Demonstration of Regularization

Holly

# Demonstration of Regularization Methods

This will be a demonstration of the three regularization methods discussed: ridge regression, Lasso (least absolute shrinkage and selection operator) and Elastic Net.

## Description of Data

The data we will be using are from the 2019 County Health Rankings. They provide data on a number of demographic, social, environmental and health characteristics on counties within the United States. We will be using this dataset to try to identify the most important predictors of life expectancy on a county-level. We have restricted the dataset to 67 features and an outcome of life expectancy in years.

Original data upon which this exercise has been based can be found here: <http://www.countyhealthrankings.org/explore-health-rankings/rankings-data-documentation>

Variable names are not originally informative. You can look up all full variable name meanings here: <http://www.countyhealthrankings.org/sites/default/files/2019%20Analytic%20Documentation_1.pdf>

### Load needed libraries

library(tidyverse)

## ── Attaching packages ────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.2.1 ✔ purrr 0.3.2  
## ✔ tibble 2.1.3 ✔ dplyr 0.8.3  
## ✔ tidyr 1.0.0 ✔ stringr 1.4.0  
## ✔ readr 1.3.1 ✔ forcats 0.4.0

## ── Conflicts ───────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

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

## Loaded glmnet 3.0-2

### Step 1: Read in data, partition, remove outcome variable and standardize

set.seed(100)  
  
chr = read\_csv("./data/chr.csv")

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:  
## cols(  
## .default = col\_double()  
## )

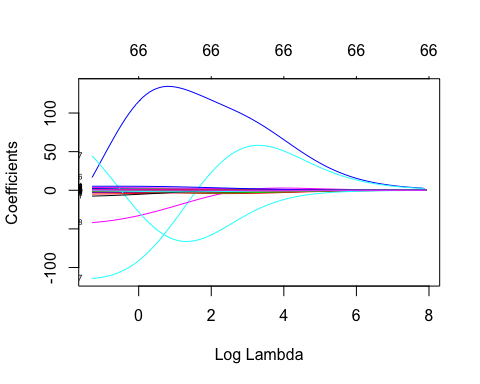
## See spec(...) for full column specifications.

chr = chr[,2:68]  
  
var.names<-c("pre\_death", "poorhealth", "poorphyshealth\_days", "poormenthealth\_days", "low\_bwt", "ad\_smoking", "ad\_obesity", "foodenv\_index", "phys\_inactivity", "exer\_access", "excess\_drink", "alc\_drivdeaths", "sti", "teen\_birth", "uninsured", "primcareproviders", "dentists", "menthealthproviders", "prevhosp", "mammo\_screen", "flu\_vacc", "hsgrad", "somecollege", "unemployed", "child\_poverty", "income\_ineq", "sing\_parent", "social\_assoc", "violent\_crime", "injury\_deaths", "pm\_air", "water\_viol", "housing\_prob", "driving\_alone", "long\_commute", "life\_exp", "age\_adj\_premortality", "freq\_physdistress", "freq\_mentdistress", "diabetes", "hiv", "food\_insecure", "ltd\_access\_healthyfood", "mvcrash\_deaths", "insuff\_sleep", "uninsured\_adults", "uninsured\_child", "other\_pcp", "medhhinc", "freelunch\_child", "res\_seg\_bw", "res\_seg\_nw", "firearm\_fatalities", "homeownership", "hous\_cost\_burden", "population", "bw18", "gte65", "nonhisp\_afam", "AmerInd\_AlasNative", "Asian", "OPacIslander", "Hisp", "nonhisp\_white", "nonprof\_english", "female", "rural")  
  
colnames(chr) = var.names  
  
  
  
#Reminder of non-tidyverse way to create data partition  
#train.indices<-createDataPartition(y=bc.data$outcome,p=0.7,list=FALSE)  
  
training.data = chr$life\_exp %>% createDataPartition(p=0.7, list=F)  
train.data = chr[training.data, ]  
test.data = chr[-training.data, ]  
  
