Mixed Effects Model

Group

2023-12-01

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
library(knitr)
# library(lme4)
# library(glmmLasso)
# library(MuMIn)
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
```

```
# load data
data_clean <- read.csv("data/data_clean.csv")

# get relevant columns: HDI + ROLI overall factors and specific subfactors
cols <- c("hdi", "year", "country", "region", colnames(data_clean)[39:90])

# extract those columns -- not subsetting by year here
# remove summary factors
data <- data_clean[,cols] %>%
  select(-contains("factor"))

# remove any rows with NA --> all data
df <- data[complete.cases(data), ]

# data in most recent year - 2021
df1 <- df[df$year==2021,]
df1 <- subset(df1, select=-c(year))
```

Methods: - run using data in a singular year (choose 2021 because most recent year and largest sample size) - choose obs from multiple years –> data on all countries available. we have 138 unique countries in dataset, all of them are in 2021 data –> this method is the same as the 1st method. - using the entire entire dataset

2021 data

```
# model with all variables
model1 <- lm(hdi~.-country, df1)

# backward selection
model2 <- step(model1, direction="backward", trace=0)

# intercept model
interceptModel <- lm(hdi~1, df1)

# interaction
interactionModel <- lm(hdi~.+region*., df1)

# forward selection
model3 <- step(interceptModel, scope = list(upper = formula(model1)),
               direction = "forward", trace=0)

# stepwise selection
model4 <- step(model2, scope = list(lower = formula(interceptModel),
                                    upper = formula(model1)),
               direction = "both", trace=0)

# # save models
# saveRDS(model1, file = "pred_models/model1.rds")
# saveRDS(model2, file = "pred_models/model2.rds")
# saveRDS(model3, file = "pred_models/model3.rds")
# saveRDS(model4, file = "pred_models/model4.rds")

# # load models
# model1 <- readRDS("pred_models/model1.rds")
# model2 <- readRDS("pred_models/model2.rds")
# model3 <- readRDS("pred_models/model3.rds")
# model4 <- readRDS("pred_models/model4.rds")

summary(model2)$coefficients
```

```
##                                        Estimate Std. Error    t value
## (Intercept)                          0.509694469 0.04547983 11.2070451
## regionEastern Europe & Central Asia  0.005897441 0.01688061  0.3493619
## regionEU + EFTA + North America     -0.012872721 0.01569093 -0.8203925
## regionLatin America & Caribbean     -0.045068576 0.01640976 -2.7464502
## regionMiddle East & North Africa    -0.017151819 0.01978869 -0.8667486
## regionSouth Asia                    -0.073888332 0.02260764 -3.2682908
## regionSub-Saharan Africa            -0.150619434 0.01654687 -9.1025953
## x1.2                                -0.227275129 0.08801271 -2.5822990
## x1.6                                 0.087797953 0.05166268  1.6994463
```

