573 Research Project

Troy Hall

4/6/2022

USER NOTE

Be sure to set all appropriate filepaths in the chunk below BEFORE running this code. Save all dependencies including the 2019 AESC CPS .dat file and .xml file, county data files, “IPUMS health var table” and “1990 Census to NAICS” to working directory BEFORE running this code.

If you focus on the regression prediction problem, please report results for the following methods: linear regression, LASSO/ridge/elastic net, regression trees and random forests, boosting, SVM, kNN.

If you focus on the classification problem, please report results for the following meth- ods: logistic regression, LDA/QDA, classification trees and random forests, boosting, SVM, and kNN.

```
workingd = c("C:/Users/troyhall/Documents/Classes/Spring '22/ECON 573/Final
Project/")
setwd(workingd)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v readr   2.0.1

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(ipumsr)
library(readxl)
library(glmnet)

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-2

library(ggplot2)
library(pls)

## Warning: package 'pls' was built under R version 4.1.2

##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings

library(class)
library(corrplot)

## corrplot 0.91 loaded

##
## Attaching package: 'corrplot'

## The following object is masked from 'package:pls':
##
##     corrplot

library(e1071)
library(leaps)

## Warning: package 'leaps' was built under R version 4.1.2

library(tree)

## Warning: package 'tree' was built under R version 4.1.3

## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

```r
library(qwraps2)
```

```
## Warning: package 'qwraps2' was built under R version 4.1.3
```

```r
library(broom)
```

Data import

```r
if (!require("ipumsr")) stop("Reading IPUMS data into R requires the ipumsr
package. It can be installed using the following command:
install.packages('ipumsr')")

ddi <- read_ipums_ddi("cps_00009.xml")
cpsdata <- read_ipums_micro(ddi)
```

```
## Use of data from IPUMS CPS is subject to conditions including that users
should
## cite the data appropriately. Use command `ipums_conditions()` for more
details.
```

Data Cleaning

```r
# Adults
cpsdata = subset(cpsdata, AGE >= 18)

# Industry movement variable
cpsdata$indmove = ifelse(cpsdata$IND1990==cpsdata$IND90LY, 1, 0)

# Live in a rural county (less than 100k people total)
cpsdata$rural = ifelse(cpsdata$COUNTY==0, 1, 0)

# Manufacturing workers
censustonaics = read_xlsx("1990censustoNAICS.xlsx")
censustonaics$`1990 Census` = as.numeric(censustonaics$`1990 Census`)
```

```
## Warning: NAs introduced by coercion
```

```r
cpsdata = cpsdata %>%
  rename("1990 Census" = "IND1990")
cpsdata = cpsdata %>%
  left_join(censustonaics, by = "1990 Census")
manfacNAICS = c(11,21,31,32,33)
cpsdata$manufacturing = ifelse(cpsdata$`1997 NAICS` %in% manfacNAICS, 1, 0)

# Health variable from 1 to 5 to continuous measure between 0 and 1
cpsdata$HEALTH = ifelse(cpsdata$HEALTH == 1, 1,
       ifelse(cpsdata$HEALTH == 2, 0.75,
              ifelse(cpsdata$HEALTH == 3, 0.5,
                     ifelse(cpsdata$HEALTH == 4, 0.25,
                            ifelse(cpsdata$HEALTH == 5, 0, NA)))))
```

```
# Reconciling SCHLCOLL var to a single category of not-in-school
cpsdata$SCHLCOLL = ifelse(cpsdata$SCHLCOLL==5, 0, cpsdata$SCHLCOLL)

# Categorizing EDUC into Below HS, HS, Some college, Associates, Bachelors,
Graduate training
cpsdata$EDUC = ifelse(cpsdata$EDUC == 91, 92, cpsdata$EDUC)
cpsdata$Education = ifelse(cpsdata$EDUC < 73, 1,
                               ifelse(cpsdata$EDUC == 73, 2,
                                      ifelse(cpsdata$EDUC == 81, 3,
                                             ifelse(cpsdata$EDUC == 92, 4,
                                                    ifelse(cpsdata$EDUC ==
111, 5,

ifelse(cpsdata$EDUC > 111, 6, cpsdata$EDUC))))))

# Adding GDP per capita by state (Probably shouldn't use this variable.
Population/GDP numbers are by state due to unavailable counties)
countypopdata = read_excel(path = "County data 2001-19 (Autosaved).xlsx",
sheet = 1)
countyGDPdata = read_csv("County GDP data 2001-19.csv")