#Store outcome   
life.exp.train = train.data$life\_exp  
life.exp.test = test.data$life\_exp  
  
x.train = model.matrix(life\_exp~., train.data)[,-1]  
x.test = model.matrix(life\_exp~., test.data)[,-1]

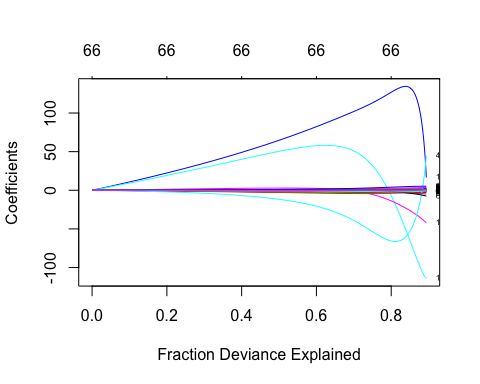
### Step 2: Running the algorithms on the training data

The glmnet package allows us to run all three of the penalized models using the same format. The value of the alpha parameter dictates whether it is a ride regression, lasso or elastic net. A value of 0 is the ridge regression, the 1 is a lasso and any value in between 0 and 1 will provide an elastic net. The package also takes an input for lambda, but by default it will vary lambda and provide you output for 100 options. There is also an option to use cross-validation to choose the optimal labmda. That requires use of cv.glmnet().

set.seed(100)  
  
#Ridge Regression  
  
model.1 = glmnet(x.train, life.exp.train, alpha=0, standardize = TRUE)  
  
plot(model.1, xvar = "lambda", label = TRUE)



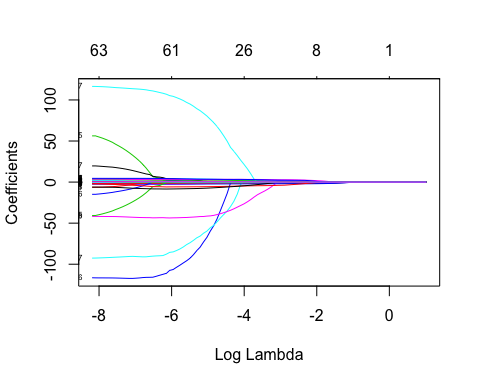
plot(model.1, xvar = "dev", label = TRUE)



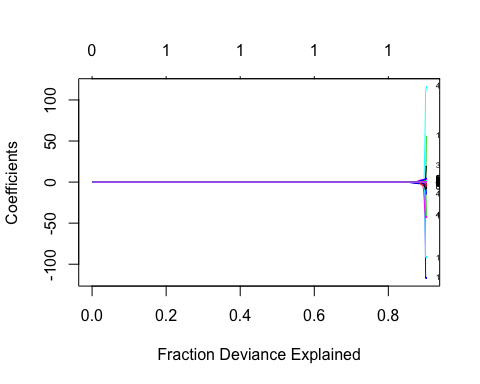
model.1$beta[,1]

