

Mixed Effects Model

Group

2023-12-01

```
library(glmnet)
```

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

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

```
library(knitr)
```

```
# library(lme4)
```

```
# library(glmnetLasso)
```

```
# 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), ]
```

```
df = df %>%
```

```
  rename(x8.7=x8.7..due.process.of.the.law.and.rights.of.the.accused)
```

```
# 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")

kable(summary(model2)$coefficients)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.5096945	0.0454798	11.2070451	0.0000000
regionEastern Europe & Central Asia	0.0058974	0.0168806	0.3493619	0.7274737
regionEU + EFTA + North America	-0.0128727	0.0156909	-0.8203925	0.4137350
regionLatin America & Caribbean	-0.0450686	0.0164098	-2.7464502	0.0070213
regionMiddle East & North Africa	-0.0171518	0.0197887	-0.8667486	0.3879329
regionSouth Asia	-0.0738883	0.0226076	-3.2682908	0.0014372
regionSub-Saharan Africa	-0.1506194	0.0165469	-9.1025953	0.0000000
x1.2	-0.2272751	0.0880127	-2.5822990	0.0111040

	Estimate	Std. Error	t value	Pr(> t)
x1.6	0.0877980	0.0516627	1.6994463	0.0920110
x3.1	0.1291358	0.0350414	3.6852353	0.0003537
x3.2	-0.1490509	0.0490690	-3.0375789	0.0029677
x3.3	-0.1457178	0.0734846	-1.9829721	0.0498184
x3.4	0.1208982	0.0471675	2.5631686	0.0116981
x4.1	-0.1800573	0.0657987	-2.7364865	0.0072235
x4.3	0.2865022	0.0725007	3.9517135	0.0001362
x4.5	0.0669026	0.0460319	1.4533955	0.1489103
x5.2	0.0427774	0.0250670	1.7065236	0.0906819
x5.3	-0.1009645	0.0364397	-2.7707261	0.0065501
x6.1	0.1602275	0.0617112	2.5964097	0.0106834
x7.3	0.2637952	0.0481400	5.4797506	0.0000003
x7.4	0.1764700	0.0707662	2.4937066	0.0141028
x7.5	-0.0517385	0.0307350	-1.6833742	0.0950881
x7.7	0.0977044	0.0638050	1.5312974	0.1285156
x8.3	0.0615752	0.0467809	1.3162455	0.1907791
x8.4	-0.0875337	0.0499075	-1.7539192	0.0821797
x8.6	-0.1134442	0.0479515	-2.3658134	0.0197091

```
# 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")