## Rows: 108056 Columns: 28

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr (26): GeoFIPS, GeoName, TableName, IndustryClassification,
Description, ...
## dbl  (2): Region, LineCode

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

countypopdata = countypopdata[c("GeoFIPS","GeoName","LineCode","2019")]
countypopdata = countypopdata %>%
  rename("Population2019" = "2019")
countypopdata$GeoFIPS = str_sub(countypopdata$GeoFIPS,2,-2)
countyGDPdata = countyGDPdata[c("GeoFIPS","GeoName","LineCode","2019")]
countyGDPdata = countyGDPdata %>%
  rename("GDP2019" = "2019")

countypopdata = subset(countypopdata, countypopdata$LineCode %in% 10)
countyGDPdata = subset(countyGDPdata, countyGDPdata$LineCode %in% 10)

## Warning: One or more parsing issues, see `problems()` for details

countydata = merge(countypopdata, countyGDPdata, by.x = "GeoFIPS", by.y =
"GeoFIPS")
```

```
countydata =
countydata[c("GeoFIPS","GeoName.x","LineCode.x","Population2019", "GDP2019")]

cpsdata$COUNTY = ifelse(cpsdata$COUNTY == 00000, cpsdata$STATEFIP,
cpsdata$COUNTY)
cpsdata$COUNTY = str_pad(cpsdata$COUNTY, width=5, side="right", pad="0")
cpsdata$STATEFIP = str_pad(cpsdata$STATEFIP, width=5, side="right", pad="0")

countydata$Population2019 = as.numeric(countydata$Population2019)
```

```
## Warning: NAs introduced by coercion
```

```
countydata$GDP2019 = as.numeric(countydata$GDP2019)
```

```
## Warning: NAs introduced by coercion
```

```
countydata = transform(countydata, GDPpercap = GDP2019*1000 / Population2019)

cpsdata1 = merge(cpsdata, countydata, by.x = "STATEFIP", by.y = "GeoFIPS")


# Adding job status vars
cpsdata1$changedjob = ifelse(cpsdata1$OCC1990 == cpsdata1$OCC90LY, 1,
                             ifelse(cpsdata1$OCC1990 != cpsdata1$OCC90LY, 2,
0))

cpsdata1$lostjob1 = ifelse(cpsdata1$OCC1990 == 999, 2, 0)
cpsdata1$lostjob2 = ifelse(cpsdata1$OCC90LY == 999, 1, 0)

cpsdata1$unempoversamp = ifelse(cpsdata1$OCC1990 + cpsdata1$OCC90LY == 3, 1,
0)
cpsdata1$lostjob = ifelse(cpsdata1$OCC90LY - cpsdata1$OCC1990 == -2, 1, 0)

# Race recode
cpsdata1$RACE = ifelse(cpsdata1$RACE == 999, 0, cpsdata1$RACE)
cpsdata1$RACE = ifelse(cpsdata1$RACE == 100, "White",
                   ifelse(cpsdata1$RACE == 200, "Black",
                       ifelse(cpsdata1$RACE == 651, "Asian",
                           ifelse(cpsdata1$RACE == 652, "Asian", "Other"))))

# Hispanic
cpsdata1$HISPAN = ifelse(cpsdata1$HISPAN > 0, 1, 0)

# Married or not
cpsdata1$MARST = ifelse(cpsdata1$MARST > 2, 1, 0)

# Population status
cpsdata1$POPSTAT = ifelse(cpsdata1$POPSTAT == 1, "Civilian",
                         ifelse(cpsdata1$POPSTAT == 2, "Military",
                                ifelse(cpsdata1$POPSTAT == 3, "Child", 0)))
```

```r
# Citizen
cpsdata1$CITIZEN = ifelse(cpsdata1$CITIZEN == 5, 0, 1)

# Nativity
cpsdata1$NATIVITY = ifelse(cpsdata1$NATIVITY == 0, 1,
                          ifelse(cpsdata1$NATIVITY == 5, 1, 0))
cpsdata1 = cpsdata1 %>%
  rename("bornabroad" = "NATIVITY")