## pre\_death poorhealth poorphyshealth\_days   
## -1.044641e-39 -4.182733e-35 -2.800615e-36   
## poormenthealth\_days low\_bwt ad\_smoking   
## -3.039589e-36 -7.135769e-35 -5.663069e-35   
## ad\_obesity foodenv\_index phys\_inactivity   
## -3.524223e-35 1.130179e-36 -3.698144e-35   
## exer\_access excess\_drink alc\_drivdeaths   
## 4.590675e-36 4.970278e-35 9.883232e-37   
## sti teen\_birth uninsured   
## -3.620104e-39 -1.303081e-37 -1.290828e-35   
## primcareproviders dentists menthealthproviders   
## 2.391599e-33 2.094149e-33 1.463303e-34   
## prevhosp mammo\_screen flu\_vacc   
## -7.035315e-40 1.322887e-35 6.772733e-36   
## hsgrad somecollege unemployed   
## 3.311604e-36 1.355549e-35 -7.441631e-35   
## child\_poverty income\_ineq sing\_parent   
## -2.161531e-35 -1.395223e-36 -1.371572e-35   
## social\_assoc violent\_crime injury\_deaths   
## 2.267857e-38 -3.384358e-39 -6.494387e-38   
## pm\_air water\_viol housing\_prob   
## -4.627325e-37 2.440544e-37 -2.670455e-36   
## driving\_alone long\_commute age\_adj\_premortality   
## -1.219253e-35 -2.302443e-36 -2.530916e-38   
## freq\_physdistress freq\_mentdistress diabetes   
## -8.736439e-35 -1.077930e-34 -7.406201e-35   
## hiv food\_insecure ltd\_access\_healthyfood   
## -1.502556e-39 -4.180127e-35 -3.149857e-36   
## mvcrash\_deaths insuff\_sleep uninsured\_adults   
## -1.710173e-37 -3.783913e-35 -1.292515e-35   
## uninsured\_child other\_pcp medhhinc   
## 4.096762e-36 -1.612955e-34 1.374378e-40   
## freelunch\_child res\_seg\_bw res\_seg\_nw   
## -9.132110e-36 4.298860e-38 -3.255461e-39   
## firearm\_fatalities homeownership hous\_cost\_burden   
## -2.284355e-37 1.343661e-36 -2.612396e-37   
## population bw18 gte65   
## 2.019381e-43 -1.285647e-35 1.302173e-36   
## nonhisp\_afam AmerInd\_AlasNative Asian   
## -6.761285e-36 -8.533026e-36 3.157120e-35   
## OPacIslander Hisp nonhisp\_white   
## 1.248409e-35 3.958457e-36 2.045484e-36   
## nonprof\_english female rural   
## 2.595595e-35 -1.593835e-35 -1.884897e-36

#LASSO  
  
model.2 = glmnet(x.train, life.exp.train, alpha = 1, standardize = TRUE)  
  
plot(model.2, xvar = "lambda", label = TRUE)



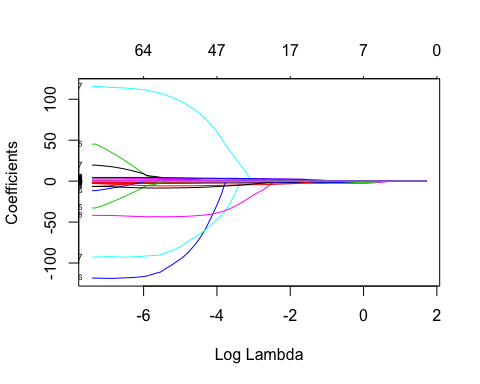
plot(model.2, xvar = "dev", label = TRUE)



model.2$beta[,1]

## pre\_death poorhealth poorphyshealth\_days   
## 0 0 0   
## poormenthealth\_days low\_bwt ad\_smoking   
## 0 0 0   
## ad\_obesity foodenv\_index phys\_inactivity   
## 0 0 0   
## exer\_access excess\_drink alc\_drivdeaths   
## 0 0 0   
## sti teen\_birth uninsured   
## 0 0 0   
## primcareproviders dentists menthealthproviders   
## 0 0 0   
## prevhosp mammo\_screen flu\_vacc   
## 0 0 0   
## hsgrad somecollege unemployed   
## 0 0 0   
## child\_poverty income\_ineq sing\_parent   
## 0 0 0   
## social\_assoc violent\_crime injury\_deaths   
## 0 0 0   
## pm\_air water\_viol housing\_prob   
## 0 0 0   
## driving\_alone long\_commute age\_adj\_premortality   
## 0 0 0   
## freq\_physdistress freq\_mentdistress diabetes   
## 0 0 0   
## hiv food\_insecure ltd\_access\_healthyfood   
## 0 0 0   
## mvcrash\_deaths insuff\_sleep uninsured\_adults   
## 0 0 0   
## uninsured\_child other\_pcp medhhinc   
## 0 0 0   
## freelunch\_child res\_seg\_bw res\_seg\_nw   
## 0 0 0   
## firearm\_fatalities homeownership hous\_cost\_burden   
## 0 0 0   
## population bw18 gte65   
## 0 0 0   
## nonhisp\_afam AmerInd\_AlasNative Asian   
## 0 0 0   
## OPacIslander Hisp nonhisp\_white   
## 0 0 0   
## nonprof\_english female rural   
## 0 0 0

