Projet MRR

Teddy ALEXANDRE

Décembre 2022

R Markdown

This is an R Markdown document. Markdown is a simple formatting sY1ntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When Y1ou click the **Knit** button a document will be generated that includes both content as well as the output of anY1 embedded R code chunks within the document. Y1ou can embed an R code chunk like this:

```
## load data
load(file="/home/talexandre/Bureau/MRR/Projet/projet-mrr-2022-bio/Ressources/project_park.RData")
dim(geno.df)
```

```
## [1] 36901   416
```

```
dim(pheno.df)
```

```
## [1] 413  38
```

```
dim(Xmat)
```

```
## [1]   413 36901
```

```
X = Xmat
Y1 = data.matrix(pheno.df$Seed.number.per.panicle)

# Replace NA Values bY11 the mean of each column
for(i in 1:ncol(X)){
  X[is.na(X[,i]), i] <- round(mean(X[,i], na.rm = TRUE))
}
Y1[is.na(Y1)] = mean(Y1, na.rm=TRUE)


# We reduce the size of X
X = X[,sample(x = 1:36901, size=36901)]
random_sample = createDataPartition(Y1, p = 0.8, list = FALSE)
X_train = X[random_sample,]
Y1_train = Y1[random_sample]

X_test = X[-random_sample,]
Y1_test = Y1[-random_sample]

dim(Xmat)
```

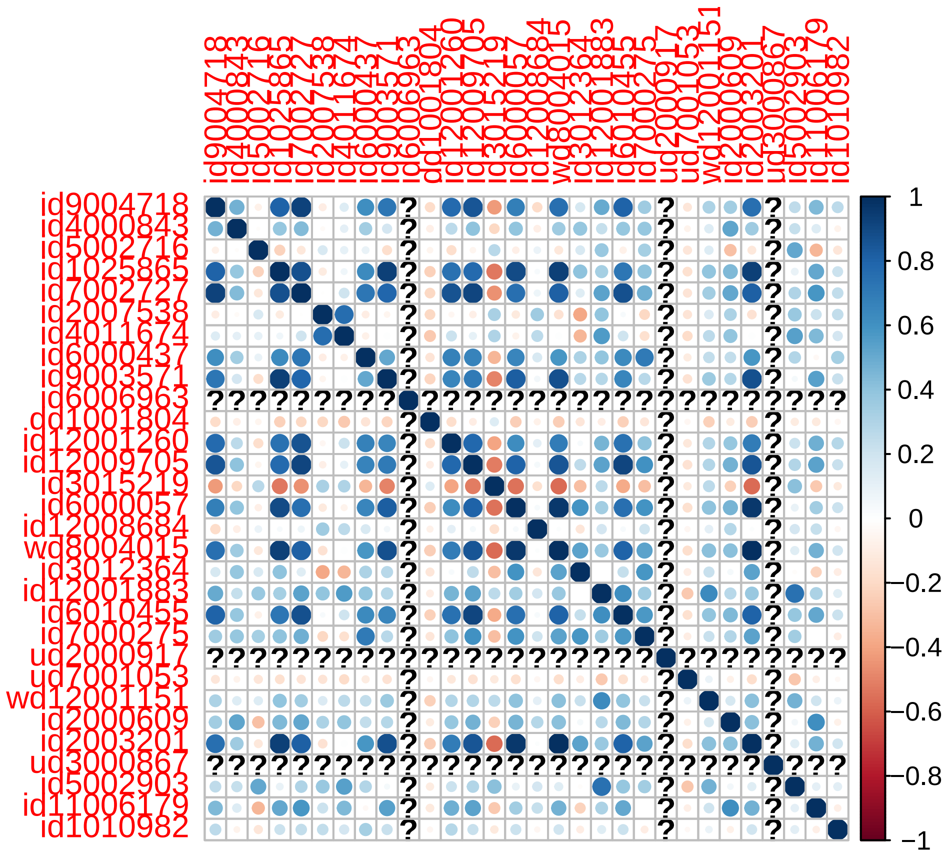
```
## [1]   413 36901
```

```
#Regular regression