# Empstat
cpsdata1$EMPSTAT = ifelse(cpsdata1$EMPSTAT == 1, "Military",
                         ifelse(cpsdata1$EMPSTAT == 10, "Employed",
                                ifelse(cpsdata1$EMPSTAT == 12, "Employed",
                                       ifelse(cpsdata1$EMPSTAT == 21,
"Unemployed",
                                              ifelse(cpsdata1$EMPSTAT == 22,
"Unemployed",
                                                     ifelse(cpsdata1$EMPSTAT
> 29, "NILF", NA))))))

# Labor force or NILF
cpsdata1$LABFORCE = ifelse(cpsdata1$LABFORCE == 2, 1, 0)

# Migration recode
cpsdata1$MIGRATE1 = ifelse(cpsdata1$MIGRATE1 == 1, "Same residence",
       ifelse(cpsdata1$MIGRATE1 == 3, "Moved, same county",
              ifelse(cpsdata1$MIGRATE1 == 4, "Moved, diff county",
                     ifelse(cpsdata1$MIGRATE1 == 5, "Moved, new state",
                            ifelse(cpsdata1$MIGRATE1 == 6, "Moved, abroad",
0)))))

# Disability recode
cpsdata1$DISABWRK = ifelse(cpsdata1$DISABWRK == 2, 1, 0)

# Adjusting income vars to non-negative and logging
cpsdata1$HHINCOME = log(cpsdata1$HHINCOME + abs(min(cpsdata1$HHINCOME)) + 1)
cpsdata1$FTOTVAL = log(cpsdata1$FTOTVAL + abs(min(cpsdata1$FTOTVAL)) + 1)
cpsdata1$INCTOT = log(cpsdata1$INCTOT + abs(min(cpsdata1$INCTOT)) + 1)



# Keeping useful vars/cleanup
cpsdata1 = cpsdata1[c("HEALTH", "RACE", "HISPAN", "SEX", "MARST", "METRO",
"Education", "STATEFIP", "HHINCOME", "AGE", "PERNUM", "VETSTAT", "FAMSIZE",
"LABFORCE", "bornabroad", "UHRSWORKT", "indmove", "lostjob", "ANYCOVNW")]

# Cleaning up hours worked
cpsdata1$UHRSWORKT = ifelse(cpsdata1$UHRSWORKT > 169, 0, cpsdata$UHRSWORKT)
```

```r
# Metropolitan area (1 for metro area, 0 for not)
cpsdata1$METRO = ifelse(cpsdata1$METRO == 0, 0,
                        ifelse(cpsdata1$METRO == 1, 0,
                               ifelse(cpsdata1$METRO > 1.5, 1, 0)))

# Binary sex
cpsdata1$SEX = ifelse(cpsdata1$SEX == 1, 0,
                      ifelse(cpsdata1$SEX == 2, 1, 0))
cpsdata1 = cpsdata1 %>%
  rename("FEMALE" = "SEX")

# Binary health var
cpsdata1$goodhealth = ifelse(cpsdata1$HEALTH > 0.26, 1, 0)

cpsdataoriginaldataset = cpsdata
cpsdata = cpsdata1
```

Setting X vars and Y of health

```r
cpshealth = cpsdata1[cpsdata1$HEALTH>0, ]
cpshealth = cpsdata1[cpsdata1$AGE>0, ]
cpshealth = cpsdata1[cpsdata1$FAMSIZE>0, ]
cpshealth = cpsdata1[cpsdata1$Education>1, ]
cpshealth = cpsdata1[cpsdata1$HHINCOME>=0, ]
cpshealth$RACE = as.factor(cpshealth$RACE)
cpshealth$Education = as.factor(cpshealth$Education)
cpshealth$STATEFIP = as.factor(cpshealth$STATEFIP)
x = data.matrix(cpshealth[ c("RACE", "HISPAN", "FEMALE", "MARST", "METRO",
"Education", "HHINCOME", "AGE", "VETSTAT", "FAMSIZE", "LABFORCE",
"bornabroad", "UHRSWORKT", "indmove", "lostjob", "ANYCOVNW")])
y = cpshealth$HEALTH

sumstats = data.frame(matrix(NA, nrow = 20, ncol = 1))
sumstats$mean = sapply(cpshealth, mean, na.rm=TRUE)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```r
summary(cpshealth)
```