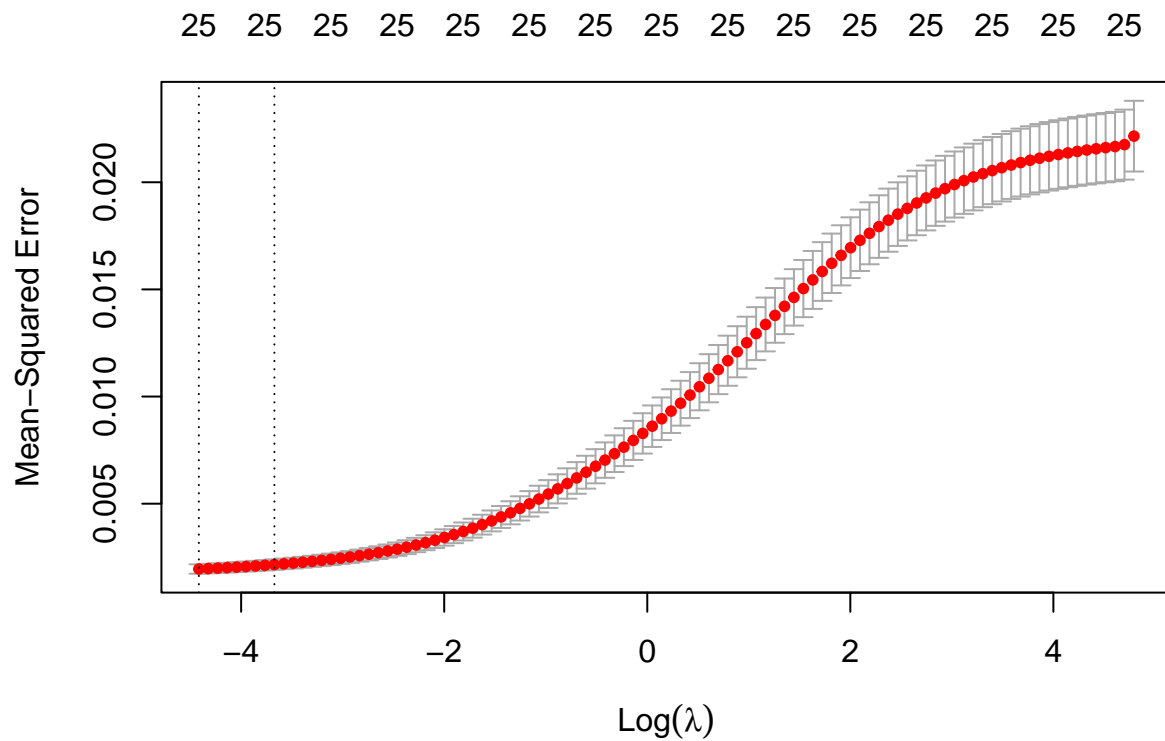
# print table
summary2021
```

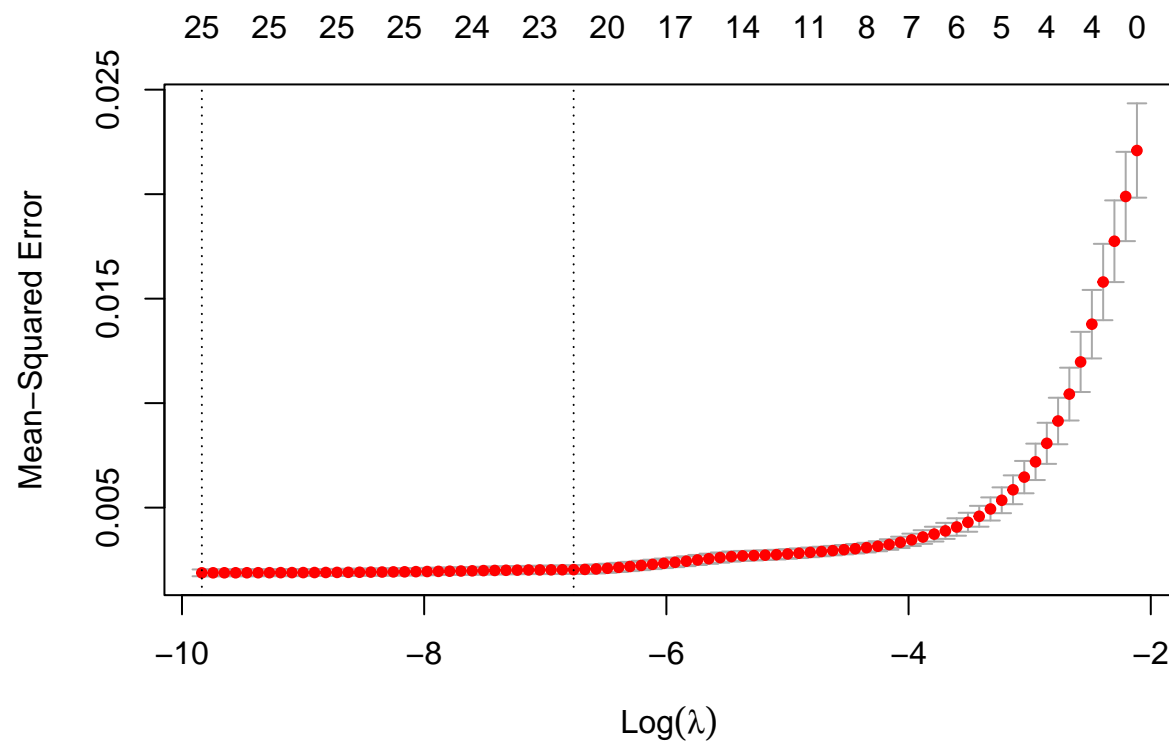
```
##           AIC      BIC r.squared
## all var    -459.2934 -312.9307 0.9537182
## backward selection -488.8380 -409.8021 0.9478570
## forward selection -485.3319 -409.2233 0.9457344
## stepwise    -488.8380 -409.8021 0.9478570
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.5096945	0.0454798	11.2070451	0.0000000
regionEastern Europe & Central Asia	0.0058974	0.0168806	0.3493619	0.7274737
regionEU + EFTA + North America	-0.0128727	0.0156909	-0.8203925	0.4137350
regionLatin America & Caribbean	-0.0450686	0.0164098	-2.7464502	0.0070213
regionMiddle East & North Africa	-0.0171518	0.0197887	-0.8667486	0.3879329
regionSouth Asia	-0.0738883	0.0226076	-3.2682908	0.0014372
regionSub-Saharan Africa	-0.1506194	0.0165469	-9.1025953	0.0000000
x1.2	-0.2272751	0.0880127	-2.5822990	0.0111040
x1.6	0.0877980	0.0516627	1.6994463	0.0920110
x3.1	0.1291358	0.0350414	3.6852353	0.0003537
x3.2	-0.1490509	0.0490690	-3.0375789	0.0029677
x3.3	-0.1457178	0.0734846	-1.9829721	0.0498184
x3.4	0.1208982	0.0471675	2.5631686	0.0116981
x4.1	-0.1800573	0.0657987	-2.7364865	0.0072235
x4.3	0.2865022	0.0725007	3.9517135	0.0001362
x4.5	0.0669026	0.0460319	1.4533955	0.1489103
x5.2	0.0427774	0.0250670	1.7065236	0.0906819
x5.3	-0.1009645	0.0364397	-2.7707261	0.0065501
x6.1	0.1602275	0.0617112	2.5964097	0.0106834
x7.3	0.2637952	0.0481400	5.4797506	0.0000003
x7.4	0.1764700	0.0707662	2.4937066	0.0141028
x7.5	-0.0517385	0.0307350	-1.6833742	0.0950881
x7.7	0.0977044	0.0638050	1.5312974	0.1285156
x8.3	0.0615752	0.0467809	1.3162455	0.1907791
x8.4	-0.0875337	0.0499075	-1.7539192	0.0821797
x8.6	-0.1134442	0.0479515	-2.3658134	0.0197091

```
##
## Call:  cv.glmnet(x = X, y = df1$hdi, alpha = 0)
##
## Measure: Mean-Squared Error
##
##      Lambda Index  Measure      SE Nonzero
## min 0.01207   100 0.001953 0.0002183      25
## 1se 0.02541    92 0.002158 0.0002433      25
```

