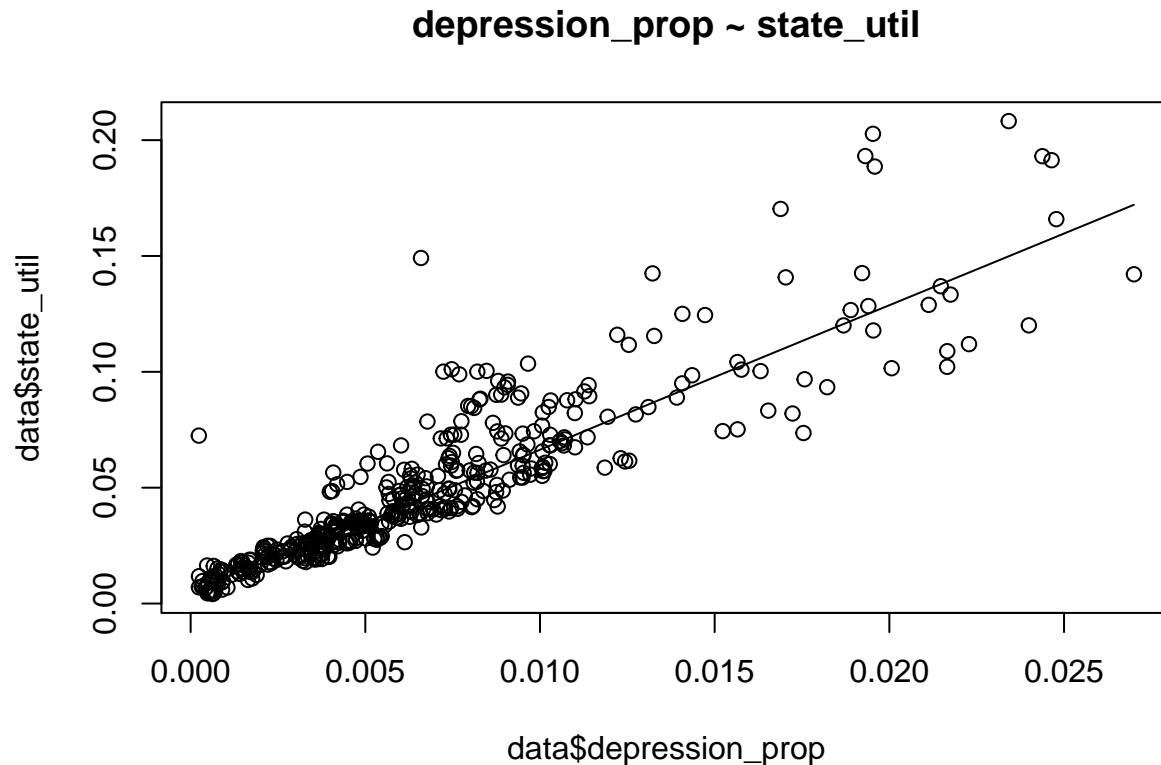
```
scatter.smooth(x=data$trauma_prop, y=data$state_util, main="trauma_prop ~ state_util")
```



```
scatter.smooth(x=data$bipolar_prop, y=data$state_util, main="bipolar_prop ~ state_util")
```



```
scatter.smooth(x=data$depression_prop, y=data$state_util, main="depression_prop ~ state_util")
```



```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.3
```

```
par(mfrow=c(1, 1))
```

```
# Create a density plot that shows public, private, and total mental healthcare utilization rate  
# frequency
```

```
plot(density(data$state_util),  
     main = "Public, Private Facility, & Total Utilization Density",  
     ylab = "Frequency",  
     xlab = "Utilization Rate",  
     col = "green",  
     lwd = 2,  
     sub = paste("Skewness (State):", round(e1071::skewness(data$state_util), 2)))
```

```
# Fill the first density with polygon
```

```
polygon(density(data$state_util), col = adjustcolor("lightgreen", alpha.f = 0.5), border = NA)
```

```
# Add second density line
```

```
lines(density(data$private_util), col = "blue", lwd = 2)
```

```
polygon(density(data$private_util), col = adjustcolor("lightblue", alpha.f = 0.5), border = NA)
```

```
# Add third density line
```

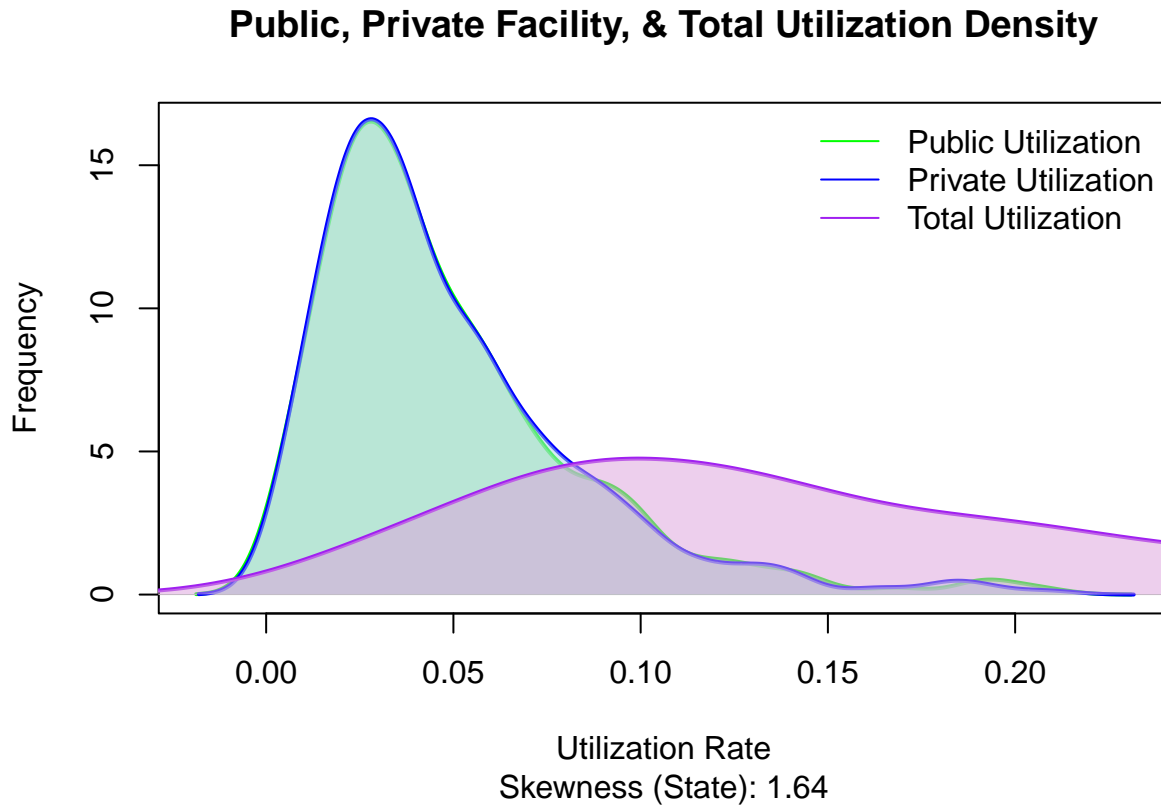
```
lines(density(data$total_util), col = "purple", lwd = 2)
```

```

polygon(density(data$total_util), col = adjustcolor("plum", alpha.f = 0.5), border = NA)

# Add legend
legend("topright", legend = c("Public Utilization", "Private Utilization", "Total Utilization "),
      col = c("green", "blue", "purple"), lwd = 1, bty = "n")

```



```

#data split: train and test dataset

clean_GTrends_acs_joined <- GTrends_acs_joined %>%
  select(-fips, -population_est, -private_psych_care, -total_util,
        -outpatient_util, -region, -mean_anxiety, -resid_psych_care,
        #-outpatient_util, -mean_anxiety, -resid_psych_care,
        -mean_all_trends, -mean_therapist_near_me, -depression_prop,
        -trauma_stress_prop, -inpatient_util,
        -contains(c("median", "total")), -total_util) #i have added region as part of eliminated featu

test_n <- (1/sqrt(19))*nrow(clean_GTrends_acs_joined)
test_prop <- round((1/sqrt(19))*nrow(clean_GTrends_acs_joined)/nrow(clean_GTrends_acs_joined), 2)
train_prop <- 1-test_prop

paste("The ideal split ratio is", train_prop, ":", test_prop, " training : testing")

## [1] "The ideal split ratio is 0.77 : 0.23  training : testing"

# Show the dimensions of the dataframe and the column names.
dim(clean_GTrends_acs_joined)

```

```
## [1] 433 25
```

```
names(clean_GTrends_acs_joined)
```

```
## [1] "year"
## [2] "state"
## [3] "anxiety_ct"
## [4] "trauma_stress_ct"
## [5] "adhd_ct"
## [6] "bipolar_ct"
## [7] "depression_ct"
## [8] "comm_psych_care"
## [9] "state_psych_care"
## [10] "mean_adhd"
## [11] "mean_ptsd"
## [12] "mean_bipolar"
## [13] "mean_depression"
## [14] "mean_mental_hospital"
## [15] "mean_psychiatrists_near_me"
## [16] "mean_psychologist_near_me"
## [17] "state_mentalhealth_util"
## [18] "anxiety_prop"
## [19] "adhd_prop"
## [20] "bipolar_prop"
## [21] "prop_families_below_poverty"
## [22] "prop_adults_without_health_insurance"
## [23] "prop_unemployed_in_labor_force"
## [24] "prop_without_internet_access"
## [25] "prop_adult_disability"
```

```
# Remove some fields used in the calculation of the proportions
```

```
cols_to_exclude = c("anxiety_ct",
                    "trauma_stress_ct",
                    "adhd_ct", "bipolar_ct",
                    "depression_ct",
                    "comm_psych_care",
                    "state_psych_care")
clean_GTrends_acs_joined <- clean_GTrends_acs_joined[,!(names(clean_GTrends_acs_joined)
                                                         %in% cols_to_exclude)]
names(clean_GTrends_acs_joined)
```

```
## [1] "year"
## [2] "state"
## [3] "mean_adhd"
## [4] "mean_ptsd"
## [5] "mean_bipolar"
## [6] "mean_depression"
## [7] "mean_mental_hospital"
## [8] "mean_psychiatrists_near_me"
## [9] "mean_psychologist_near_me"
## [10] "state_mentalhealth_util"
## [11] "anxiety_prop"
## [12] "adhd_prop"
```



```
## [13] "bipolar_prop"
## [14] "prop_families_below_poverty"
## [15] "prop_adults_without_health_insurance"
## [16] "prop_unemployed_in_labor_force"
## [17] "prop_without_internet_access"
## [18] "prop_adult_disability"
```

```
#write the merged dataframe to a CSV file with a time stamp in the name.
# This way we don't overwrite the file in case someone else is working on the file.
# timeStamp <- format(Sys.time(), "%Y%m%d_%H%M%S")
# file_name <- paste("~/GitHub/DSE63110M_SP2025R2_Data-Science-Capstone/Data/clean_GTrends_acs_joined_"
# write.csv(clean_GTrends_acs_joined, file_name, row.names = FALSE)
```

```
train <- createDataPartition(clean_GTrends_acs_joined$state_mentalhealth_util,
                             p = 0.77,
                             list = FALSE,
                             times = 1)
```

```
GTrend_training_set <- clean_GTrends_acs_joined[train, ]
```

```
test_set <- clean_GTrends_acs_joined[-train, ]
```

```
dim(GTrend_training_set)
```

```
## [1] 336 18
```

```
dim(test_set)
```

```
## [1] 97 18
```