```
##      HEALTH          RACE           HISPAN           FEMALE
##  Min.   :0.0000   Asian: 7375   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.5000   Black:15439   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.7500   Other: 4281   Median :0.0000   Median :1.0000
##  Mean   :0.6655   White:93106   Mean   :0.1555   Mean   :0.5191
##  3rd Qu.:1.0000                 3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.0000                 Max.   :1.0000   Max.   :1.0000
##
##      MARST            METRO        Education    STATEFIP        HHINCOME
##  Min.   :0.0000   Min.   :0.0000   1:12740   48000  : 8374   Min.   : 0.00
##  1st Qu.:0.0000   1st Qu.:1.0000   2:34682   12000  : 6418   1st Qu.:10.88
##  Median :0.0000   Median :1.0000   3:21210   36000  : 5866   Median :11.40
##  Mean   :0.4409   Mean   :0.8005   4:12212   40000  : 4341   Mean   :11.37
##  3rd Qu.:1.0000   3rd Qu.:1.0000   5:25025   10000  : 4137   3rd Qu.:11.87
##  Max.   :1.0000   Max.   :1.0000   6:14332   50000  : 4068   Max.   :14.63
##                                              (Other):86997
##       AGE            PERNUM          VETSTAT         FAMSIZE
##  Min.   :18.00   Min.   : 1.000   Min.   :0.000   Min.   : 1.000
##  1st Qu.:33.00   1st Qu.: 1.000   1st Qu.:1.000   1st Qu.: 2.000
##  Median :46.00   Median : 1.000   Median :1.000   Median : 3.000
##  Mean   :47.31   Mean   : 1.703   Mean   :1.071   Mean   : 2.949
##  3rd Qu.:61.00   3rd Qu.: 2.000   3rd Qu.:1.000   3rd Qu.: 4.000
##  Max.   :85.00   Max.   :16.000   Max.   :2.000   Max.   :14.000
##
##     LABFORCE        bornabroad       UHRSWORKT        indmove
##  Min.   :0.0000   Min.   :0.0000   Min.   :  0.0   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:  0.0   1st Qu.:1.0000
##  Median :1.0000   Median :0.0000   Median : 40.0   Median :1.0000
##  Mean   :0.6605   Mean   :0.1691   Mean   :253.1   Mean   :0.8103
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.: 60.0   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :999.0   Max.   :1.0000
##
##     lostjob            ANYCOVNW       goodhealth
##  Min.   :0.000000   Min.   :1.000   Min.   :0.0000
##  1st Qu.:0.000000   1st Qu.:1.000   1st Qu.:1.0000
##  Median :0.000000   Median :1.000   Median :1.0000
##  Mean   :0.001656   Mean   :1.101   Mean   :0.8611
##  3rd Qu.:0.000000   3rd Qu.:1.000   3rd Qu.:1.0000
##  Max.   :1.000000   Max.   :2.000   Max.   :1.0000
##
```

End cleaning, begin models

OLS