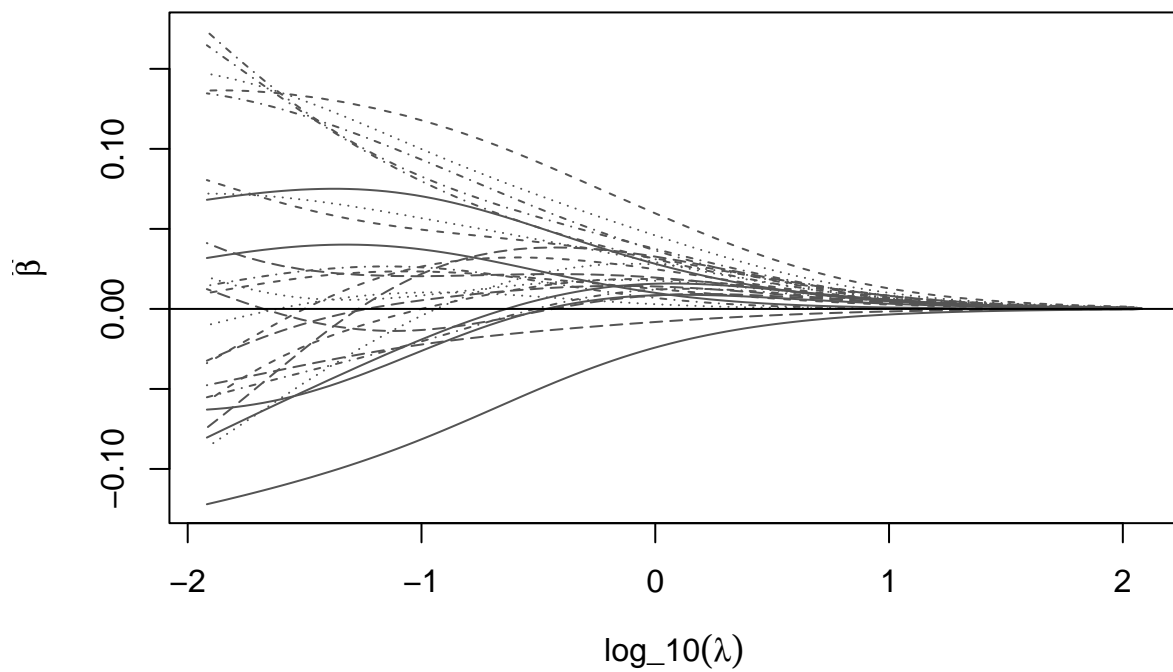
```
##
## Call:  cv.glmnet(x = X, y = df1$hdi, alpha = 1)
##
## Measure: Mean-Squared Error
##
##      Lambda Index  Measure      SE Nonzero
## min 0.0000535   84 0.001880 0.0001632      25
## 1se 0.0011524   51 0.002032 0.0002059      20
```