#modreg = lm(Y1_train ~., data = as.data.frame(X_train))
```

```
#summarY1(modreg)
cor = cor(X[1:30,1:30])
```

```
## Warning in cor(X[1:30, 1:30]): l'écart type est nul
```

```
corrplot(cor)
```

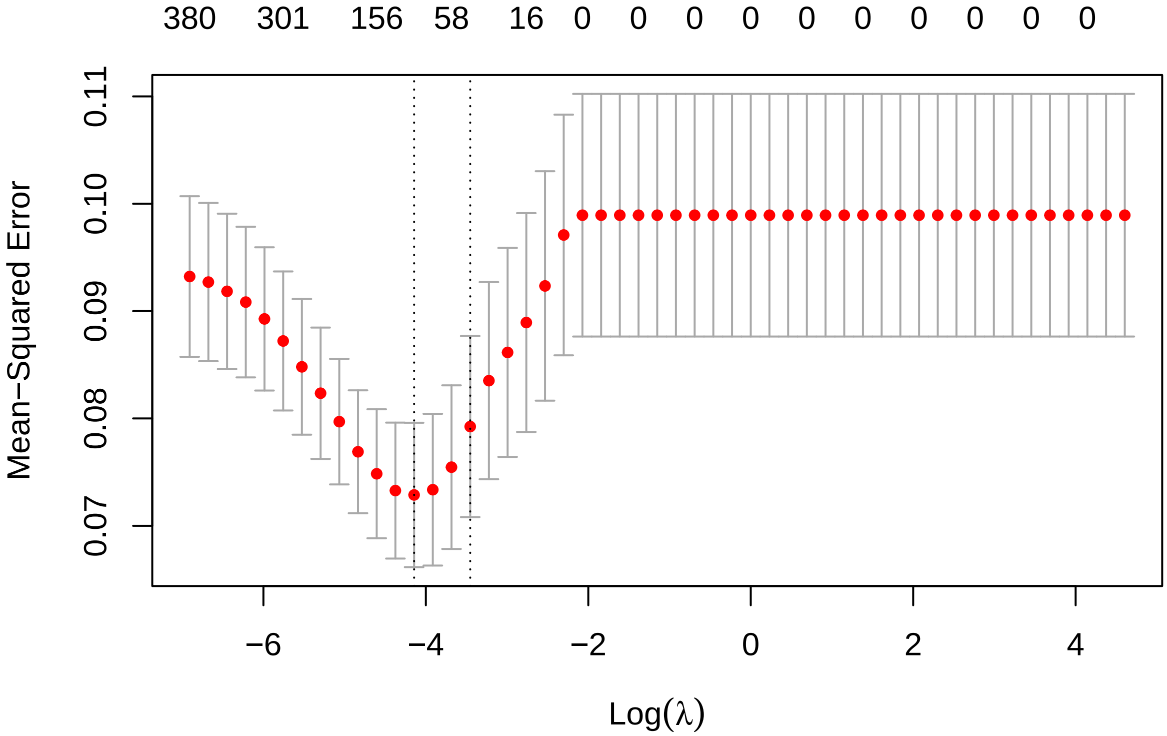


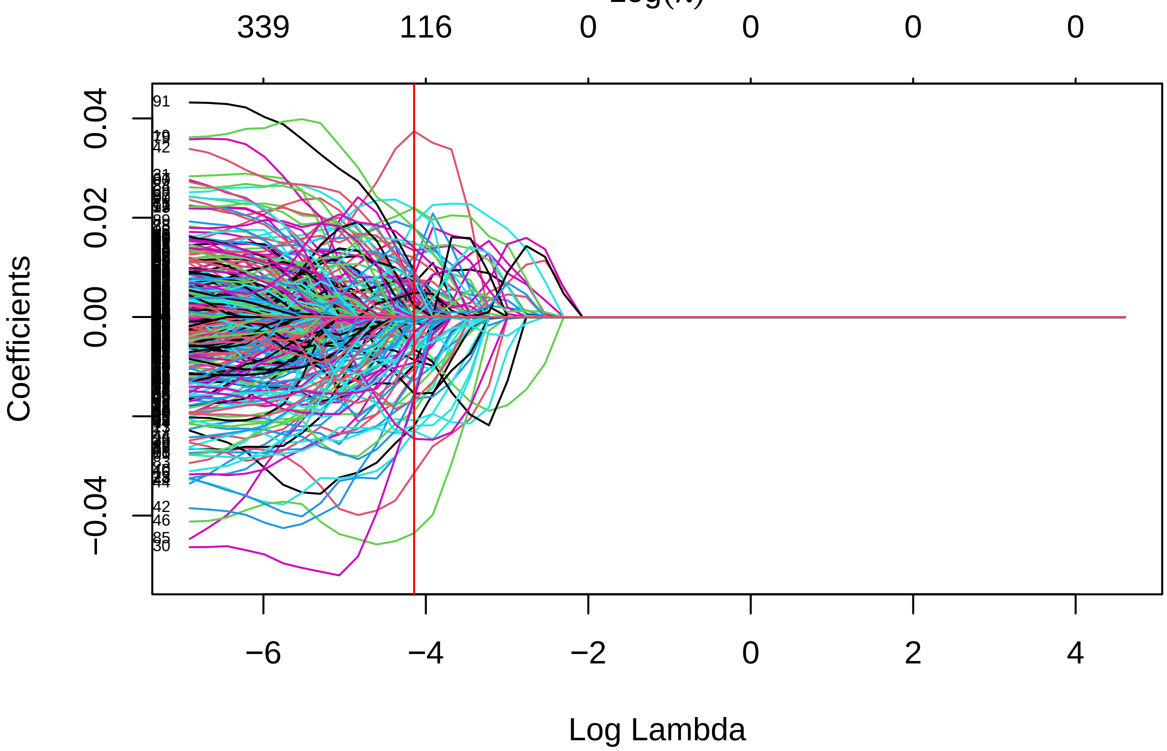

```
# Utils
# Calcule le meilleur lambda issu d'une régression Lasso (alpha = 1)
findBestLambda <- function(inputs,outputs,alphaPen) {
  lambdas <- 10^seq(2, -3, by = -.1)

  #k-fold cross validation to determine the best lambda
  reg <- cv.glmnet(inputs, outputs, alpha = alphaPen, lambda = lambdas,
                   standardize = FALSE, nfolds = 15, type.measure="mse")
  plot(reg)
  plot(reg$glmnet.fit, "lambda", label=TRUE)


  # Best lambda obtained from cross-validation
  lambda_best <- reg$lambda.min; #or 1se
  abline(v=log(lambda_best),col="red")
  text(log(lambda_best)+0.5 ,20, "log(lambda_best)", srt=0.2, col = "red",pos=4)
  lambda_best
}

makePredictionsPen <- function(model,testInputs,bestLambda) {
```

```r
  predictions <- predict.glmnet(model, s = bestLambda, newx = testInputs)
  predictions
}

# Calcule les métriques importantes : R2, RSE et MAE
computeMetrics <- function(predictions,testOutputs,modelName) {
  metrics <- data.frame(