```
OLS = lm(y ~ factor(RACE) + HISPAN + FEMALE + MARST + METRO + Education +
factor(STATEFIP) + HHINCOME + AGE+ VETSTAT + FAMSIZE + LABFORCE+ bornabroad +
UHRSWORKT + indmove + lostjob + ANYCOVNW, data = cpshealth)
summary(OLS)
```

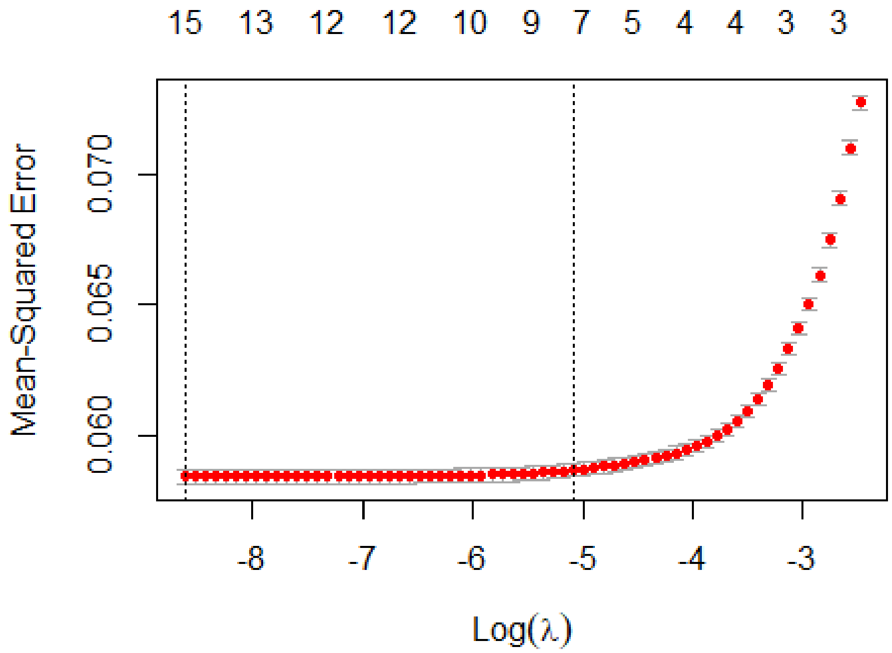
```
##
## Call:
## lm(formula = y ~ factor(RACE) + HISPAN + FEMALE + MARST + METRO +
##     Education + factor(STATEFIP) + HHINCOME + AGE + VETSTAT +
##     FAMSIZE + LABFORCE + bornabroad + UHRSWORKT + indmove + lostjob +
##     ANYCOVNW, data = cpshealth)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9712 -0.1711  0.0113  0.1908  0.7241
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            7.882e-02  1.494e-02   5.277 1.31e-07 ***
## factor(RACE)Black     -6.205e-03  3.944e-03  -1.573 0.115626
## factor(RACE)Other     -1.026e-02  4.993e-03  -2.054 0.039954 *
## factor(RACE)White      2.394e-02  3.522e-03   6.796 1.08e-11 ***
## HISPAN                -9.594e-03  2.420e-03  -3.964 7.37e-05 ***
## FEMALE                -4.091e-04  1.450e-03  -0.282 0.777890
## MARST                 -2.283e-02  1.616e-03 -14.131  < 2e-16 ***
## METRO                  2.044e-03  1.932e-03   1.058 0.290027
## Education2             2.502e-02  2.582e-03   9.688  < 2e-16 ***
## Education3             4.804e-02  2.832e-03  16.960  < 2e-16 ***
## Education4             4.857e-02  3.202e-03  15.169  < 2e-16 ***
## Education5             7.932e-02  2.873e-03  27.606  < 2e-16 ***
## Education6             9.637e-02  3.239e-03  29.757  < 2e-16 ***
## factor(STATEFIP)11000  4.820e-02  6.377e-03   7.559 4.10e-14 ***
## factor(STATEFIP)12000  4.807e-02  4.864e-03   9.883  < 2e-16 ***
## factor(STATEFIP)13000  3.121e-02  5.756e-03   5.422 5.90e-08 ***
## factor(STATEFIP)15000  1.526e-03  6.612e-03   0.231 0.817526
## factor(STATEFIP)16000  4.146e-02  6.432e-03   6.445 1.16e-10 ***
## factor(STATEFIP)17000  1.107e-02  5.463e-03   2.027 0.042663 *
## factor(STATEFIP)18000  2.110e-03  6.466e-03   0.326 0.744237
## factor(STATEFIP)19000  2.091e-02  7.226e-03   2.894 0.003802 **
## factor(STATEFIP)20000  2.765e-02  5.696e-03   4.855 1.21e-06 ***
## factor(STATEFIP)21000 -1.195e-02  7.368e-03  -1.622 0.104892
## factor(STATEFIP)22000  6.892e-03  5.793e-03   1.190 0.234147
## factor(STATEFIP)23000  1.280e-02  8.522e-03   1.503 0.132948
## factor(STATEFIP)24000  1.890e-02  6.896e-03   2.741 0.006121 **
## factor(STATEFIP)25000  1.342e-02  5.896e-03   2.276 0.022865 *
## factor(STATEFIP)26000  1.500e-02  5.837e-03   2.569 0.010191 *
## factor(STATEFIP)27000  2.422e-02  6.987e-03   3.466 0.000529 ***
## factor(STATEFIP)28000 -8.733e-03  