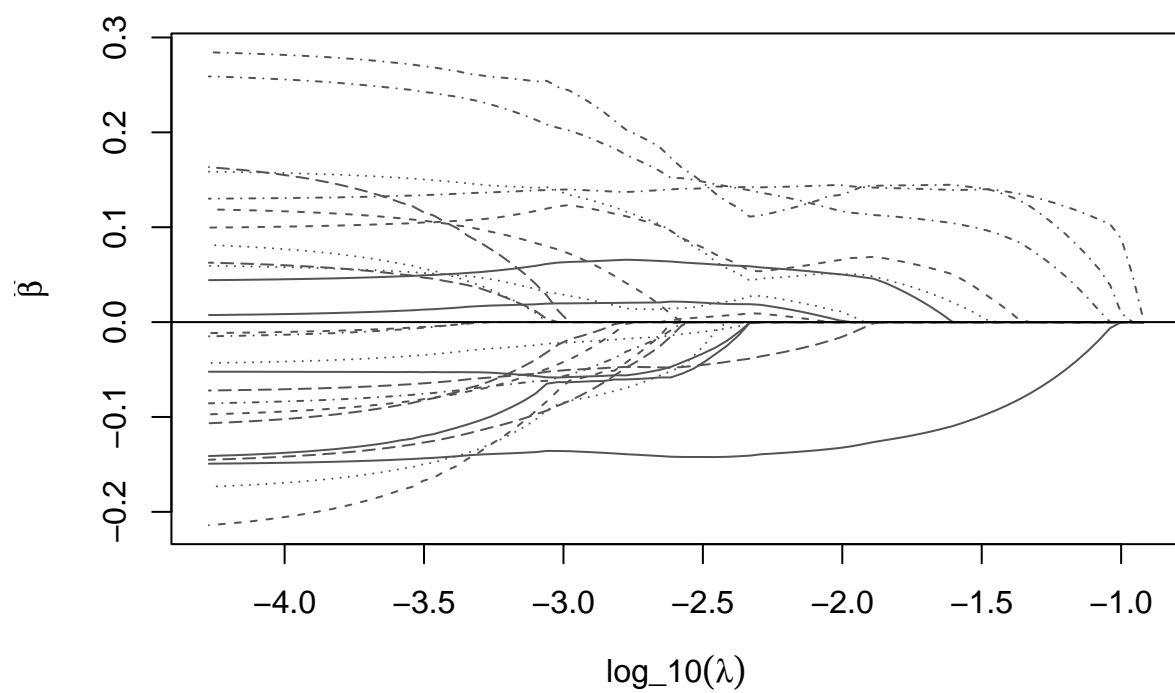


```
## [1] 0.01207299
## [1] 5.349086e-05
## [1] 0.00195249
## [1] 0.001879776
```