```
## x3.1                                  0.129135778 0.03504139  3.6852353
## x3.2                                 -0.149050944 0.04906900 -3.0375789
## x3.3                                 -0.145717839 0.07348456 -1.9829721
## x3.4                                  0.120898221 0.04716749  2.5631686
## x4.1                                 -0.180057265 0.06579870 -2.7364865
## x4.3                                  0.286502180 0.07250075  3.9517135
## x4.5                                  0.066902605 0.04603193  1.4533955
## x5.2                                  0.042777430 0.02506700  1.7065236
## x5.3                                 -0.100964462 0.03643971 -2.7707261
## x6.1                                  0.160227455 0.06171116  2.5964097
## x7.3                                  0.263795201 0.04814000  5.4797506
## x7.4                                  0.176470031 0.07076616  2.4937066
## x7.5                                 -0.051738456 0.03073497 -1.6833742
## x7.7                                  0.097704399 0.06380498  1.5312974
## x8.3                                  0.061575186 0.04678093  1.3162455
## x8.4                                 -0.087533727 0.04990750 -1.7539192
## x8.6                                 -0.113444195 0.04795146 -2.3658134
##                                          Pr(>|t|)
## (Intercept)                           5.253944e-20
## regionEastern Europe & Central Asia 7.274737e-01
## regionEU + EFTA + North America     4.137350e-01
## regionLatin America & Caribbean     7.021328e-03
## regionMiddle East & North Africa    3.879329e-01
## regionSouth Asia                      1.437163e-03
## regionSub-Saharan Africa              3.882660e-15
## x1.2                                  1.110404e-02
## x1.6                                  9.201098e-02
## x3.1                                  3.537236e-04
## x3.2                                  2.967656e-03
## x3.3                                  4.981840e-02
## x3.4                                  1.169811e-02
## x4.1                                  7.223498e-03
## x4.3                                  1.361531e-04
## x4.5                                  1.489103e-01
## x5.2                                  9.068192e-02
## x5.3                                  6.550129e-03
## x6.1                                  1.068340e-02
## x7.3                                  2.653958e-07
## x7.4                                  1.410275e-02
## x7.5                                  9.508811e-02
## x7.7                                  1.285156e-01
## x8.3                                  1.907791e-01
## x8.4                                  8.217975e-02
## x8.6                                  1.970907e-02
```

```r
# make tables
summary2021 <- data.frame("AIC" = c(AIC(model1),
                                    AIC(model2), AIC(model3), AIC(model4)),

                          "BIC" = c(BIC(model1),
                                    BIC(model2), BIC(model3), BIC(model4)),

            "r.squared" = c(summary(model1)$r.squared,
                            summary(model2)$r.squared,
```

```
                          summary(model3)$r.squared, summary(model4)$r.squared))
rownames(summary2021) <- c("all var", "backward selection", "forward selection", "stepwise")

# print table
kable(summary2021)
```

	AIC	BIC	r.squared
all var	-459.2934	-312.9307	0.9537182
backward selection	-488.8380	-409.8021	0.9478570
forward selection	-485.3319	-409.2233	0.9457344
stepwise	-488.8380	-409.8021	0.9478570

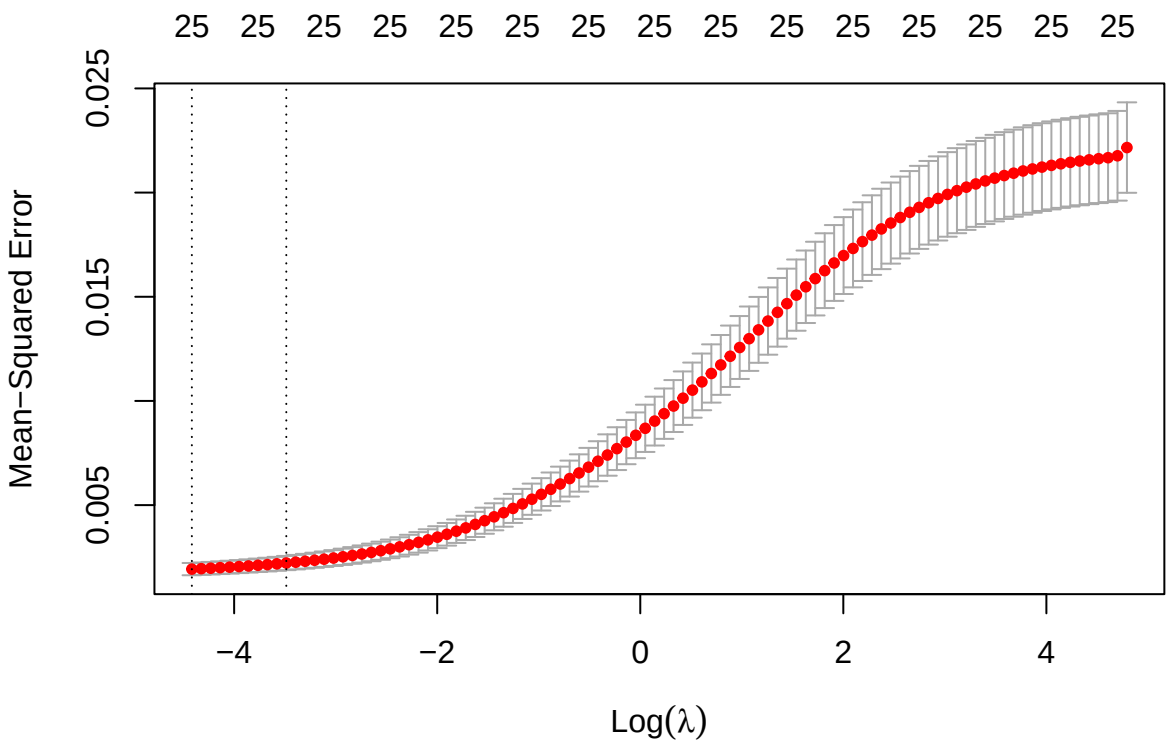
```
# get predictors
X = model.matrix(model2)[,-1]