TARGET ENCODING OF STATE BY Njagi

```
unique(clean_GTrends_acs_joined$state)
```

```
## [1] "AL" "AZ" "AR" "CA" "CO" "CT" "DE" "FL" "HI" "ID" "IL" "IN" "IA" "KS" "KY"
## [16] "LA" "MA" "MS" "MO" "MT" "NE" "NV" "NJ" "NM" "NY" "NC" "ND" "OH" "OK" "OR"
## [31] "PA" "RI" "SC" "SD" "TN" "TX" "UT" "VT" "VA" "WA" "WI" "WY" "MN" "MI" "AK"
## [46] "GA"
```

```
is.factor(clean_GTrends_acs_joined$state) #checking whether region is a factor = false
```

```
## [1] FALSE
```

```
GTrend_training_set$state <- factor(GTrend_training_set$state)
```

```
class(GTrend_training_set$state)
```

```
## [1] "factor"
```

```
levels(GTrend_training_set$state)
```

```
## [1] "AK" "AL" "AR" "AZ" "CA" "CO" "CT" "DE" "FL" "GA" "HI" "IA" "ID" "IL" "IN"
## [16] "KS" "KY" "LA" "MA" "MI" "MN" "MO" "MS" "MT" "NC" "ND" "NE" "NJ" "NM" "NV"
## [31] "NY" "OH" "OK" "OR" "PA" "RI" "SC" "SD" "TN" "TX" "UT" "VA" "VT" "WA" "WI"
## [46] "WY"
```

we are going to apply target encoding (state_mentalhealth_util). To avoid overfitting we are going to use a smoothed version of target encoding

```
main_mean <- mean( GTrend_training_set$state_mentalhealth_util)
```

```
smoothing_factor <- 10
```

#calculating the smoothed state means from the training set

```
state_encoded_by_smoothedmean <- GTrend_training_set %>%
  group_by(state) %>%
  summarise(state_encoded = (mean(state_mentalhealth_util) * n() + main_mean * smoothing_factor) / (n() + smoothing_factor))
```

#merging the smoothed encoded state means with the training set

```
GTrend_training_set_f <- GTrend_training_set %>%
  left_join(state_encoded_by_smoothedmean, by = "state") %>%
  select(-state)
```

#merging smoothed encoded state means with the test_set

```
test_set$state <- factor(test_set$state)
```

```
test_set_f <- test_set %>%
  left_join(state_encoded_by_smoothedmean, by = "state") %>%
  select(-state)
```

```
dim(test_set_f)
```

```
## [1] 97 18
```

#center and scale

```
# test_set_f[, c(-10)] <- scale(test_set_f[, c(-10)],
#                               center = apply(GTrend_training_set_f[, c(-10)], 2, mean),
#                               scale = apply(GTrend_training_set_f[, c(-10)], 2, sd))
```

```
state_util_index <-
test_set_f[, c(-10)] <- scale(test_set_f[, c(-10)],
                              center = apply(GTrend_training_set_f[, c(-10)], 2, mean),
                              scale = apply(GTrend_training_set_f[, c(-10)], 2, sd))
```

#(-10) is the state_mentalhealth_util, i want to exclude it from center and scale since its already a predictor

```
GTrend_training_set_f[, -10] <- scale(GTrend_training_set_f[, -10])
```

```
head(GTrend_training_set_f)
```

```
## # A tibble: 6 x 18
##   year mean_adhd mean_ptsd mean_bipolar mean_depression mean_mental_hospital
##   <dbl>   <dbl>   <dbl>       <dbl>         <dbl>         <dbl>
## 1 -1.58   -0.329   -1.79         0.947         -1.13         -0.247
## 2 -1.58   -0.704   -1.03         0.748         -0.647         -0.295
## 3 -1.58   -0.576   -1.41         0.798         -0.957          0.157
## 4 -1.58   -0.256   -1.76         2.59         -0.194         -2.35
## 5 -1.58   -0.549    0.0103       -0.445        -0.284         -2.11
## 6 -1.58   -0.576   -0.786        0.798         0.402         -0.698
## # i 12 more variables: mean_psychiatrists_near_me <dbl>,
## #   mean_psychologist_near_me <dbl>, state_mentalhealth_util <dbl>,
## #   anxiety_prop <dbl>, adhd_prop <dbl>, bipolar_prop <dbl>,
## #   prop_families_below_poverty <dbl>,
## #   prop_adults_without_health_insurance <dbl>,
## #   prop_unemployed_in_labor_force <dbl>, prop_without_internet_access <dbl>,
## #   prop_adult_disability <dbl>, state_encoded <dbl>
```

```
#generating codebook
```

```
library(tibble)
```

```
codebook <- tibble(
  variable = names(clean_GTrends_acs_joined),
  class = sapply(clean_GTrends_acs_joined, class),
  "Number of Missing Values" = sapply(clean_GTrends_acs_joined, function(x) sum(is.na(x))),
  "Number of Unique Values" = sapply(clean_GTrends_acs_joined, function(x) length(unique(x)))
)
```

```
print(codebook)
```

```
## # A tibble: 18 x 4
##   variable                class Number of Missing Va~1 Number of Unique Val~2
##   <chr>                   <chr>           <int>           <int>
## 1 year                   nume~              0              10
## 2 state                  char~              0              46
## 3 mean_adhd              nume~              0             205
## 4 mean_ptsd              nume~              0             114
## 5 mean_bipolar           nume~              0              97
## 6 mean_depression        nume~              0             230
## 7 mean_mental_hospital   nume~              0             272
## 8 mean_psychiatrists_near~ nume~              0              59
## 9 mean_psychologist_near_me nume~              0             153
## 10 state_mentalhealth_util nume~              0             433
## 11 anxiety_prop          nume~              0             433
## 12 adhd_prop             nume~              0             433
## 13 bipolar_prop          nume~              0             433
## 14 prop_families_below_pove~ nume~              0             433
```

```
## 15 prop_adults_without_heal~ nume~ 0 433
## 16 prop_unemployed_in_labor~ nume~ 0 433
## 17 prop_without_internet_ac~ nume~ 0 433
## 18 prop_adult_disability nume~ 0 433
## # i abbreviated names: 1: 'Number of Missing Values',
## # 2: 'Number of Unique Values'
```

```
codebook$variable
```

```
## [1] "year"
## [2] "state"
## [3] "mean_adhd"
## [4] "mean_ptsd"
## [5] "mean_bipolar"
## [6] "mean_depression"
## [7] "mean_mental_hospital"
## [8] "mean_psychiatrists_near_me"
## [9] "mean_psychologist_near_me"
## [10] "state_mentalhealth_util"
## [11] "anxiety_prop"
## [12] "adhd_prop"
## [13] "bipolar_prop"
## [14] "prop_families_below_poverty"
## [15] "prop_adults_without_health_insurance"
## [16] "prop_unemployed_in_labor_force"
## [17] "prop_without_internet_access"
## [18] "prop_adult_disability"
```

```
# Create an empty dataframe with three fields store storing model train and test RMSE values.
```

```
mse_df <- tibble(
  Model = character(),
  Train_MSE = numeric(),
  Test_MSE = numeric()
)

# Function to add rows to the mse_df
add_rmse_row <- function(df, model_name, train_mse, test_mse) {
  new_row <- tibble(
    Model = model_name,
    Train_MSE = train_mse,
    Test_MSE = test_mse
  )
  updated_df <- bind_rows(df, new_row)
  return(updated_df)
}
```

INITIAL MODELS BY Njagi

1. LINEAR REGRESSION (ELASTIC NET REGULARIZATION)

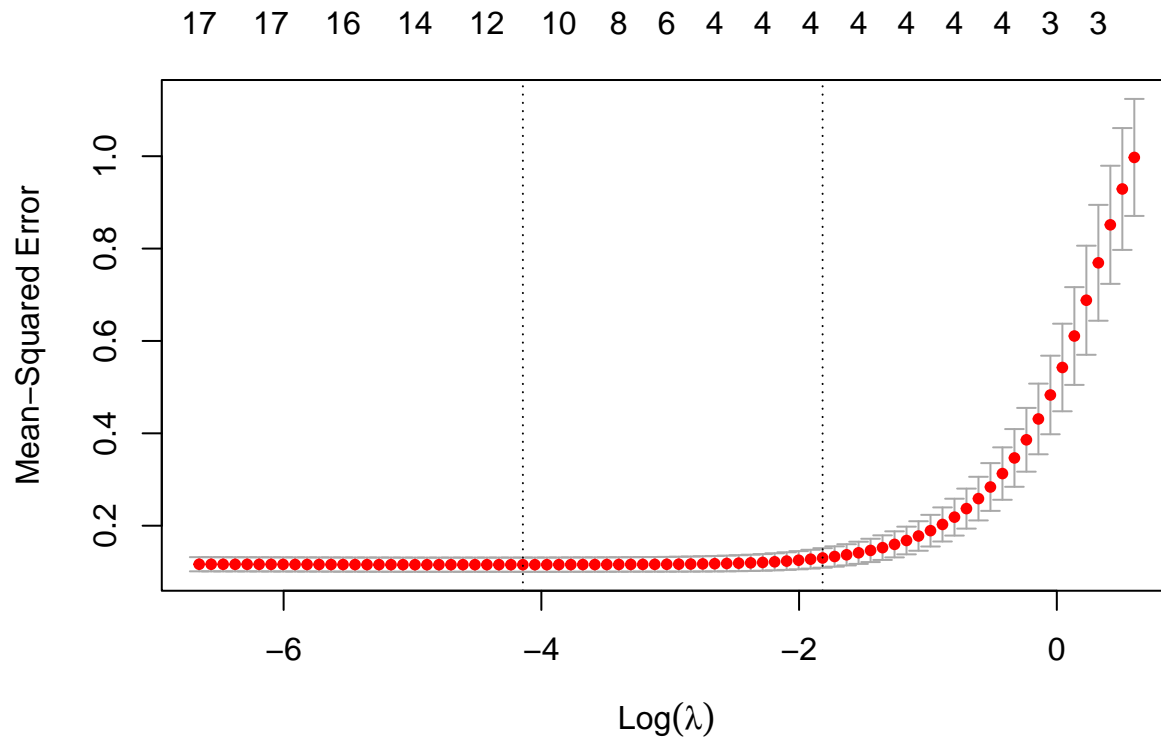
```
# DEVELOPING THE MODEL (LR. ENR)
```

```
x <- model.matrix(state_mentalhealth_util ~ ., data = GTrend_training_set_f, intercept = FALSE)
y <- GTrend_training_set_f$state_mentalhealth_util

#Performing cross_validation to find the best lambda
set.seed(123) # for consistent and replicable results

cv_model <- cv.glmnet(x, y, alpha = 0.5, family = "gaussian", nfolds = 5)

plot(cv_model) #plotting cross-validation curve
```



```
#getting the best/ optimal lambda

best_lambda <- cv_model$lambda.min
best_lambda_1se <- cv_model$lambda.1se

#developing the model using the best lambda

model_min <- glmnet(x, y, alpha = 0.5, lambda = best_lambda, family = "gaussian")
model_lambda_1se <- glmnet(x, y, alpha = 0.5, lambda = best_lambda_1se, family = "gaussian")

#preparing the test set into matrix

x_test <- model.matrix(state_mentalhealth_util ~ ., data = test_set_f, intercept = FALSE)
y_test <- test_set_f$state_mentalhealth_util

#ensure x and x_test have the same number of columns. its a good practise after using model.matrix
```

```

common_columns <- intersect(colnames(x), colnames(x_test))
x <- x[, common_columns]
x_test <- x_test[, common_columns]

# use test set to make predictions, use lambda min and lambda_1se

y_pred_min <- predict(model_min, newx = x_test)
y_pred_1se <- predict(model_lambda_1se, newx = x_test)

#calculate the mean squared error

mse_min <- mean((y_test - y_pred_min)^2)
mse_1se <- mean((y_test - y_pred_1se)^2)

print(paste("MSE (MIN):", mse_min))