beta estimates trajectory, ridge



beta estimates trajectory, lasso



	Coefficients
(Intercept)	0.5577719
year	-0.0011990
country	0.0001410
region	-0.0247326
x1.1	0.0014900
x1.2	-0.1219489
x1.3	0.0243800
x1.4	-0.1756091
x1.5	0.0062781
x1.6	0.0653311
x2.1	0.0999359
x2.2	0.0029083
x2.3	0.0405825
x2.4	-0.0408441
x3.1	0.1169816
x3.2	-0.0240944
x3.3	-0.1000027
x3.4	0.1025341
x4.1	-0.0668292
x4.2	0.0254143
x4.3	0.3104820
x4.4	0.0000000
x4.5	-0.0779066
x4.6	-0.0328524
x4.7	-0.0327217
x4.8	0.0117447
x5.1	-0.0008649
x5.2	0.0282021
x5.3	-0.0890270
x6.1	0.2465939
x6.2	0.0000000
x6.3	-0.0423686
x6.4	0.0168778
x6.5	-0.0594553
x7.1	0.1048465
x7.2	-0.0436665
x7.3	0.2837542
x7.4	0.0598052
x7.5	-0.0502392
x7.6	0.0398430
x7.7	0.0205859
x8.1	-0.0303904
x8.2	-0.0121323
x8.3	0.0357835
x8.4	-0.0450268
x8.5	-0.1168357
x8.6	-0.0031291
x8.7	0.0000000

	Coefficients
(Intercept)	0.6266329
year	0.0000000
country	0.0000000
region	-0.0113137
x1.1	0.0000000
x1.2	0.0000000
x1.3	0.0000000
x1.4	0.0000000
x1.5	0.0000000
x1.6	0.0000000
x2.1	0.0000000
x2.2	0.0308178
x2.3	0.1071129
x2.4	0.0000000
x3.1	0.0730544
x3.2	0.0000000
x3.3	0.0000000
x3.4	0.0000000
x4.1	0.0000000
x4.2	0.0000000
x4.3	0.0830522
x4.4	0.0000000
x4.5	0.0000000
x4.6	0.0000000
x4.7	0.0000000
x4.8	0.0000000
x5.1	0.0000000
x5.2	0.0000000
x5.3	0.0000000
x6.1	0.0000000
x6.2	0.0000000
x6.3	0.0000000
x6.4	0.0000000
x6.5	0.0000000
x7.1	0.0000000
x7.2	0.0000000
x7.3	0.0000000
x7.4	0.0000000
x7.5	0.0000000
x7.6	0.0000000
x7.7	0.0000000
x8.1	0.0000000
x8.2	0.0000000
x8.3	0.0000000
x8.4	0.0000000
x8.5	0.0000000
x8.6	0.0000000
x8.7	0.0007588

	Coefficients
(Intercept)	0.5532716
year	0.0000000
country	0.0000000
region	-0.0244448
x1.1	0.0000000
x1.2	0.0000000
x1.3	0.0000000
x1.4	0.0000000
x1.5	0.0000000
x1.6	0.0000000
x2.1	0.0000000
x2.2	0.1046325
x2.3	0.0215568
x2.4	0.0000000
x3.1	0.1071083
x3.2	0.0000000
x3.3	0.0000000
x3.4	0.0086581
x4.1	0.0000000
x4.2	0.0173546
x4.3	0.0000004
x4.4	0.0000000
x4.5	0.0000000
x4.6	0.0000000
x4.7	0.0000000
x4.8	0.0000000
x5.1	0.0000000
x5.2	0.0000000
x5.3	0.0000000
x6.1	0.0548718
x6.2	0.0000000
x6.3	0.0000000
x6.4	0.0000000
x6.5	0.0000000
x7.1	0.0967958
x7.2	0.0000000
x7.3	0.0417988
x7.4	0.0000000
x7.5	0.0000000
x7.6	0.0000000
x7.7	0.0000000
x8.1	0.0000000
x8.2	0.0000000
x8.3	0.0000000
x8.4	0.0000000
x8.5	0.0000000
x8.6	0.0000000
x8.7	0.0681442

[1] 1e-04

	Coefficients
(Intercept)	0.5019589
regionEastern Europe & Central Asia	0.0081791
regionEU + EFTA + North America	-0.0106054
regionLatin America & Caribbean	-0.0416201
regionMiddle East & North Africa	-0.0136916
regionSouth Asia	-0.0704784
regionSub-Saharan Africa	-0.1478552
x1.2	-0.2102050
x1.6	0.0776178
x3.1	0.1300811
x3.2	-0.1425622
x3.3	-0.1368213
x3.4	0.1163628
x4.1	-0.1702576
x4.3	0.2854997
x4.5	0.0603777
x5.2	0.0444464
x5.3	-0.0957877
x6.1	0.1588891
x7.3	0.2590122
x7.4	0.1585950
x7.5	-0.0518093
x7.7	0.0990876
x8.3	0.0579647
x8.4	-0.0860547
x8.6	-0.1052835