    Model = modelName,
    R2 = R2(predictions, testOutputs),
    RMSE = RMSE(predictions, testOutputs),
    MAE = MAE(predictions, testOutputs)
    )
  colnames(metrics) <- c('Model Name', 'R2','RMSE','MAE')
  metrics
}
```

```r
# Régression LASSO (alpha = 1 dans glmnet)
processPenalization <- function(X_train, Y_train, X_test, Y_test, alphaPen) {
  lambda_best <- findBestLambda(X_train,Y_train,alphaPen)
  modpen <- glmnet(X_train, Y_train, alpha = alphaPen, lambda = lambda_best, standardize = FALSE)

  pred_train <- makePredictionsPen(modpen, X_train, lambda_best)
  pred_test <- makePredictionsPen(modpen, X_test, lambda_best)
  length(pred_test)
  text_train = ""
  text_test = ""
  if (alphaPen == 1) {
    text_train = "Lasso (on train)"
    text_test = "Lasso (on test)"
  }
  else if (alphaPen == 0) {
    text_train = "Ridge (on train)"
    text_test = "Ridge (on test)"
  }
  else {
    text_train = "Elastic-Net (on train)"
    text_test = "Elastic-Net (on test)"
  }
  res_train <- computeMetrics(pred_train,Y_train,text_train)
  res_test <- computeMetrics(pred_test,Y_test,text_test)

  coef(modpen)

  print(rbind(res_train, res_test))
  return(c(pred_train, pred_test))
}
```

```r