```

```
## [1] "MSE (MIN): 0.118617218318104"
```

```
print(paste("MSE (1SE):", mse_1se))
```

```
## [1] "MSE (1SE): 0.184072699215542"
```

Principal Component Regression (PCR)

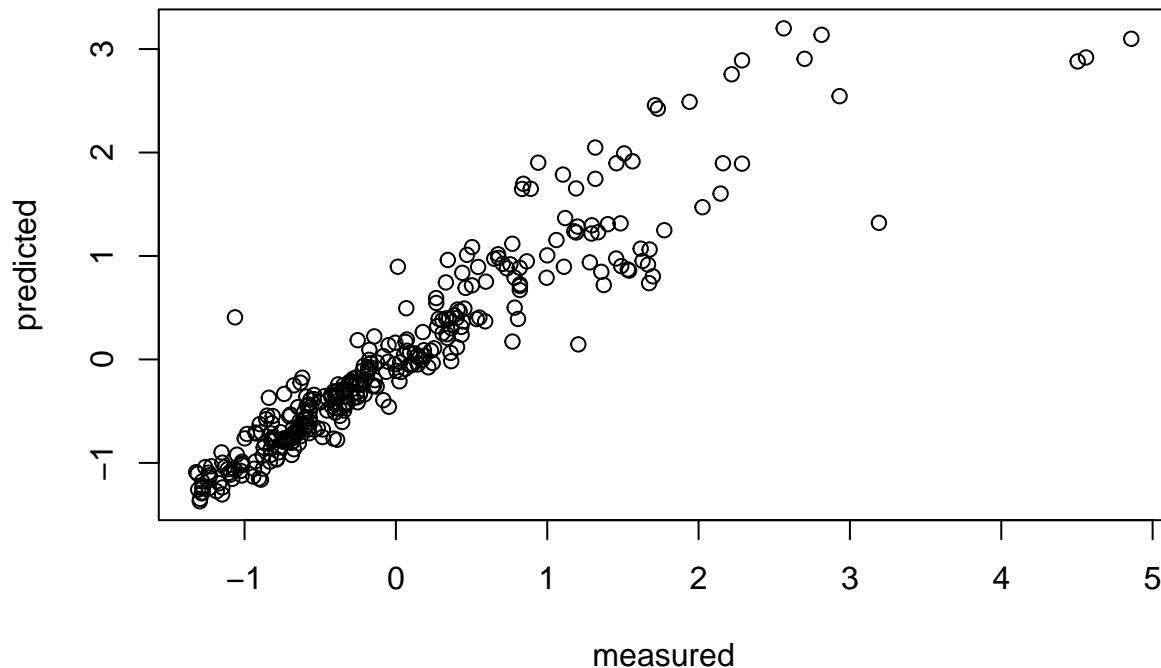
```

pcr_m_selected <- 1

# Get the PCR fit for the training data set
pcr_fit <- pcr(state_mentalhealth_util ~ ., data = GTrend_training_set_f ,
               scale=TRUE, validation="CV")
# plot the PCR fit
plot(pcr_fit)

```

state_mentalhealth_util, 17 comps, validation

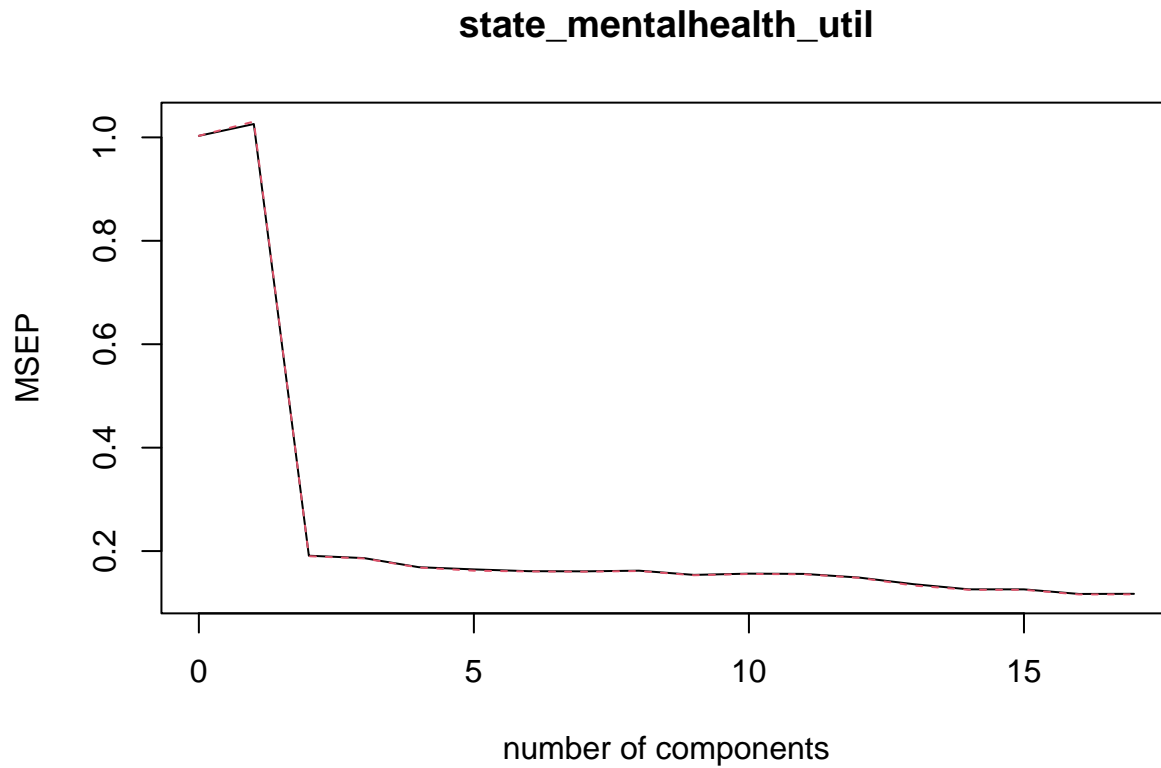


```
# Show the summary of the PCR fit.
summary(pcr_fit)
```

```
## Data:      X dimension: 336 17
## Y dimension: 336 1
## Fit method: svdpc
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##      (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           1.001    1.013    0.4371    0.4318    0.4109    0.4057    0.4014
## adjCV         1.001    1.015    0.4359    0.4310    0.4100    0.4025    0.4007
##      7 comps  8 comps  9 comps 10 comps 11 comps 12 comps 13 comps
## CV      0.4010  0.4026  0.3924    0.3955    0.3948    0.3857    0.3687
## adjCV    0.4002  0.4018  0.3915    0.3946    0.3939    0.3851    0.3651
##      14 comps 15 comps 16 comps 17 comps
## CV      0.3550  0.3549  0.3422    0.3425
## adjCV    0.3537  0.3537  0.3408    0.3411
##
## TRAINING: % variance explained
##      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X           22.8132  42.84   55.93   65.93   73.62   81.08
## state_mentalhealth_util 0.4181  81.43  82.08  83.95  84.70  84.73
##      7 comps  8 comps  9 comps 10 comps 11 comps
## X           85.04  88.59  91.16  93.13  94.94
## state_mentalhealth_util 84.93  84.93  85.75  85.79  86.02
##      12 comps 13 comps 14 comps 15 comps 16 comps
## X           96.33  97.4   98.39  99.10  99.66
```

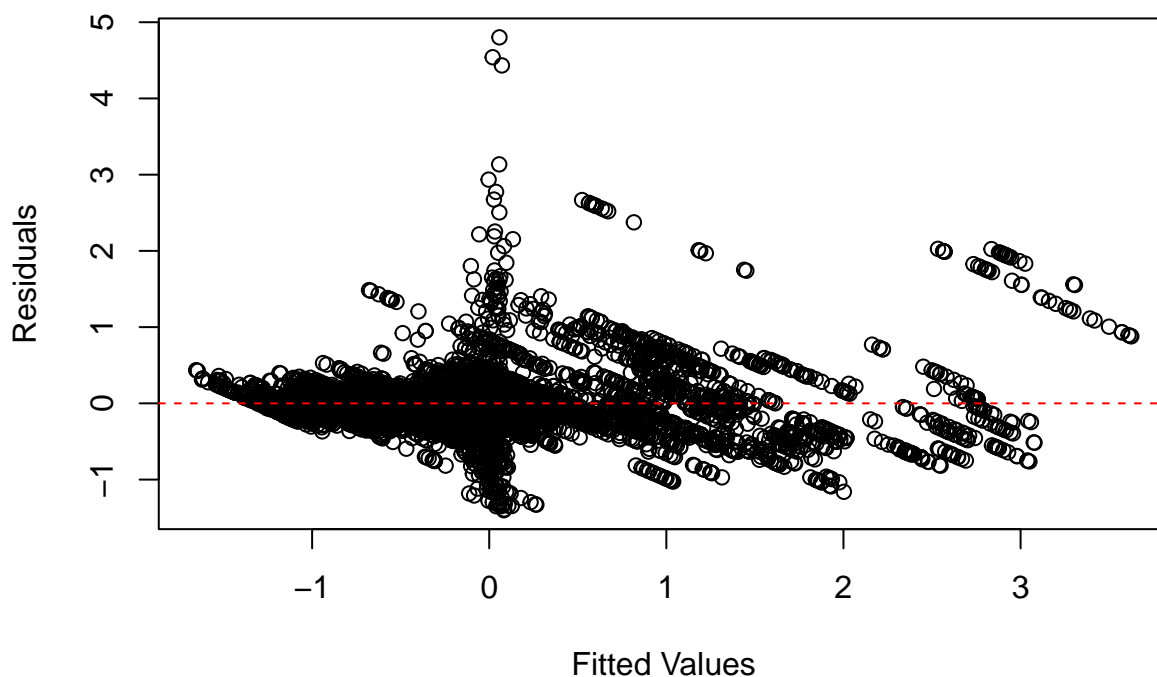
```
## state_mentalhealth_util      86.89      88.5      88.93      89.08      90.04
##                               17 comps
## X                             100.00
## state_mentalhealth_util      90.05
```

```
# Show the validation plot.
validationplot(pcr_fit, val.type="MSEP")
```



```
# Plot the residuals vs the fitted values.
pcr_fitted_vals <- as.vector(fitted(pcr_fit, ncomp=5))
pcr_residuals <- as.vector(residuals(pcr_fit, ncomp=5))
plot(pcr_fitted_vals, pcr_residuals,
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "PCR: Residuals vs Fitted")
abline(h = 0, col = "red", lty = 2)
```