# fit
ridge <- cv.glmnet(X, df1$hdi, alpha=0)
lasso <- cv.glmnet(X, df1$hdi, alpha=1)

ridge
```

```
##
## Call:  cv.glmnet(x = X, y = df1$hdi, alpha = 0)
##
## Measure: Mean-Squared Error
##
##      Lambda Index  Measure        SE Nonzero
## min 0.01207   100 0.001928 0.0002940      25
## 1se 0.03061    90 0.002220 0.0003403      25
```

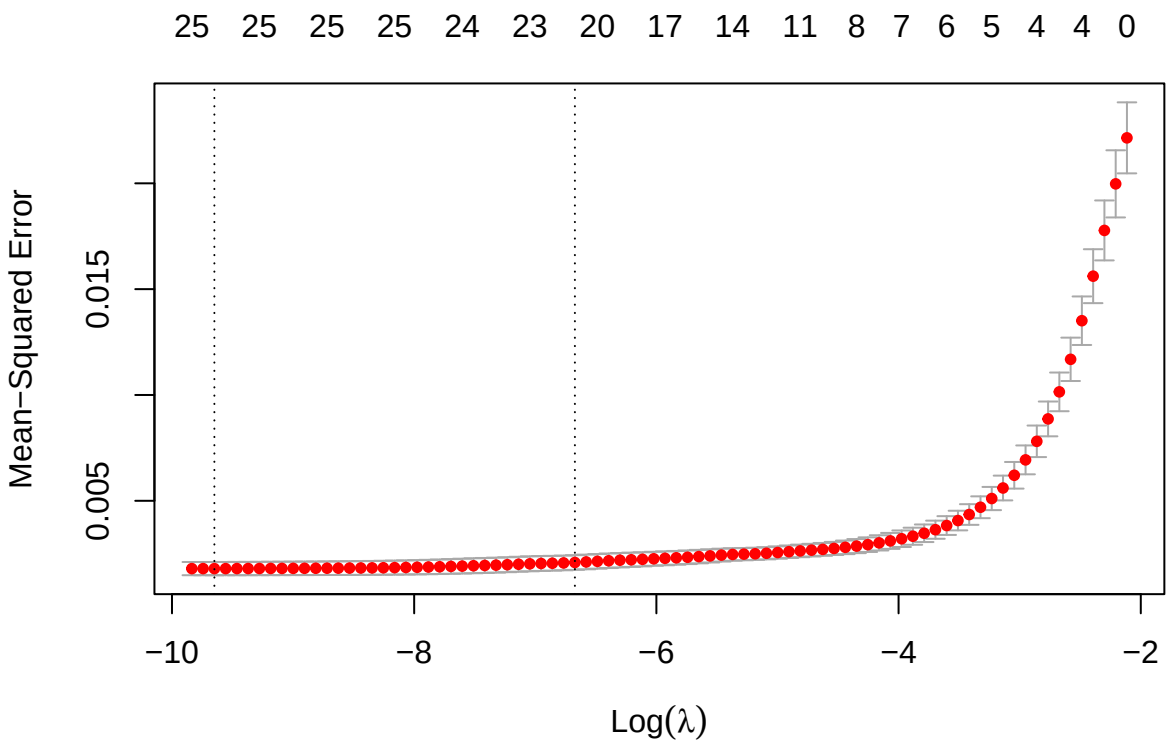
```
lasso
```