processPenalization(X_train, Y1_train, X_test, Y1_test, alphaPen = 1)
```





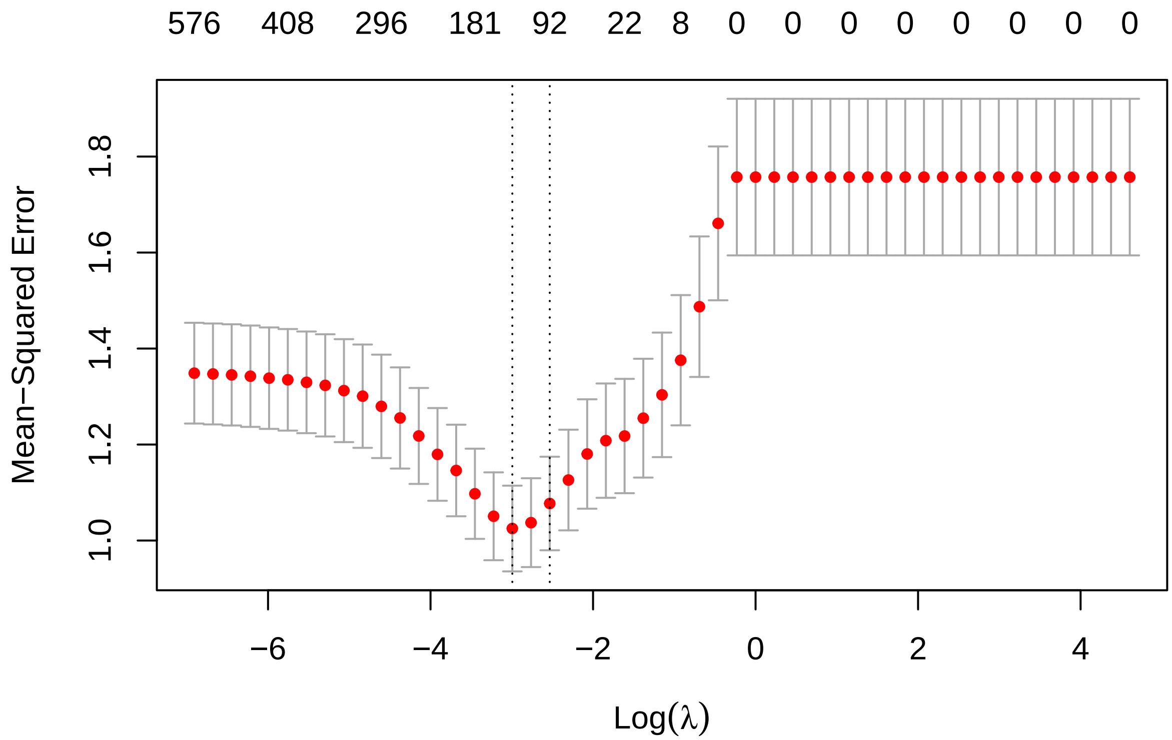

```
##         Model Name        R2      RMSE       MAE
## 1 Lasso (on train) 0.7484788 0.1769063 0.1380839
## 2  Lasso (on test) 0.1240237 0.3024971 0.2340818
```

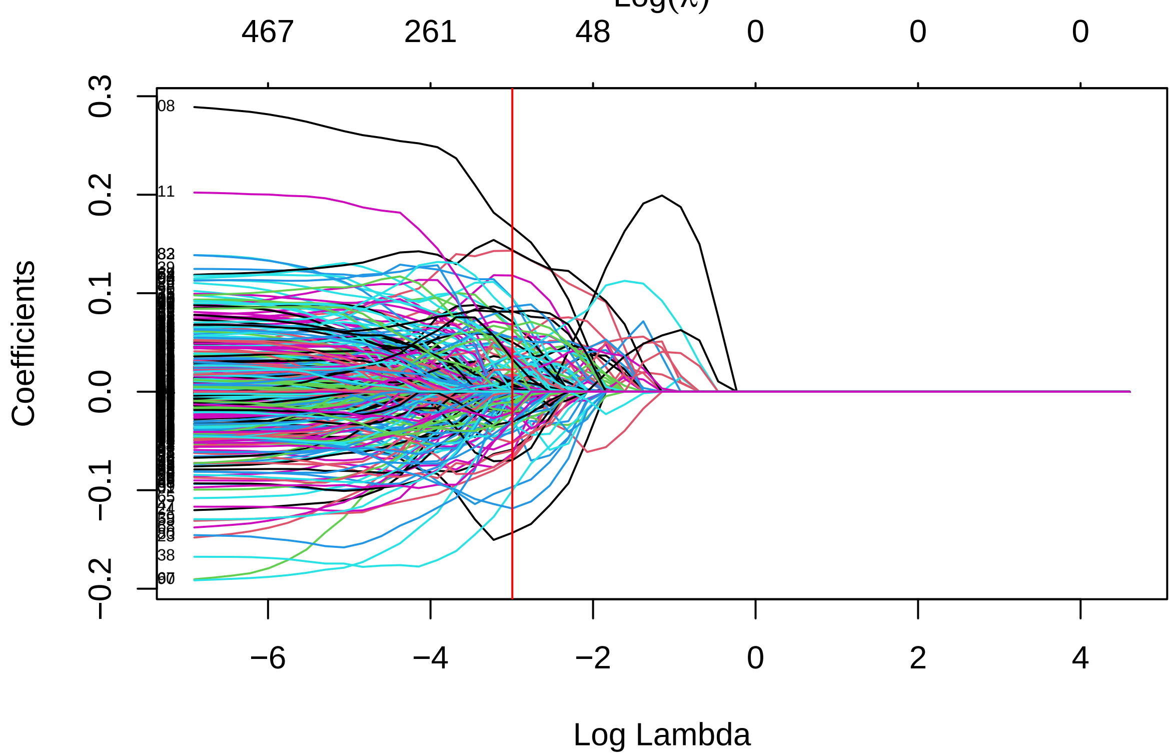
```
##   [1] 5.165940 4.671945 5.282544 5.261381 5.156861 4.656456 4.429130 4.794339
##   [9] 5.001301 4.971392 4.879863 4.608395 4.967465 4.669079 5.202363 5.183087
##  [17] 4.917289 4.959059 4.974603 4.915096 4.801273 4.946196 4.969583 4.430612
##  [25] 4.484558 4.759796 4.992068 4.788728 5.023791 4.618671 4.808880 4.735849
```

```
##  [33] 4.678018 4.846692 4.442719 4.737325 4.841976 5.129885 5.032283 4.779067
##  [41] 4.963961 4.840137 4.892735 4.915293 5.054170 4.839636 5.023933 5.058246
##  [49] 4.889880 4.914920 4.613514 4.716216 5.145660 4.930167 4.745069 5.015662
##  [57] 5.196817 5.107122 4.932255 4.764276 4.678116 4.932653 4.921824 4.856060
##  [65] 4.934049 4.942240 4.878696 4.841589 5.123378 4.820731 4.989234 4.853538
##  [73] 4.771073 4.982268 4.524492 4.645100 4.908123 4.864671 4.668502 5.027123
##  [81] 5.031399 4.925452 4.874112 4.855063 4.719701 4.903951 5.042975 4.862072
##  [89] 4.963987 5.219492 