PCR: Residuals vs Fitted



```
# Get the predictions
pcr_preds_train <- predict(pcr_fit, data=GTrend_training_set_f, ncomp=pcr_m_selected)
pcr_preds_test  <- predict(pcr_fit, data=test_set, ncomp=pcr_m_selected)

# Store and print the pcr mean square error for M_selected.
pcr_train_mse <- mean((pcr_preds_train-GTrend_training_set_f$state_mentalhealth_util)^2)
pcr_test_mse  <- mean((pcr_preds_test-test_set$state_mentalhealth_util)^2)

#add the test and train RMSEs to the mse_df
mse_df <- add_rmse_row(mse_df, "Principal Component Regression", pcr_train_mse, pcr_test_mse)

paste("PCR Train MSE for M Selected:",pcr_m_selected,"is", pcr_train_mse)
```

```
## [1] "PCR Train MSE for M Selected: 1 is 0.992854810781946"
```

```
paste("PCR Test MSE for M Selected:",pcr_m_selected,"is", pcr_test_mse)
```

```
## [1] "PCR Test MSE for M Selected: 1 is 0.0078702750045775"
```

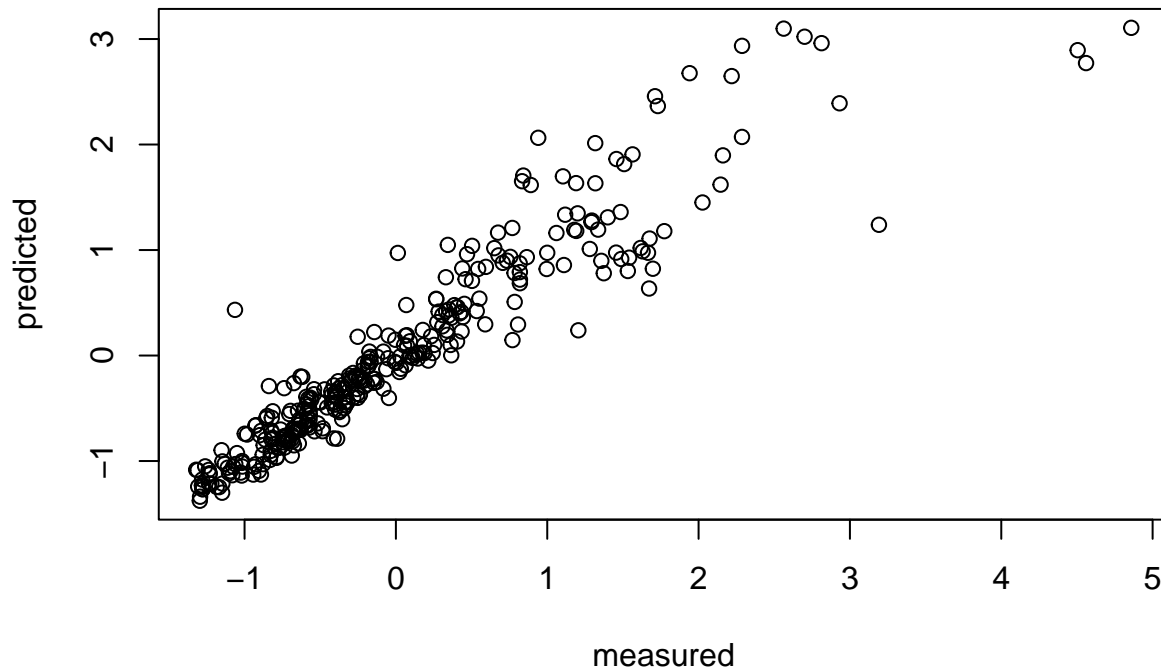
Partial Least Squares Regression (PLSR)

```
# Set the PLS M selected value.
plsr_M_selected <- 15

# Get the PCR fit for the training data set
plsr_fit <- plsr(state_mentalhealth_util ~ ., data=GTrend_training_set_f ,
                 scale=TRUE, validation="CV", ncomp=plsr_M_selected)
```

```
# Plot the PLSR fit
plot(plsr_fit)
```

state_mentalhealth_util, 15 comps, validation

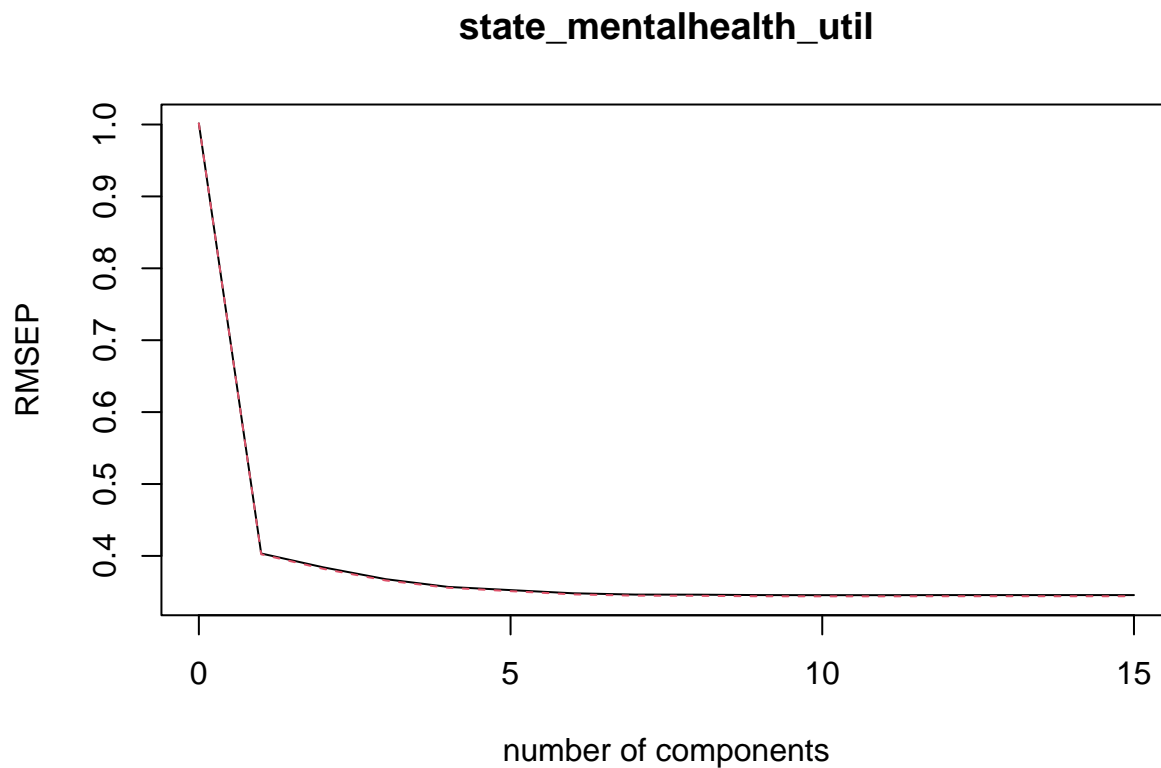


```
# print the summary of the partial least square regression fit.
summary(plsr_fit)
```

```
## Data:      X dimension: 336 17
## Y dimension: 336 1
## Fit method: kernelppls
## Number of components considered: 15
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##      (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           1.001  0.4034  0.3840  0.3676  0.3569  0.3524  0.3480
## adjCV        1.001  0.4023  0.3819  0.3658  0.3557  0.3510  0.3465
##      7 comps  8 comps  9 comps 10 comps 11 comps 12 comps 13 comps
## CV      0.3462  0.3461  0.3455  0.3454  0.3454  0.3455  0.3455
## adjCV    0.3447  0.3445  0.3439  0.3438  0.3438  0.3439  0.3439
##      14 comps 15 comps
## CV      0.3455  0.3455
## adjCV    0.3439  0.3439
##
## TRAINING: % variance explained
##              1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X              19.94  29.75  42.73  55.11  64.89  71.27
## state_mentalhealth_util 84.60  86.78  88.13  89.19  89.58  89.84
##              7 comps  8 comps  9 comps 10 comps 11 comps
```

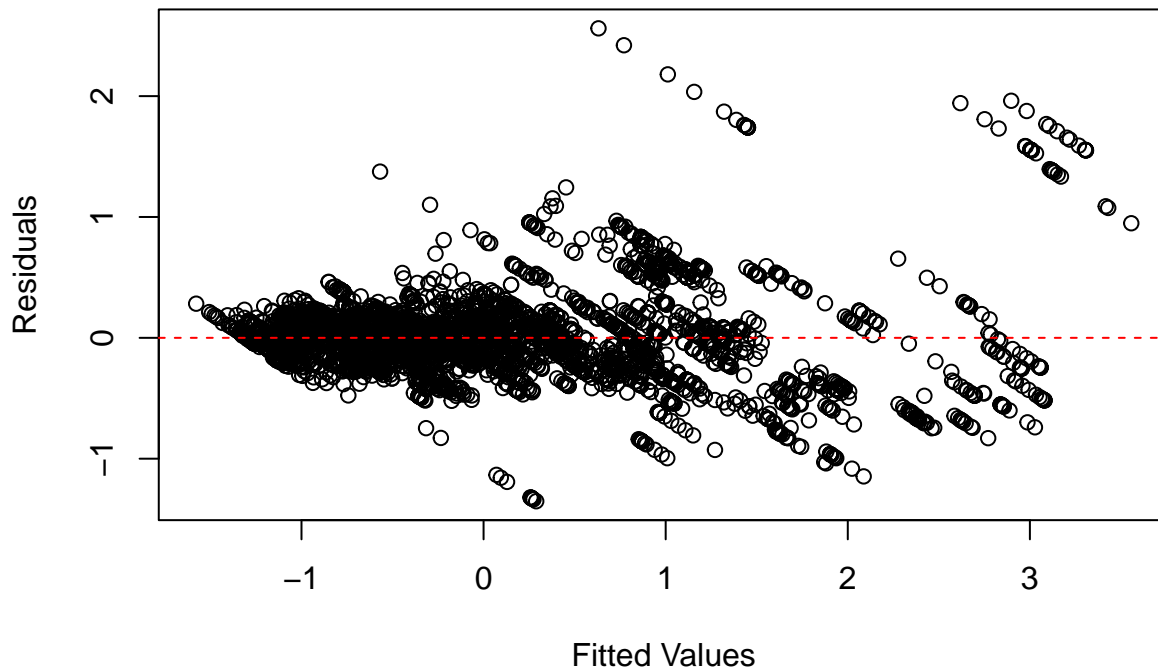
```
## X          77.23    80.15    82.18    88.21    90.85
## state_mentalhealth_util 89.96    90.02    90.05    90.05    90.05
##          12 comps  13 comps  14 comps  15 comps
## X          92.30    94.34    96.41    97.59
## state_mentalhealth_util 90.05    90.05    90.05    90.05
```

```
# Show the validation plot
validationplot(plsr_fit)
```



```
# Plot the residuals vs the fitted values.
plsr_fitted_vals <- as.vector(fitted(plsr_fit, ncomp=5))
plsr_residuals <- as.vector(residuals(plsr_fit, ncomp=5))
plot(plsr_fitted_vals, plsr_residuals,
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "PLSR: Residuals vs Fitted")
abline(h = 0, col = "red", lty = 2)
```