6.183e-03  -1.412 0.157865
## factor(STATEFIP)29000  2.351e-02  6.848e-03   3.433 0.000597 ***
## factor(STATEFIP)30000  4.176e-02  6.438e-03   6.486 8.83e-11 ***
## factor(STATEFIP)31000  8.286e-03  7.141e-03   1.160 0.245898
## factor(STATEFIP)32000  1.784e-02  6.700e-03   2.663 0.007744 **
## factor(STATEFIP)33000  2.280e-02  6.906e-03   3.302 0.000960 ***
## factor(STATEFIP)34000  3.509e-02  5.891e-03   5.956 2.59e-09 ***
## factor(STATEFIP)35000  1.588e-02  6.127e-03   2.592 0.009552 **
## factor(STATEFIP)36000  1.748e-02  4.937e-03   3.540 0.000400 ***
## factor(STATEFIP)37000  1.434e-02  5.674e-03   2.527 0.011506 *
## factor(STATEFIP)38000  1.629e-02  6.909e-03   2.358 0.018377 *
## factor(STATEFIP)39000  8.290e-03  5.722e-03   1.449 0.147434
## factor(STATEFIP)40000  6.216e-03  5.278e-03   1.178 0.238866
## factor(STATEFIP)41000  2.213e-02  6.525e-03   3.392 0.000695 ***
## factor(STATEFIP)42000  3.673e-03  5.567e-03   0.660 0.509367
## factor(STATEFIP)44000  5.579e-02  7.927e-03   7.039 1.95e-12 ***
## factor(STATEFIP)45000  1.699e-02  6.313e-03   2.691 0.007123 **
## factor(STATEFIP)46000 -5.320e-03  7.386e-03  -0.720 0.471344
## factor(STATEFIP)47000  2.567e-02  6.009e-03   4.272 1.94e-05 ***
## factor(STATEFIP)48000  1.503e-03  4.664e-03   0.322 0.747252
## factor(STATEFIP)49000  2.494e-02  6.367e-03   3.916 8.99e-05 ***
## factor(STATEFIP)50000  8.931e-03  5.348e-03   1.670 0.094952 .
## factor(STATEFIP)51000  1.196e-02  6.041e-03   1.980 0.047750 *
## factor(STATEFIP)53000  2.886e-02  6.036e-03   4.781 1.75e-06 ***
## factor(STATEFIP)54000 -4.818e-02  6.050e-03  -7.964 1.68e-15 ***
## factor(STATEFIP)55000  2.081e-02  6.924e-03   3.006 0.002652 **
## factor(STATEFIP)56000  1.384e-02  7.010e-03   1.974 0.048358 *
## HHINCOME               5.885e-02  1.151e-03  51.123  < 2e-16 ***
## AGE                   -3.802e-03  4.892e-05 -77.711  < 2e-16 ***
## VETSTAT               -2.151e-02  2.696e-03  -7.978 1.50e-15 ***
## FAMSIZE               -1.350e-03  5.220e-04  -2.586 0.009705 **
## LABFORCE               7.114e-02  1.842e-03  38.633  < 2e-16 ***
## bornabroad             1.757e-02  2.385e-03   7.368 1.75e-13 ***
## UHRSWORKT              7.414e-06  1.849e-06   4.010 6.08e-05 ***
## indmove                9.774e-05  1.817e-03   0.054 0.957114
## lostjob                1.285e-02  1.715e-02   0.749 0.453670
## ANYCOVNW               5.704e-04  2.456e-03   0.232 0.816321
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.241 on 120135 degrees of freedom
## Multiple R-squared:  0.202,  Adjusted R-squared:  0.2016
## F-statistic: 467.9 on 65 and 120135 DF,  p-value: < 2.2e-16