4.885740 4.836547 4.746213 4.369857 5.173686 4.938684
##  [97] 4.871850 4.480877 4.805564 5.244693 5.187821 4.925754 4.819663 4.908971
## [105] 4.888866 4.970726 4.869298 4.933283 4.609145 4.698342 4.996877 5.177113
## [113] 4.822510 4.737511 4.875636 4.716890 5.050765 5.005615 5.088512 4.969241
## [121] 5.192002 4.880960 4.912608 5.058565 4.981175 4.932973 4.920260 5.123924
## [129] 5.259389 4.771473 5.154618 4.659235 4.169670 5.158600 4.784354 4.907259
## [137] 5.114076 4.665985 4.984006 4.777676 4.741975 4.743028 4.833732 4.923854
## [145] 4.901243 4.739452 4.966500 5.005910 5.189640 4.954917 5.123090 4.877162
## [153] 4.394462 4.529016 4.941144 4.559914 4.900331 4.860071 4.714590 4.761878
## [161] 4.697510 4.900266 4.752469 4.908529 4.981720 4.907625 4.408214 4.883606
## [169] 4.276148 4.962263 4.732538 4.852743 4.642130 4.660169 4.978742 4.782764
## [177] 4.583488 4.916231 4.710052 4.890085 4.591290 4.511069 4.820941 4.635358
## [185] 4.767353 4.498850 4.962627 4.723741 5.094774 4.969495 4.770370 4.428575
## [193] 4.910454 4.906652 4.933853 4.964950 4.278875 4.375610 4.519615 4.698038
## [201] 4.745239 4.788656 4.959126 5.067362 4.753360 4.906199 4.692959 4.618572
## [209] 4.961082 4.693260 4.465380 4.962653 