PLSR: Residuals vs Fitted



```
# Get the predictions
plsr_train_preds <- predict(plsr_fit, data=GTrend_training_set_f, ncomp=plsr_M_selected)
plsr_test_preds <- predict(plsr_fit, data=test_set_f, ncomp=plsr_M_selected)

# Store and print the MSE value for the PLSR
plsr_train_mse <- mean((plsr_train_preds-GTrend_training_set_f$state_mentalhealth_util)^2)
plsr_test_mse <- mean((plsr_test_preds-test_set_f$state_mentalhealth_util)^2)

#add the test and train RMSEs to the mse_df
mse_df <- add_rmse_row(mse_df, "Partial Least Squares Regression", plsr_train_mse, plsr_test_mse)

paste("PLSR Train MSE for M Selected:",plsr_M_selected,"is", plsr_train_mse)
```

```
## [1] "PLSR Train MSE for M Selected: 15 is 0.099169665276029"
```

```
paste("PLSR Test MSE for M Selected:",plsr_M_selected,"is", plsr_test_mse)
```

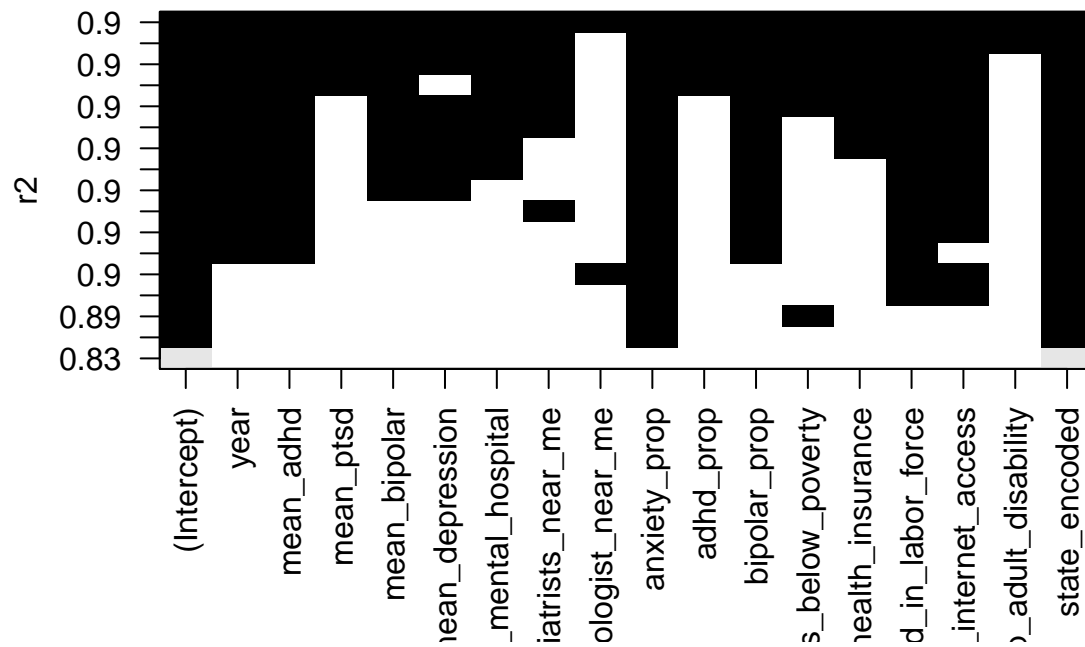
```
## [1] "PLSR Test MSE for M Selected: 15 is 2.35645183671798"
```

Best Subset Selection

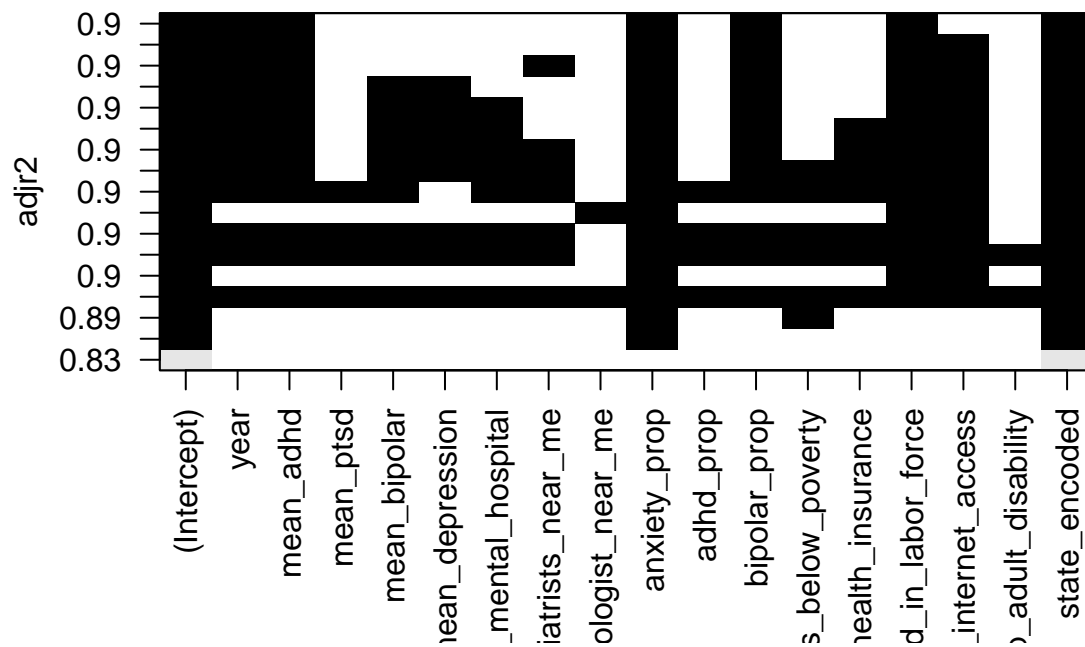
```
# Load library needed for regsubsets() function
library(leaps)

# The regsubsets() function (part of the leaps library) performs best sub- set selection
# by identifying the best model that contains a given number of predictors, where best
# is quantified using RSS.
```

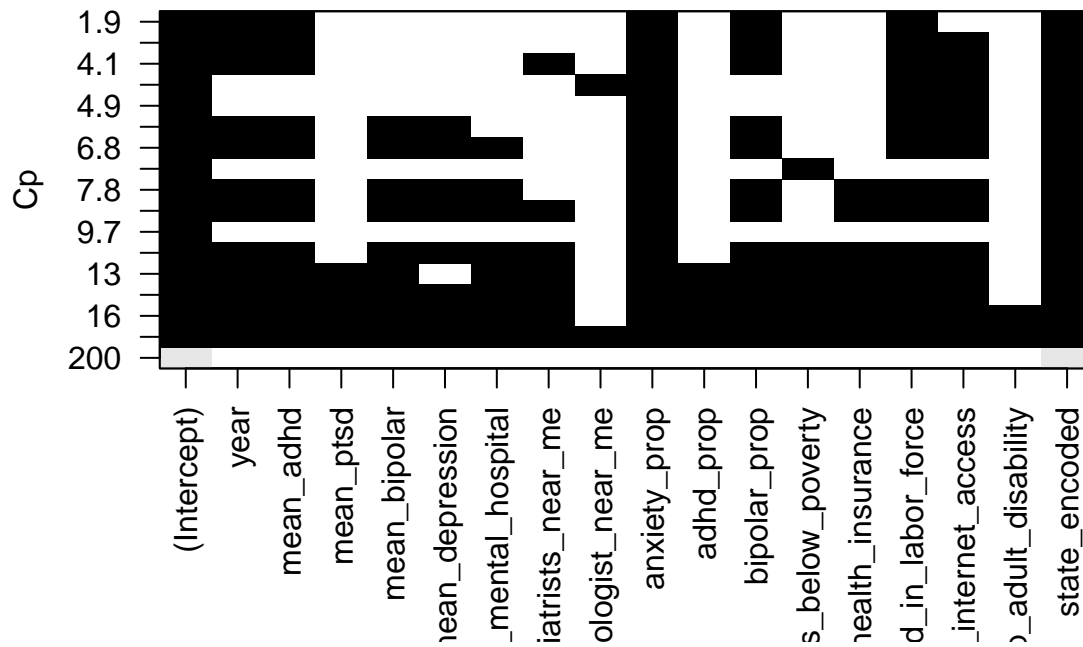
```
reg_fit_train <- regsubsets(state_mentalhealth_util ~ ., data=GTrend_training_set_f, nvmax=23)
plot(reg_fit_train, scale="r2")
```



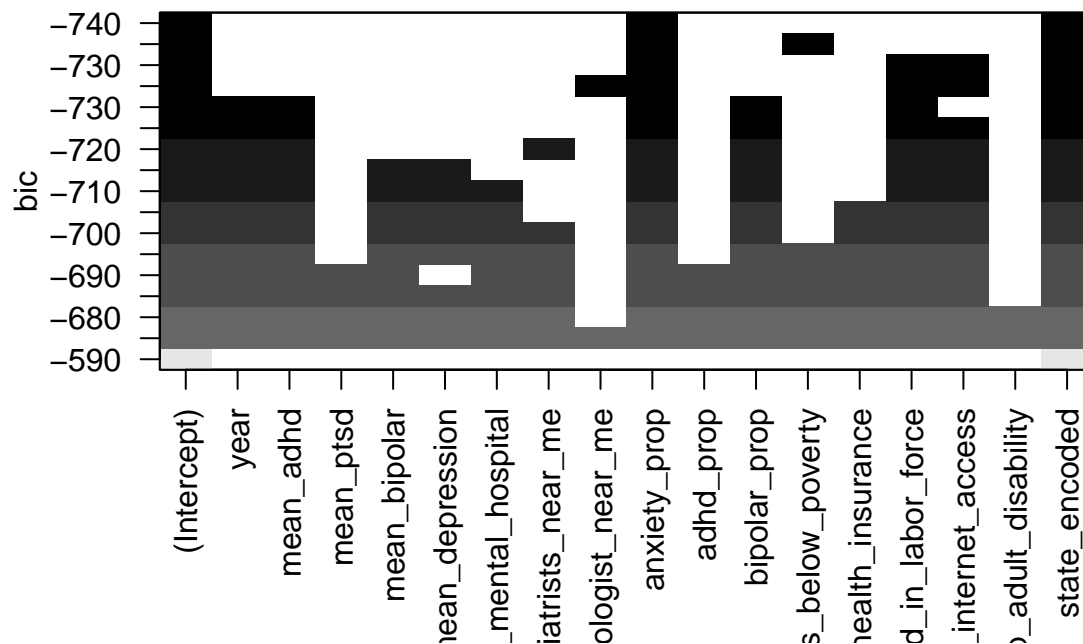
```
plot(reg_fit_train, scale="adjr2")
```



```
plot(reg_fit_train, scale="Cp")
```



```
plot(reg_fit_train, scale="bic")
```