```
##
## Call:  cv.glmnet(x = X, y = df1$hdi, alpha = 1)
##
## Measure: Mean-Squared Error
##
##        Lambda Index  Measure        SE Nonzero
## min 0.0000644    82 0.001795 0.0003129      25
## 1se 0.0012648    50 0.002084 0.0003469      20
```

```
# plot
plot(ridge)
```


25 25 25 25 25 25 25 25 25 25 25 25 25 25 25
Mean−Squared Error
0.005
0.015
0.025
−4
−2
0
2
4
Log(λ)

```
plot(lasso)
```




```
# best lambda
ridge$lambda.min
```

```
## [1] 0.01207299
```

```
lasso$lambda.min
```

```
## [1] 6.442994e-05
```

```
# get mse values
min(ridge$cvm)
```

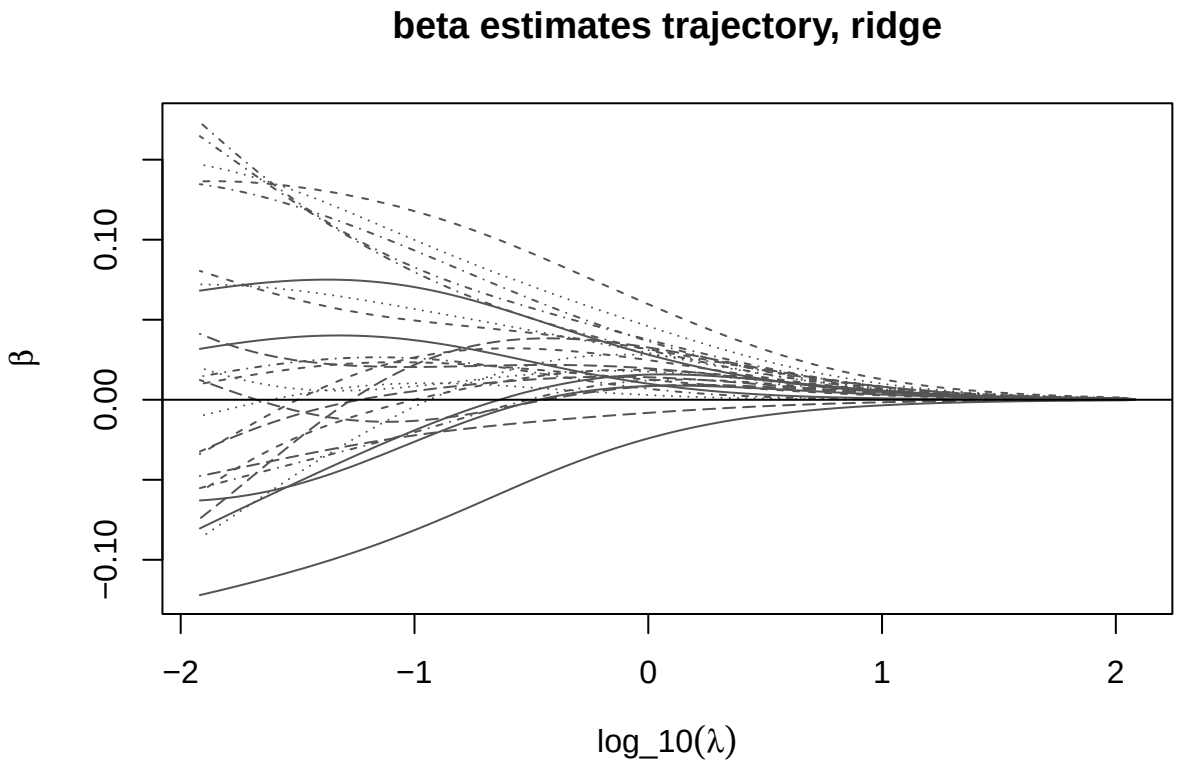
```
## [1] 0.001928398
```