4.943523 4.914629 4.804813 4.611510
## [217] 5.028188 4.905219 5.111382 4.967168 4.958069 5.080073 4.862990 4.754546
## [225] 4.612155 4.925957 4.746341 4.893886 4.933055 4.717979 5.005747 4.883834
## [233] 4.575243 4.733297 4.997526 4.754880 4.951058 5.059457 4.915855 4.859185
## [241] 4.856840 5.077512 5.222023 4.946697 4.851425 4.935265 4.835191 4.774284
## [249] 4.594476 4.529142 4.944417 4.889074 4.958725 4.633454 5.000237 4.713455
## [257] 4.940016 4.943535 5.146968 4.710119 5.137149 4.920582 5.005582 4.855890
## [265] 4.921826 4.858935 5.032930 5.025721 5.097147 5.081778 4.818283 5.129710
## [273] 4.687140 4.999277 4.902737 4.951608 4.998039 4.671808 4.749559 4.936223
## [281] 4.931669 4.903314 4.892362 4.811733 4.777514 4.868094 4.810022 4.850804
## [289] 5.130479 4.745054 4.748338 4.838638 4.923675 5.062490 4.675986 4.660579
## [297] 4.891491 4.797926 4.905544 4.951580 4.695999 4.934091 4.625600 5.122379
## [305] 4.108097 4.832304 5.042883 4.939255 4.901613 4.932923 5.001524 5.026298
## [313] 4.899728 5.134371 4.945451 4.840646 4.586947 4.960514 4.790551 4.714561
## [321] 4.847583 4.809113 4.827694 5.067270 4.917631 4.631267 4.650433 4.791815
## [329] 4.935498 4.832052 4.497929 4.689043 4.729415 4.912189 4.825900 