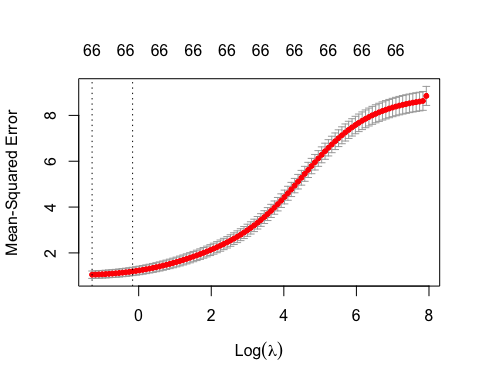
#Elastic Net  
  
model.3 = glmnet(x.train, life.exp.train, alpha = 0.5, standardize = TRUE)  
  
plot(model.3, xvar="lambda", label=TRUE)



### Step 3: Using cross-validation to select the optimal value for lambda (tuning parameter)

Reminder when lambda is 0, you will obtain OLS regressio coefficients (i.e. no regularization) When lambda approaches large numbers, the regression coefficents will shrink toward 0

model.1.cv = cv.glmnet(x.train, life.exp.train, alpha = 0)  
plot(model.1.cv)



model.1.cv$lambda.min

## [1] 0.2769684

model.1.train.final = glmnet(x.train, life.exp.train, alpha = 0, lambda = model.1.cv$lambda.min)  
  
coef(model.1.train.final)

## 67 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) 9.021441e+01  
## pre\_death -3.584473e-04  
## poorhealth 5.437197e-01  
## poorphyshealth\_days -2.328341e-02  
## poormenthealth\_days -8.472745e-02  
## low\_bwt 2.987797e+00  
## ad\_smoking 7.152633e-02  
## ad\_obesity -1.382442e+00  
## foodenv\_index -1.651177e-02  
## phys\_inactivity -1.791057e+00  
## exer\_access 3.010981e-02  
## excess\_drink -4.926021e-01  
## alc\_drivdeaths 9.773472e-03  
## sti -1.870197e-04  
## teen\_birth -3.156286e-03  
## uninsured 4.483475e-01  
## primcareproviders 1.478602e+01  
## dentists -1.130063e+02  
## menthealthproviders -4.214774e+01  
## prevhosp -2.113512e-05  
## mammo\_screen -6.846172e-02  
## flu\_vacc -2.613681e-01  
## hsgrad -2.106240e-01  
## somecollege 5.431905e-01  
## unemployed -1.890544e+00  
## child\_poverty -6.588387e-01  
## income\_ineq 5.124795e-02  
## sing\_parent 2.437990e-01  
## social\_assoc -6.141228e-03  
## violent\_crime -1.795008e-04  
## injury\_deaths -8.165181e-03  
## pm\_air -8.953747e-02  
## water\_viol 4.922816e-02  
## housing\_prob 1.076685e-01  
## driving\_alone -2.646691e+00  
## long\_commute -4.780854e-01  
## age\_adj\_premortality -1.043811e-02  
## freq\_physdistress 2.777219e+00  
## freq\_mentdistress -6.099310e-01  
## diabetes 4.435263e-01  
## hiv -1.832418e-04  
## food\_insecure -1.216124e+00  
## ltd\_access\_healthyfood 7.178899e-01  
## mvcrash\_deaths -1.729747e-03  
## insuff\_sleep -2.124469e+00  
## uninsured\_adults -9.719083e-01  
## uninsured\_child 1.952680e+00  
## other\_pcp 4.385860e+01  
## medhhinc 1.625789e-05  
## freelunch\_child -3.042469e-01  
## res\_seg\_bw 3.661754e-03  
## res\_seg\_nw -6.524025e-04  
## firearm\_fatalities -1.619081e-03  
## homeownership 4.631898e-02  
## hous\_cost\_burden 5.833970e-01  
## population -9.769097e-09  
## bw18 -4.915141e+00  
## gte65 3.561275e+00  
## nonhisp\_afam 6.221978e-01  
## AmerInd\_AlasNative 6.727415e-01  
## Asian 3.510615e+00  
## OPacIslander -7.484634e+00  
## Hisp 3.591323e-01  
## nonhisp\_white -5.621868e-01  
## nonprof\_english 5.197662e+00  
## female -3.653002e-01  
## rural 3.978236e-01