```
min(lasso$cvm)
```

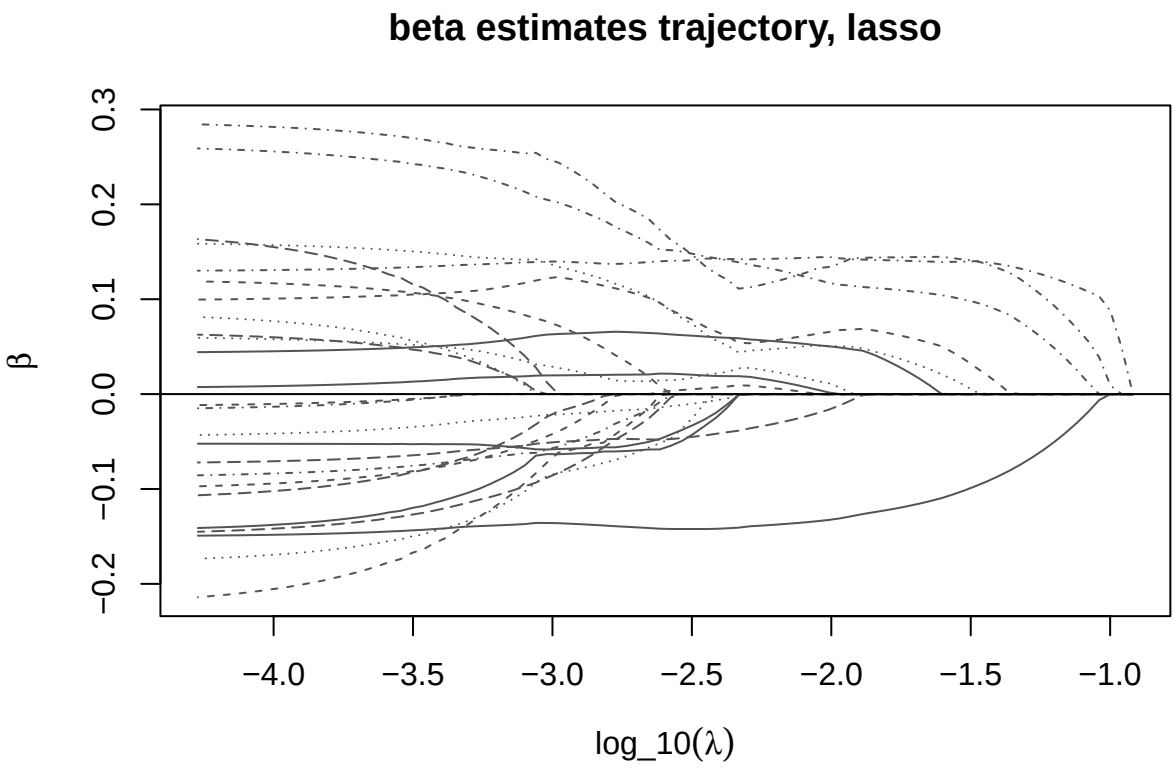
```
## [1] 0.001795371
```

```
# beta trajectories
matplot(log(ridge$glmnet.fit$lambda,10),
        t(ridge$glmnet.fit$beta),
        type="l",col="gray33",lwd=1,
        xlab=expression(log_10(lambda)),
```

```
        ylab=expression(hat(beta)),
        main="beta estimates trajectory, ridge")
abline(h=0)
```




```
matplot(log(lasso$glmnet.fit$lambda,10),
        t(lasso$glmnet.fit$beta),
        type="l",col="gray33",lwd=1,
        xlab=expression(log_10(lambda)),
        ylab=expression(hat(beta)),
        main="beta estimates trajectory, lasso")
abline(h=0)
```