The summary() command outputs the best set of variables for each model size.

```
reg.summary <- summary(reg_fit_train)
print(reg.summary)
```

```
## Subset selection object
## Call: regsubsets.formula(state_mentalhealth_util ~ ., data = GTrend_training_set_f,
##       nvmax = 23)
## 17 Variables (and intercept)
##
```

Forced in Forced out

```

## year                                FALSE      FALSE
## mean_adhd                           FALSE      FALSE
## mean_ptsd                           FALSE      FALSE
## mean_bipolar                         FALSE      FALSE
## mean_depression                     FALSE      FALSE
## mean_mental_hospital                 FALSE      FALSE
## mean_psychiatrists_near_me           FALSE      FALSE
## mean_psychologist_near_me            FALSE      FALSE
## anxiety_prop                         FALSE      FALSE
## adhd_prop                            FALSE      FALSE
## bipolar_prop                         FALSE      FALSE
## prop_families_below_poverty          FALSE      FALSE
## prop_adults_without_health_insurance FALSE      FALSE
## prop_unemployed_in_labor_force        FALSE      FALSE
## prop_without_internet_access          FALSE      FALSE
## prop_adult_disability                 FALSE      FALSE
## state_encoded                        FALSE      FALSE
## 1 subsets of each size up to 17
## Selection Algorithm: exhaustive
##      year mean_adhd mean_ptsd mean_bipolar mean_depression
## 1 ( 1 ) " " " " " " " "
## 2 ( 1 ) " " " " " " " "
## 3 ( 1 ) " " " " " " " "
## 4 ( 1 ) " " " " " " " "
## 5 ( 1 ) " " " " " " " "
## 6 ( 1 ) "*" "*" " " " " " "
## 7 ( 1 ) "*" "*" " " " " " "
## 8 ( 1 ) "*" "*" " " " " " "
## 9 ( 1 ) "*" "*" " " "*" "*"
## 10 ( 1 ) "*" "*" " " "*" "*"
## 11 ( 1 ) "*" "*" " " "*" "*"
## 12 ( 1 ) "*" "*" " " "*" "*"
## 13 ( 1 ) "*" "*" " " "*" "*"
## 14 ( 1 ) "*" "*" "*" "*" " "
## 15 ( 1 ) "*" "*" "*" "*" "*"
## 16 ( 1 ) "*" "*" "*" "*" "*"
## 17 ( 1 ) "*" "*" "*" "*" "*"
##      mean_mental_hospital mean_psychiatrists_near_me
## 1 ( 1 ) " " " "
## 2 ( 1 ) " " " "
## 3 ( 1 ) " " " "
## 4 ( 1 ) " " " "
## 5 ( 1 ) " " " "
## 6 ( 1 ) " " " "
## 7 ( 1 ) " " " "
## 8 ( 1 ) " " "*"
## 9 ( 1 ) " " " "
## 10 ( 1 ) "*" " "
## 11 ( 1 ) "*" " "
## 12 ( 1 ) "*" "*"
## 13 ( 1 ) "*" "*"
## 14 ( 1 ) "*" "*"
## 15 ( 1 ) "*" "*"
## 16 ( 1 ) "*" "*"

```

```

## 17 ( 1 ) "*" "*"
##      mean_psychologist_near_me anxiety_prop adhd_prop bipolar_prop
## 1 ( 1 ) " " " " " "
## 2 ( 1 ) " " "*" " " " "
## 3 ( 1 ) " " "*" " " " "
## 4 ( 1 ) " " "*" " " " "
## 5 ( 1 ) "*" "*" " " " "
## 6 ( 1 ) " " "*" " " "*"
## 7 ( 1 ) " " "*" " " "*"
## 8 ( 1 ) " " "*" " " "*"
## 9 ( 1 ) " " "*" " " "*"
## 10 ( 1 ) " " "*" " " "*"
## 11 ( 1 ) " " "*" " " "*"
## 12 ( 1 ) " " "*" " " "*"
## 13 ( 1 ) " " "*" " " "*"
## 14 ( 1 ) " " "*" "*" "*"
## 15 ( 1 ) " " "*" "*" "*"
## 16 ( 1 ) " " "*" "*" "*"
## 17 ( 1 ) "*" "*" "*" "*"
##      prop_families_below_poverty prop_adults_without_health_insurance
## 1 ( 1 ) " " " "
## 2 ( 1 ) " " " "
## 3 ( 1 ) "*" " "
## 4 ( 1 ) " " " "
## 5 ( 1 ) " " " "
## 6 ( 1 ) " " " "
## 7 ( 1 ) " " " "
## 8 ( 1 ) " " " "
## 9 ( 1 ) " " " "
## 10 ( 1 ) " " " "
## 11 ( 1 ) " " "*"
## 12 ( 1 ) " " "*"
## 13 ( 1 ) "*" "*"
## 14 ( 1 ) "*" "*"
## 15 ( 1 ) "*" "*"
## 16 ( 1 ) "*" "*"
## 17 ( 1 ) "*" "*"
##      prop_unemployed_in_labor_force prop_without_internet_access
## 1 ( 1 ) " " " "
## 2 ( 1 ) " " " "
## 3 ( 1 ) " " " "
## 4 ( 1 ) "*" "*"
## 5 ( 1 ) "*" "*"
## 6 ( 1 ) "*" " "
## 7 ( 1 ) "*" "*"
## 8 ( 1 ) "*" "*"
## 9 ( 1 ) "*" "*"
## 10 ( 1 ) "*" "*"
## 11 ( 1 ) "*" "*"
## 12 ( 1 ) "*" "*"
## 13 ( 1 ) "*" "*"
## 14 ( 1 ) "*" "*"
## 15 ( 1 ) "*" "*"
## 16 ( 1 ) "*" "*"

```



```
## 17 ( 1 ) "*" "*"
##      prop_adult_disability state_encoded
## 1 ( 1 ) " " "*"
## 2 ( 1 ) " " "*"
## 3 ( 1 ) " " "*"
## 4 ( 1 ) " " "*"
## 5 ( 1 ) " " "*"
## 6 ( 1 ) " " "*"
## 7 ( 1 ) " " "*"
## 8 ( 1 ) " " "*"
## 9 ( 1 ) " " "*"
## 10 ( 1 ) " " "*"
## 11 ( 1 ) " " "*"
## 12 ( 1 ) " " "*"
## 13 ( 1 ) " " "*"
## 14 ( 1 ) " " "*"
## 15 ( 1 ) " " "*"
## 16 ( 1 ) "*" "*"
## 17 ( 1 ) "*" "*"

```

```
names(reg.summary)
```

```
## [1] "which" "rsq" "rss" "adjr2" "cp" "bic" "outmat" "obj"
```

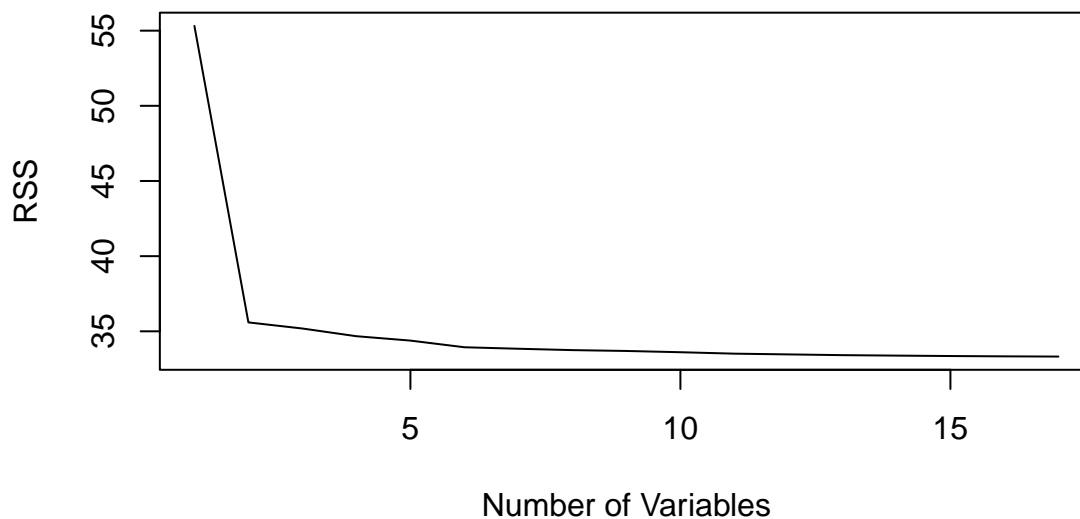
```
#Print the R2 statistic
```

```
reg.summary$rsq
```

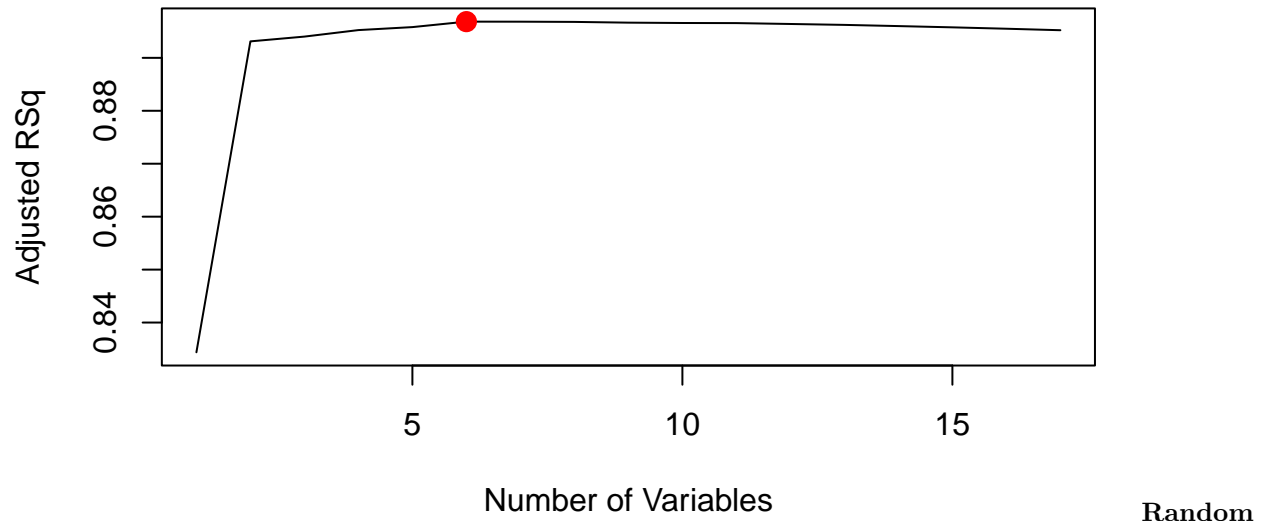
```
## [1] 0.8348754 0.8937489 0.8949626 0.8964947 0.8973628 0.8986859 0.8989819
## [8] 0.8992507 0.8994236 0.8996708 0.8999579 0.9001067 0.9002513 0.9003445
## [15] 0.9004271 0.9004911 0.9005343
```

```
#par(mfrow=c(1,2))
```

```
plot(reg.summary$rss, xlab="Number of Variables", ylab="RSS", type="l")
```



```
plot(reg.summary$adjr2 , xlab = "Number of Variables",ylab = "Adjusted RSq", type = "l")
# which.max(reg.summary$adjr2)
plot(reg.summary$adjr2 , xlab = "Number of Variables", ylab = "Adjusted RSq", type = "l")
points(which.max(reg.summary$adjr2), reg.summary$adjr2[which.max(reg.summary$adjr2)],
       col = "red", cex = 2, pch = 20)
```