### Step 4: Apply model to test set and evaluate model

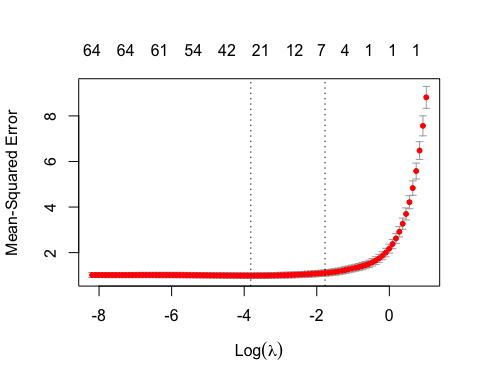
model.1.test.pred = model.1.train.final %>% predict(x.test) %>% as.vector()  
  
data.frame(RMSE = RMSE(model.1.test.pred, life.exp.test), RSQ = R2(model.1.test.pred, life.exp.test))

## RMSE RSQ  
## 1 1.10401 0.8576341

### Exercise

Using cross-validation, find the optimal values for lambda when using lasso and elastic net, setting the alpha of the elastic net to 0.5. Then apply the final models to the test set. Which model would you choose if this were your study? Why? (Note, again normally we wouldn’t compare models within the test set. We would either have a validation set, or would assess error in the training set.)

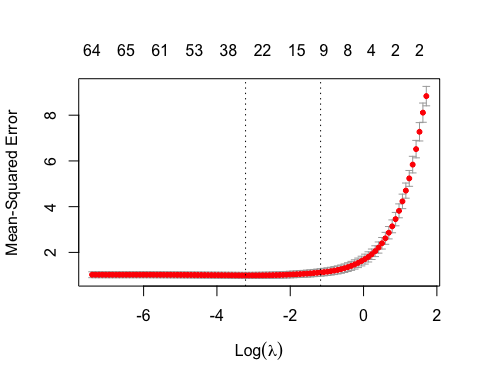
# LASSO  
model.lasso.cv = cv.glmnet(x.train, life.exp.train, alpha = 1)  
plot(model.lasso.cv)



model.lasso.train.final = glmnet(x.train, life.exp.train, alpha = 1, lambda = model.lasso.cv$lambda.min)  
  
model.lasso.test.pred = model.lasso.train.final %>% predict(x.test) %>% as.vector()  
  
data.frame(RMSE = RMSE(model.lasso.test.pred, life.exp.test), RSQ = R2(model.lasso.test.pred, life.exp.test))

## RMSE RSQ  
## 1 1.095779 0.8626421

# ELASTIC  
model.elastic.cv = cv.glmnet(x.train, life.exp.train, alpha = 0.5)  
plot(model.elastic.cv)



model.elastic.train.final = glmnet(x.train, life.exp.train, alpha = 0.5, lambda = model.elastic.cv$lambda.min)  
  
model.elastic.test.pred = model.elastic.train.final %>% predict(x.test) %>% as.vector()  
  
data.frame(RMSE = RMSE(model.elastic.test.pred, life.exp.test), RSQ = R2(model.elastic.test.pred, life.exp.test))