```
best_lasso <- glmnet(x=subset(df,select=-hdi), y=df$hdi, lambda=lasso$lambda.min)
coef(best_lasso)
```

```
## 48 x 1 sparse Matrix of class "dgCMatrix"
##                                                              s0
## (Intercept)                                         0.5568561921
## year                                               -0.0011816220
## country                                             0.0001412369
## region                                             -0.0247041138
## x1.1                                                0.0004458603
## x1.2                                               -0.1201170037
## x1.3                                                0.0234750712
## x1.4                                               -0.1755269929
## x1.5                                                0.0034371081
## x1.6                                                0.0646162587
## x2.1                                                0.0988601402
## x2.2                                                0.0033237669
## x2.3                                                0.0391652592
## x2.4                                               -0.0405615287
## x3.1                                                0.1167591850
## x3.2                                               -0.0236075927
## x3.3                                               -0.0995309096
## x3.4                                                0.1023867705
## x4.1                                               -0.0661349071
## x4.2                                                0.0247504196
```

```
## x4.3                                                    0.3104562519
## x4.4                                                    .
## x4.5                                                   -0.0778113987
## x4.6                                                   -0.0311902398
## x4.7                                                   -0.0300071961
## x4.8                                                    0.0106856804
## x5.1                                                   -0.0004840530
## x5.2                                                    0.0282273123
## x5.3                                                   -0.0884538363
## x6.1                                                    0.2467080785
## x6.2                                                    .
## x6.3                                                   -0.0411646745
## x6.4                                                    0.0168579509
## x6.5                                                   -0.0579930355
## x7.1                                                    0.1054108244
## x7.2                                                   -0.0432727159
## x7.3                                                    0.2817190226
## x7.4                                                    0.0576878977
## x7.5                                                   -0.0501633418
## x7.6                                                    0.0394606365
## x7.7                                                    0.0205498939
## x8.1                                                   -0.0303229100
## x8.2                                                   -0.0119293846
## x8.3                                                    0.0351309788
## x8.4                                                   -0.0455678605
## x8.5                                                   -0.1142080204
## x8.6                                                   -0.0023784042
## x8.7..due.process.of.the.law.and.rights.of.the.accused  .
```

```
remaining_lasso <- glmnet(x=subset(df,select=-hdi), y=df$hdi, lambda=0.05011872)
coef(remaining_lasso)
```

```
## 48 x 1 sparse Matrix of class "dgCMatrix"
##                                                                s0
## (Intercept)                                          0.6266328572
## year                                                 .
## country                                              .
## region                                              -0.0113137262
## x1.1                                                 .
## x1.2                                                 .
## x1.3                                                 .
## x1.4                                                 .
## x1.5                                                 .
## x1.6                                                 .
## x2.1                                                 .
## x2.2                                                 0.0308178138
## x2.3                                                 0.1071129425
## x2.4                                                 .
## x3.1                                                 0.0730543746
## x3.2                                                 .
## x3.3                                                 .
## x3.4                                                 .
## x4.1                                                 .
## x4.2                                                 .
```

```
## x4.3                                                    0.0830521785
## x4.4                                                    .
## x4.5                                                    .
## x4.6                                                    .
## x4.7                                                    .
## x4.8                                                    .
## x5.1                                                    .
## x5.2                                                    .
## x5.3                                                    .
## x6.1                                                    .
## x6.2                                                    .
## x6.3                                                    .
## x6.4                                                    .
## x6.5                                                    .
## x7.1                                                    .
## x7.2                                                    .
## x7.3                                                    .
## x7.4                                                    .
## x7.5                                                    .
## x7.6                                                    .
## x7.7                                                    .
## x8.1                                                    .
## x8.2                                                    .
## x8.3                                                    .
## x8.4                                                    .
## x8.5                                                    .
## x8.6                                                    .
## x8.7..due.process.of.the.law.and.rights.of.the.accused  0.0007588301
```