ols_result = tidy(OLS)
write.table(ols_result, file = "olstab.txt", sep = ",", quote = FALSE,
row.names = F)
```

LASSO

```
lasso.model = cv.glmnet(x, y, type.measure = "mse")

#plot it
plot(lasso.model)
```



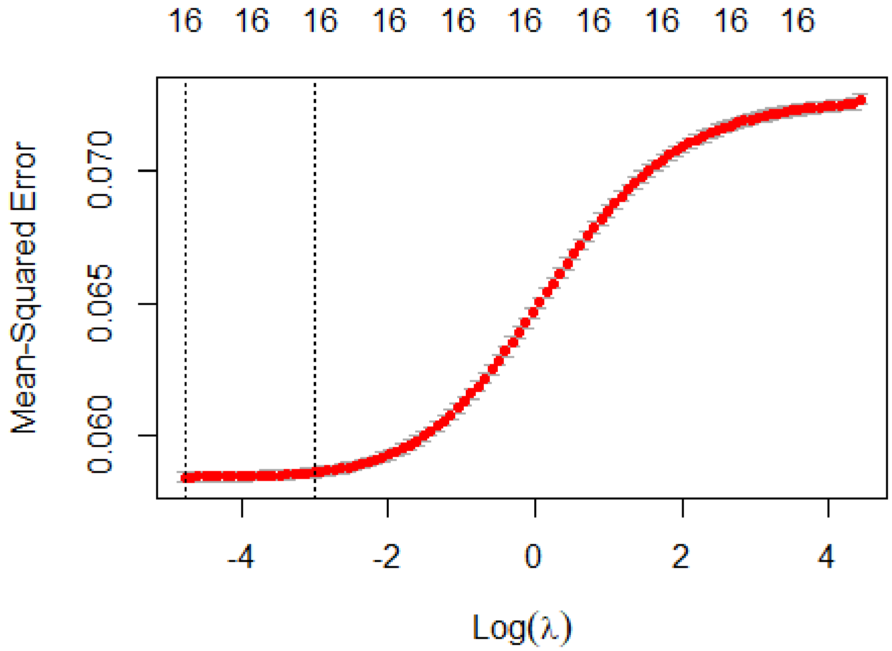
```
#predicts lowest mse with 5 variable model

#checking coef's for lasso
coef(lasso.model)

## 17 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept)  0.0568692341
## RACE         0.0027331371
## HISPAN       .
## FEMALE       .
## MARST       -0.0113268326
## METRO        .
## Education    0.0167298081
## HHINCOME     0.0587363969
## AGE         -0.0034401217
## VETSTAT     -0.0036661160
## FAMSIZE      .
## LABFORCE     0.0696933591
## bornabroad   0.0005167682
## UHRSWORKT    .
## indmove      .
## lostjob      .
## ANYCOVNW     .
```

Ridge

```r
ridge.model = cv.glmnet(x,y, type.measure='mse', alpha=0)
#plot it
plot(ridge.model)
```



```r
#selects 8 model but with log-L of -3.something
coef(ridge.model)
```

```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept)  9.149541e-02
## RACE         1.003285e-02
## HISPAN      -6.402117e-03
## FEMALE      -2.336425e-03
## MARST       -1.696660e-02
## METRO        8.877182e-03
## Education    1.763242e-02
## HHINCOME     5.312037e-02
## AGE         -3.124688e-03
## VETSTAT     -2.562623e-02
## FAMSIZE      1.629458e-03
## LABFORCE     6.873288e-02
## bornabroad   1.713803e-02
## UHRSWORKT    1.245815e-05
## indmove     -3.527772e-03
## lostjob      1.511455e-02
## ANYCOVNW    -5.735121e-04
```

PCR

```
pcr.fit <- pcr(y ~ x, data = cpshealth, scale = TRUE, validation = "CV")
```

KNN

```
# Prepping the dataset (no factor vars)
cpsKNNt = cpshealth[complete.cases(cpshealth), ]
cpsKNN = cpshealth[c("RACE", "HISPAN", "FEMALE", "MARST", "METRO",
"Education", "HHINCOME", "AGE", "VETSTAT", "FAMSIZE", "LABFORCE",
"bornabroad", "UHRSWORKT", "indmove", "lostjob", "ANYCOVNW")]
cpsKNN$RACE = as.factor(cpsKNN$RACE)
cpsKNN$RACE = as.numeric(cpsKNN$RACE)
cpsKNN$AGE = as.numeric(cpsKNN$AGE)

# Prepping KNN
samp = sample(1:nrow(cpsKNN), 0.5 * nrow(cpsKNN))
knn.train = cpsKNN[samp,]
knn.test = cpsKNN[-samp,]
train.health = cpsKNNt$HEALTH[samp]
test.health = cpsKNNt$HEALTH[-samp]

# Running KNN and reporting results
knn.results = data.frame(k = 1:10, testerror = NA)
for(i in 1:10){
knn.pred <- knn(knn.train, knn.test, train.health, k = i)
knn.results$testerror[i] = mean(knn.pred == test.health)
}
knn.results
```

```
##     k testerror
## 1   1 0.3198782
## 2   2 0.3139216
## 3   3 0.3238881
## 4   4 0.3299446
## 5   5 0.3336051
## 6   6 0.3359678
## 7   7 0.3418080
## 8   8 0.3425401
## 9   9 0.3451856
## 10 10 0.3468827
```

```
# Running KNN and reporting results BACKUP (manually adjust K)
# knn.pred <- knn(knn.train, knn.test, train.health, k = 1)
# mean(knn.pred == test.health)
```