## RMSE RSQ  
## 1 1.104217 0.8604078

### Step 5: Using caret to select best tuning parameters

I will demonstrate how you can use the caret package to construct penalized regressions. By default, caret will vary both alpha and lambda to select the best values via cross-validation. Because the alpha is not set at 0 or 1, this is typically results in an elastic net. But, you can set the alpha level at a fixed value in order to obtain ridge or lasso results.

tuneLength sets the number of combinations of different values of alpha and lambda to compare.

set.seed(123)  
en.model = train(  
 life\_exp ~., data = train.data, method = "glmnet",  
 trControl = trainControl("cv", number = 10),  
 tuneLength = 10  
 )  
  
en.model$bestTune

## alpha lambda  
## 55 0.6 0.03644533

# Model coefficients  
coef(en.model$finalModel, en.model$bestTune$lambda)

## 67 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 8.996010e+01  
## pre\_death -2.705044e-04  
## poorhealth .   
## poorphyshealth\_days .   
## poormenthealth\_days .   
## low\_bwt 1.648604e+00  
## ad\_smoking .   
## ad\_obesity -1.095288e+00  
## foodenv\_index .   
## phys\_inactivity -2.616403e-01  
## exer\_access .   
## excess\_drink .   
## alc\_drivdeaths .   
## sti .   
## teen\_birth .   
## uninsured .   
## primcareproviders .   
## dentists .   
## menthealthproviders -2.078831e+01  
## prevhosp .   
## mammo\_screen -2.593944e-02  
## flu\_vacc -4.034535e-02  
## hsgrad .   
## somecollege .   
## unemployed .   
## child\_poverty .   
## income\_ineq 4.677286e-02  
## sing\_parent .   
## social\_assoc .   
## violent\_crime -3.327130e-05  
## injury\_deaths -2.395108e-03  
## pm\_air -9.639323e-02  
## water\_viol .   
## housing\_prob .   
## driving\_alone -2.221969e+00  
## long\_commute -1.385142e-01  
## age\_adj\_premortality -1.654244e-02  
## freq\_physdistress .   
## freq\_mentdistress .   
## diabetes .   
## hiv .   
## food\_insecure .   
## ltd\_access\_healthyfood 8.850385e-01  
## mvcrash\_deaths .   
## insuff\_sleep -8.203673e-01  
## uninsured\_adults .   
## uninsured\_child 1.162145e+00  
## other\_pcp 7.130210e+00  
## medhhinc 1.082276e-05  
## freelunch\_child .   
## res\_seg\_bw .   
## res\_seg\_nw .   
## firearm\_fatalities .   
## homeownership .   
## hous\_cost\_burden .   
## population .   
## bw18 -4.495829e+00  
## gte65 1.755908e+00  
## nonhisp\_afam .   
## AmerInd\_AlasNative .   
## Asian 3.189782e+00  
## OPacIslander -3.960023e+00  
## Hisp .   
## nonhisp\_white -5.949705e-01  
## nonprof\_english 3.135536e+00  
## female .   
## rural 1.975465e-01

# Make predictions  
  
en.pred <- en.model %>% predict(x.test)  
  
# Model prediction performance  
data.frame(  
 RMSE = RMSE(en.pred, test.data$life\_exp),  
 Rsquare = R2(en.pred, test.data$life\_exp))

## RMSE Rsquare  
## 1 1.103606 0.8605858

### Exercise:

The following code will allow you to fix the alpha (I have it set to 0 for a ridge) and run either a ridge or lasso analysis. Use that code to run both ridge and Lasso using the caret package and obtain coefficients and evaluation metrics.