```
best_ridge <- glmnet(x=subset(df,select=-hdi), y=df$hdi, lambda=ridge$lambda.min)
coef(best_ridge)
```

```
## 48 x 1 sparse Matrix of class "dgCMatrix"
##                                                                s0
## (Intercept)                                          5.532716e-01
## year                                                 .
## country                                              .
## region                                              -2.444478e-02
## x1.1                                                 .
## x1.2                                                 .
## x1.3                                                 .
## x1.4                                                 .
## x1.5                                                 .
## x1.6                                                 .
## x2.1                                                 .
## x2.2                                                 1.046325e-01
## x2.3                                                 2.155684e-02
## x2.4                                                 .
## x3.1                                                 1.071083e-01
## x3.2                                                 .
## x3.3                                                 .
## x3.4                                                 8.658088e-03
## x4.1                                                 .
## x4.2                                                 1.735460e-02
```

```
## x4.3                                                    3.810274e-07
## x4.4                                                    .
## x4.5                                                    .
## x4.6                                                    .
## x4.7                                                    .
## x4.8                                                    .
## x5.1                                                    .
## x5.2                                                    .
## x5.3                                                    .
## x6.1                                                    5.487182e-02
## x6.2                                                    .
## x6.3                                                    .
## x6.4                                                    .
## x6.5                                                    .
## x7.1                                                    9.679583e-02
## x7.2                                                    .
## x7.3                                                    4.179879e-02
## x7.4                                                    .
## x7.5                                                    .
## x7.6                                                    .
## x7.7                                                    .
## x8.1                                                    .
## x8.2                                                    .
## x8.3                                                    .
## x8.4                                                    .
## x8.5                                                    .
## x8.6                                                    .
## x8.7..due.process.of.the.law.and.rights.of.the.accused  6.814420e-02
```

```r
lambda_values <- seq(from = 1e-04, to = 10^4, length.out = 5000)
bic_values <- numeric(length(lambda_values))
X <- model.matrix(model2)[,-1]
Y <- df1$hdi

for (i in seq_along(lambda_values)) {
  # Using glmnet for Lasso Regression
  lasso_model <- glmnet(X, df1$hdi, alpha = 1, lambda = lambda_values[i])

  # Calculate BIC
  rss <- sum((predict(lasso_model, newx = X, s = "lambda.min") - Y)^2)
  n <- length(Y)
  k <- sum(coef(lasso_model, s = "lambda.min") != 0)
  bic_values[i] <- n * log(rss / n) + k * log(n)
}

# Find the index of the minimum BIC value
best_lambda_index <- which.min(bic_values)
best_lambda <- lambda_values[best_lambda_index]

# Fit the final Lasso Regression model using the selected lambda
final_lasso_model <- glmnet(X, Y, alpha = 1, lambda = best_lambda)

# Display results
best_lambda
```

```
## [1] 1e-04
```

```
coef(final_lasso_model, s = "lambda.min")
```

```
## 26 x 1 sparse Matrix of class "dgCMatrix"
##                                                s1
## (Intercept)                          0.501958870
## regionEastern Europe & Central Asia  0.008179147
## regionEU + EFTA + North America     -0.010605432
## regionLatin America & Caribbean     -0.041620126
## regionMiddle East & North Africa    -0.013691624
## regionSouth Asia                    -0.070478397
## regionSub-Saharan Africa            -0.147855175
## x1.2                                -0.210204952
## x1.6                                 0.077617775
## x3.1                                 0.130081058
## x3.2                                -0.142562221
## x3.3                                -0.136821314
## x3.4                                 0.116362845
## x4.1                                -0.170257590
## x4.3                                 0.285499749
## x4.5                                 0.060377702
## x5.2                                 0.044446400
## x5.3                                -0.095787682
## x6.1                                 0.158889079
## x7.3                                 0.259012162
## x7.4                                 0.158594980
## x7.5                                -0.051809311
## x7.7                                 0.099087641
## x8.3                                 0.057964655
## x8.4                                -0.086054714
## x8.6                                -0.105283458
```