Forest

```
library(randomForest)
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
```

```
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
```

```
##
```

```
## combine
```

```
## The following object is masked from 'package:ggplot2':
```

```
##
```

```
## margin
```

```
set.seed(42)
```

```
# Bagging
```

```
bag.data <- randomForest(state_mentalhealth_util ~ ., data=GTrend_training_set_f, mtry=24, importance=T)
bag.data
```

```
##
```

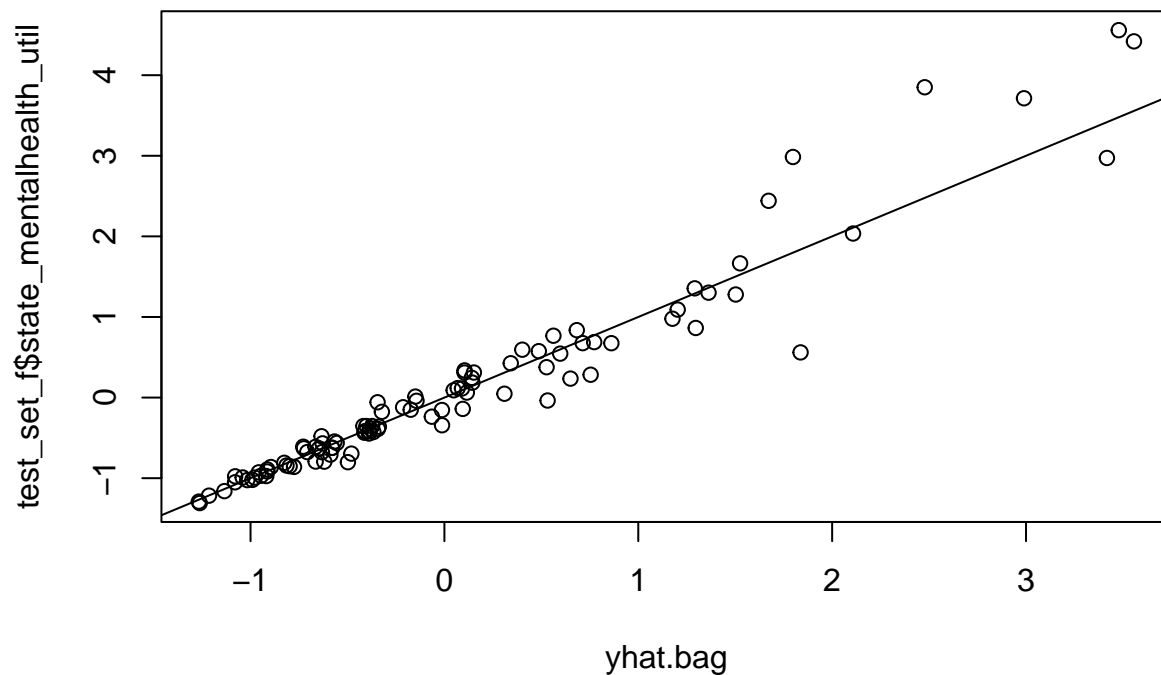
```
## Call:
```

```
## randomForest(formula = state_mentalhealth_util ~ ., data = GTrend_training_set_f, mtry = 24, importance = T)
```

```
##           Type of random forest: regression
##           Number of trees: 500
## No. of variables tried at each split: 17
##
##           Mean of squared residuals: 0.1045568
##           % Var explained: 89.51
```

```
yhat.bag <- predict(bag.data, newdata=test_set_f)

plot(yhat.bag, test_set_f$state_mentalhealth_util)
abline(0,1)
```



```
bagged_mse <- mean((yhat.bag - test_set_f$state_mentalhealth_util)^2)
paste ("Test MSE associated with the bagged regression is:", bagged_mse)
```

```
## [1] "Test MSE associated with the bagged regression is: 0.105812003090838"
```

```
# Random Forest
rf_model <- randomForest(state_mentalhealth_util ~ ., data=GTrend_training_set_f ,
                          mtry = 12, importance = TRUE)
print(rf_model)
```

```
##
## Call:
## randomForest(formula = state_mentalhealth_util ~ ., data = GTrend_training_set_f,      mtry = 12, i
##           Type of random forest: regression
##           Number of trees: 500
## No. of variables tried at each split: 12
##
##           Mean of squared residuals: 0.0959313
##           % Var explained: 90.38
```

```

yhat_train_rf <- predict(rf_model, newdata = GTrend_training_set_f)
yhat_test_rf <- predict(rf_model, newdata = test_set_f)
rf_train_mse <- mean((yhat_train_rf-test_set_f$state_mentalhealth_util)^2)
rf_test_mse <- mean((yhat_test_rf-test_set_f$state_mentalhealth_util)^2)

#add the test and train RMSEs to the mse_df
mse_df <- add_rmse_row(mse_df, "Random Forest", rf_train_mse, rf_test_mse)

paste("Train MSE associated with the Random Forest is: =", rf_train_mse)

```

```
## [1] "Train MSE associated with the Random Forest is: = 2.36039652693596"
```

```
paste("Test MSE associated with the Random Forest is: =", rf_test_mse)
```

```
## [1] "Test MSE associated with the Random Forest is: = 0.11296086794388"
```

```

imp <- importance(rf_model)
# Let's sort the output of the importance() function
imp_df <- data.frame(Variable = rownames(imp), imp)
imp_sorted <- imp_df[order(-imp_df$X.IncMSE), ]
head(imp_sorted)

```

```
##
```

	Variable	X.IncMSE	IncNodePurity
## state_encoded	state_encoded	54.769275	198.198052
## anxiety_prop	anxiety_prop	20.448293	61.582477
## adhd_prop	adhd_prop	16.885457	34.844884
## bipolar_prop	bipolar_prop	14.112888	9.646262
## prop_families_below_poverty	prop_families_below_poverty	8.611730	2.093496
## prop_adult_disability	prop_adult_disability	8.318198	3.368368

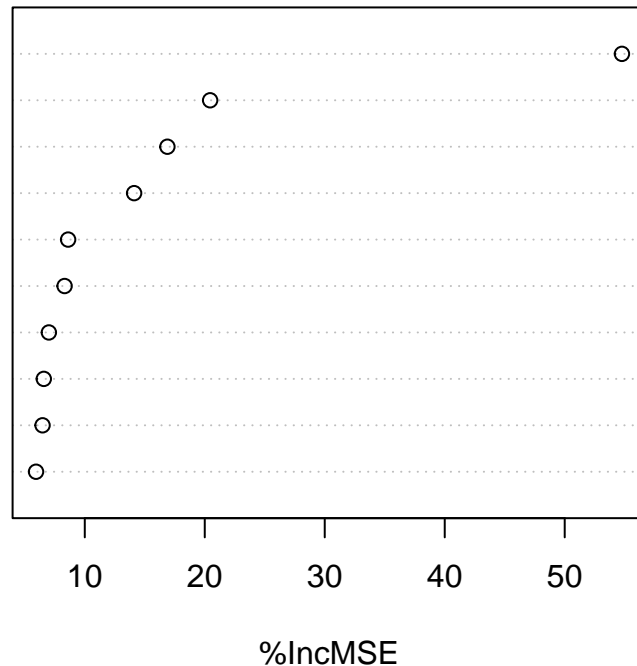
```

# Show the importance plot
#varImpPlot(rf_model)
varImpPlot(
  x = rf_model,      # trained random forest
  sort = TRUE,       # sort by importance
  n.var = 10,        # show top 10 variables
  type = 1,          # mean decrease in accuracy
  main = "Top 10 Important Variables"
)

```

Top 10 Important Variables

state_encoded
 anxiety_prop
 adhd_prop
 bipolar_prop
 prop_families_below_poverty
 prop_adult_disability
 mean_mental_hospital
 year
 mean_ptsd
 mean_psychologist_near_me



```
set.seed(42)

# Set up a 5 fold cross-validation for the random forest model.
rf_control <- trainControl(method="cv", number=5)

# Define the tuning grid with values for mtry at 8, 10, 12, or 14.
tune_grid <- expand.grid(.mtry = c(6, 8, 10, 12, 14, 16, 18))

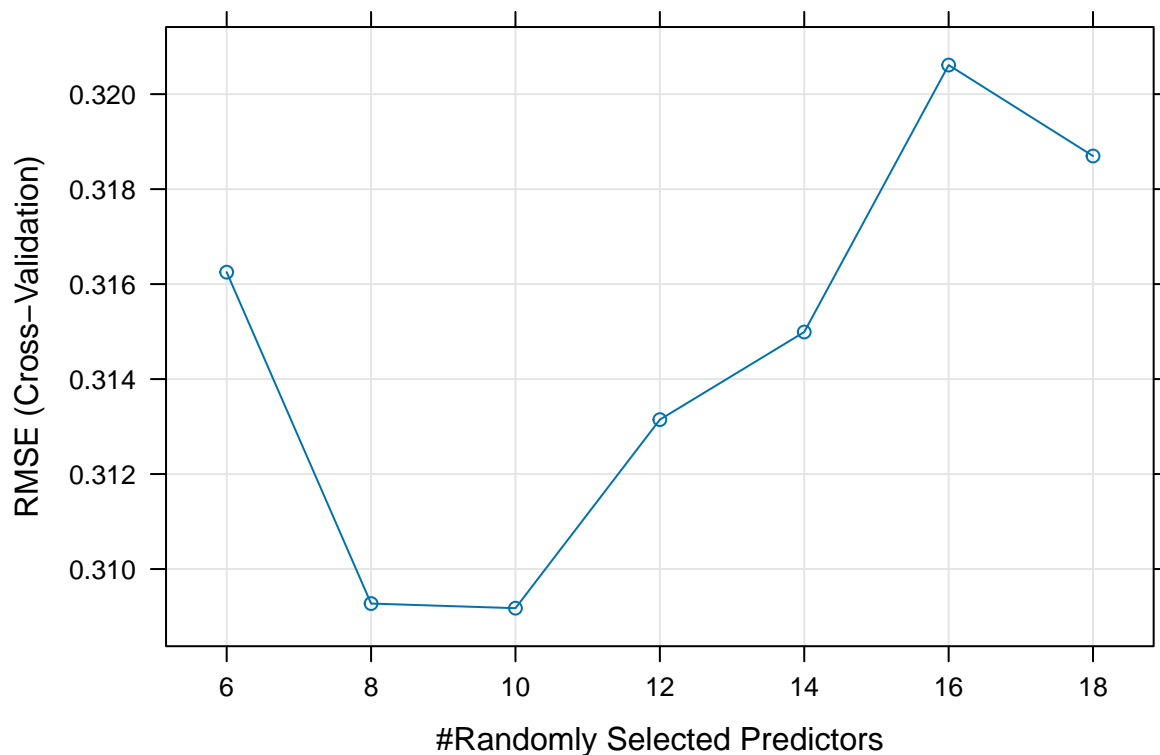
# Train the random forest model using k-fold cross validation
rf_cv_model <- train(state_mentalhealth_util ~ .,
  data = GTrend_training_set_f,
  method = "rf",
  trControl = rf_control,
  tuneGrid = tune_grid,
  importance = TRUE)

# Print the results
print(rf_cv_model)
```

```
## Random Forest
##
## 336 samples
## 17 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 268, 269, 269, 269, 269
## Resampling results across tuning parameters:
##
##  mtry  RMSE      Rsquared  MAE
```

```
##      6      0.3162512  0.9084503  0.1684284
##      8      0.3092728  0.9113747  0.1614190
##     10      0.3091755  0.9117237  0.1593729
##     12      0.3131462  0.9092811  0.1613469
##     14      0.3149912  0.9092144  0.1622431
##     16      0.3206130  0.9059745  0.1659186
##     18      0.3186975  0.9071473  0.1665847
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 10.
```

```
# Show validation plot
plot(rf_cv_model)
```