If the caret package will select the optimal alpha and lambda value, why might you still choose lasso or ridge over elastic net (or an automated process of choosing alpha as in caret)?

#Create grid to search lambda  
lambda = 10^seq(-3,3, length = 100)  
  
set.seed(100)  
  
model.4 = train(  
 life\_exp ~., data = train.data, method = "glmnet", trControl = trainControl("cv", number = 10), tuneGrid = expand.grid(alpha = 0, lambda = lambda)  
)  
  
model.4$bestTune

## alpha lambda  
## 41 0 0.2656088

# Model coefficients  
coef(model.4$finalModel, model.4$bestTune$lambda)

## 67 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) 9.018814e+01  
## pre\_death -3.541750e-04  
## poorhealth 7.530885e-01  
## poorphyshealth\_days -1.237654e-02  
## poormenthealth\_days -8.506307e-02  
## low\_bwt 3.049105e+00  
## ad\_smoking 4.870326e-02  
## ad\_obesity -1.416881e+00  
## foodenv\_index -1.205913e-02  
## phys\_inactivity -1.809717e+00  
## exer\_access 3.441098e-02  
## excess\_drink -3.826116e-01  
## alc\_drivdeaths 7.963232e-03  
## sti -1.936751e-04  
## teen\_birth -3.183451e-03  
## uninsured 2.881478e-01  
## primcareproviders 1.651209e+01  
## dentists -1.140929e+02  
## menthealthproviders -4.187674e+01  
## prevhosp -2.115857e-05  
## mammo\_screen -6.852471e-02  
## flu\_vacc -2.545751e-01  
## hsgrad -2.106243e-01  
## somecollege 5.645382e-01  
## unemployed -1.997760e+00  
## child\_poverty -7.728254e-01  
## income\_ineq 4.764991e-02  
## sing\_parent 1.993557e-01  
## social\_assoc -6.113099e-03  
## violent\_crime -1.813222e-04  
## injury\_deaths -8.314198e-03  
## pm\_air -8.996460e-02  
## water\_viol 5.001249e-02  
## housing\_prob 4.045120e-02  
## driving\_alone -2.666379e+00  
## long\_commute -4.745170e-01  
## age\_adj\_premortality -1.053421e-02  
## freq\_physdistress 2.281412e+00  
## freq\_mentdistress -6.373538e-01  
## diabetes 5.070399e-01  
## hiv -1.844928e-04  
## food\_insecure -1.059238e+00  
## ltd\_access\_healthyfood 7.522076e-01  
## mvcrash\_deaths -1.655927e-03  
## insuff\_sleep -2.028710e+00  
## uninsured\_adults -8.704658e-01  
## uninsured\_child 1.963261e+00  
## other\_pcp 4.455302e+01  
## medhhinc 1.575231e-05  
## freelunch\_child -2.794262e-01  
## res\_seg\_bw 3.674658e-03  
## res\_seg\_nw -6.337591e-04  
## firearm\_fatalities -1.330792e-03  
## homeownership 3.603907e-02  
## hous\_cost\_burden 6.449476e-01  
## population -9.839047e-09  
## bw18 -4.876077e+00  
## gte65 3.602634e+00  
## nonhisp\_afam 6.303873e-01  
## AmerInd\_AlasNative 7.069451e-01  
## Asian 3.502702e+00  
## OPacIslander -7.540999e+00  
## Hisp 3.589263e-01  
## nonhisp\_white -5.721625e-01  
## nonprof\_english 5.183767e+00  
## female -3.742071e-01  
## rural 4.025355e-01

# Make predictions  
  
pred4 = model.4 %>% predict(x.test)  
  
# Model prediction performance  
data.frame(  
 RMSE = RMSE(pred4, test.data$life\_exp),  
 Rsquare = R2(pred4, test.data$life\_exp))

## RMSE Rsquare  
## 1 1.102672 0.8580168