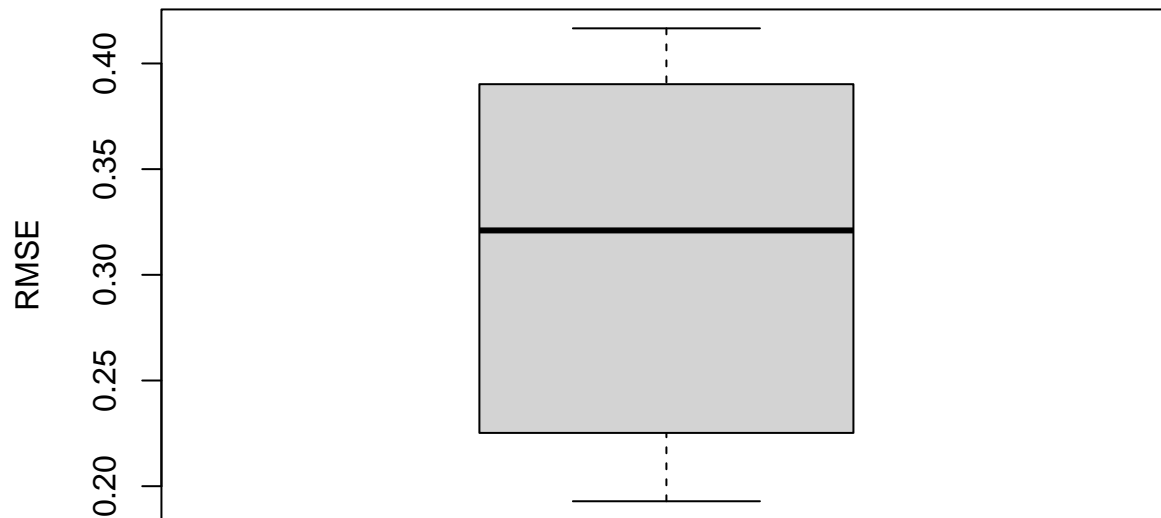


```
names(rf_cv_model)
```

```
## [1] "method"      "modelInfo"   "modelType"   "results"     "pred"
## [6] "bestTune"    "call"        "dots"        "metric"      "control"
## [11] "finalModel"  "preProcess"  "trainingData" "ptype"       "resample"
## [16] "resampledCM" "perfNames"   "maximize"    "yLimits"     "times"
## [21] "levels"     "terms"       "coefnames"   "xlevels"
```

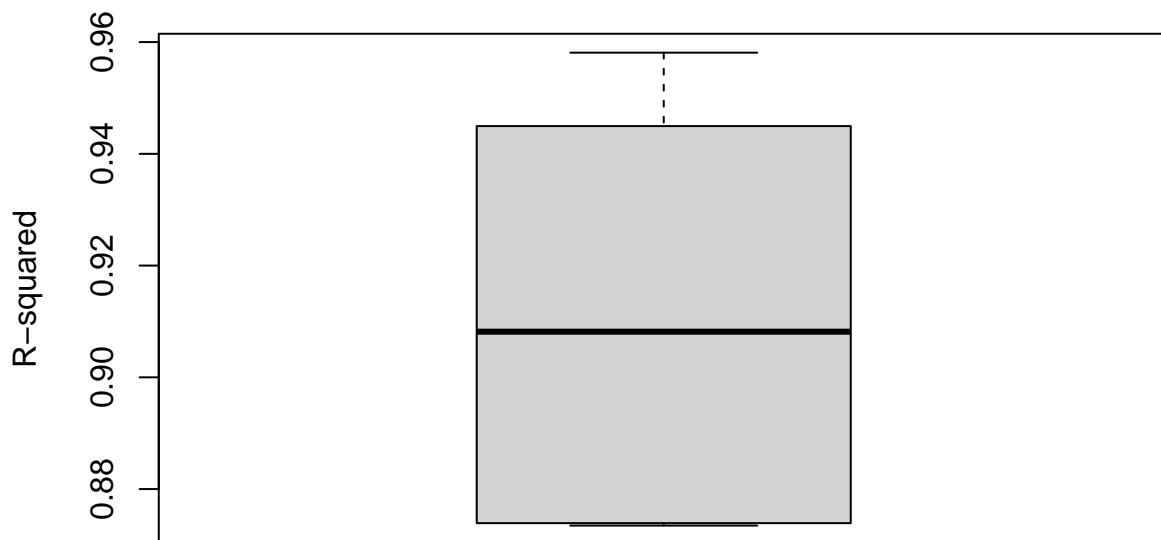
```
# Show RMSE across folds while using na.omit to remove null values before plotting
boxplot(na.omit(rf_cv_model$resample$RMSE),
        main = "Validation RMSE Across Folds",
        ylab = "RMSE")
```

Validation RMSE Across Folds



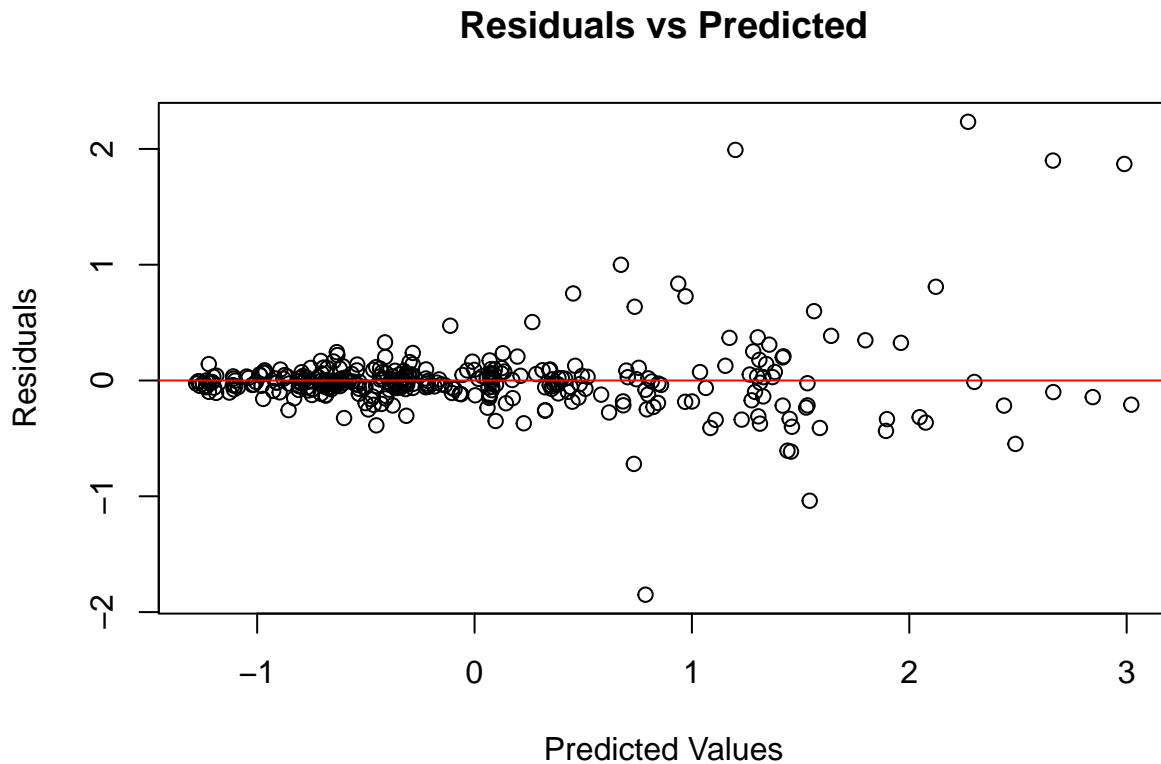
```
# Show R squared across folds with na.omit() as above.  
boxplot(na.omit(rf_cv_model$resample$Rsquared),  
        main = "Validation R-squared Across Folds",  
        ylab = "R-squared")
```

Validation R-squared Across Folds



```
# Show the residuals plot  
# Residuals  
residuals <- rf_cv_model$finalModel$y - rf_cv_model$finalModel$predicted  
  
# Plot residuals vs fitted  
plot(rf_cv_model$finalModel$predicted, residuals,  
     xlab = "Predicted Values",  
     ylab = "Residuals",
```

```
main = "Residuals vs Predicted")
abline(h = 0, col = "red")
```



```
mse_df
```

```
## # A tibble: 3 x 3
##   Model                                Train_MSE Test_MSE
##   <chr>                                <dbl>    <dbl>
## 1 Principal Component Regression      0.993  0.00787
## 2 Partial Least Squares Regression    0.0992  2.36
## 3 Random Forest                      2.36    0.113
```

```
names(GTrend_training_set_f)
```

```
## [1] "year"
## [2] "mean_adhd"
## [3] "mean_ptsd"
## [4] "mean_bipolar"
## [5] "mean_depression"
## [6] "mean_mental_hospital"
## [7] "mean_psychiatrists_near_me"
## [8] "mean_psychologist_near_me"
## [9] "state_mentalhealth_util"
## [10] "anxiety_prop"
## [11] "adhd_prop"
## [12] "bipolar_prop"
## [13] "prop_families_below_poverty"
```



```
## [14] "prop_adults_without_health_insurance"  
## [15] "prop_unemployed_in_labor_force"  
## [16] "prop_without_internet_access"  
## [17] "prop_adult_disability"  
## [18] "state_encoded"
```