4.976974
## [337] 4.965620 4.958818 4.611895 4.929373 4.798468 4.936080 5.048927 4.960607
## [345] 5.080521 4.999760 4.924493 5.066242 4.938309 5.178841 4.679973 4.918425
## [353] 4.866686 4.778130 5.076129 4.919232 4.398082 4.602523 5.009426 5.038499
## [361] 4.873083 4.791925 4.560866 4.986316 4.971310 4.899541 5.012144 4.494469
## [369] 4.797512 4.827939 4.726285 5.048889 5.152459 4.876045 4.933213 5.028193
## [377] 4.647653 4.655604 5.029606 4.582035 4.848490 4.828155 4.957152 4.951501
## [385] 4.768208 4.844290 4.922121 5.000516 4.985048 4.976457 5.054407 4.805959
## [393] 4.908039 4.844285 4.804319 4.840824 4.624661 4.807461 4.851752 4.979384
## [401] 4.872977 4.930700 5.160161 5.185675 4.914412 4.686541 5.006312 4.746588
## [409] 4.930435 4.994679 4.798901 4.790677 4.769491
```

```r
Y2 = pheno.df$Straighthead.suseptability
Y2[is.na(Y2)] = mean(Y2, na.rm=TRUE)
Y2_train = Y2[random_sample]
Y2_test = Y2[-random_sample]
```

```r
processPenalization(X_train, Y2_train, X_test, Y2_test, alpha = 1)
```





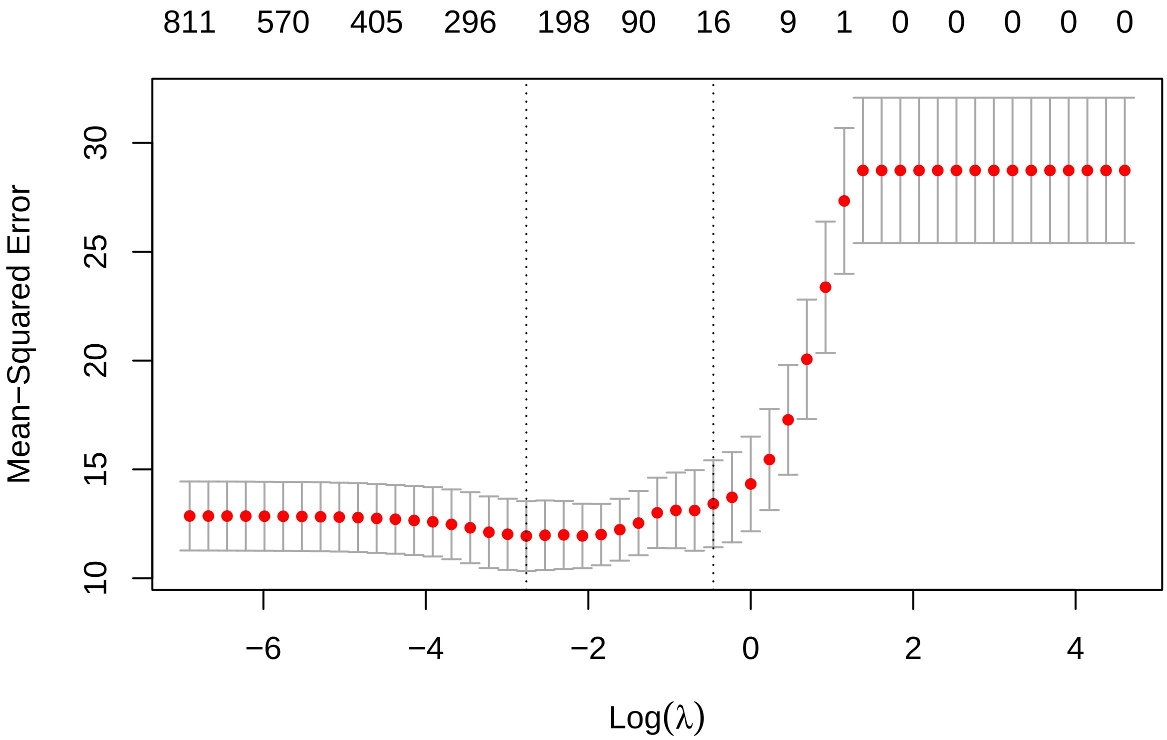

```
##         Model Name        R2      RMSE       MAE
## 1 Lasso (on train) 0.8202784 0.6077878 0.4746704
## 2  Lasso (on test) 0.4387308 0.8865810 0.6741890
```

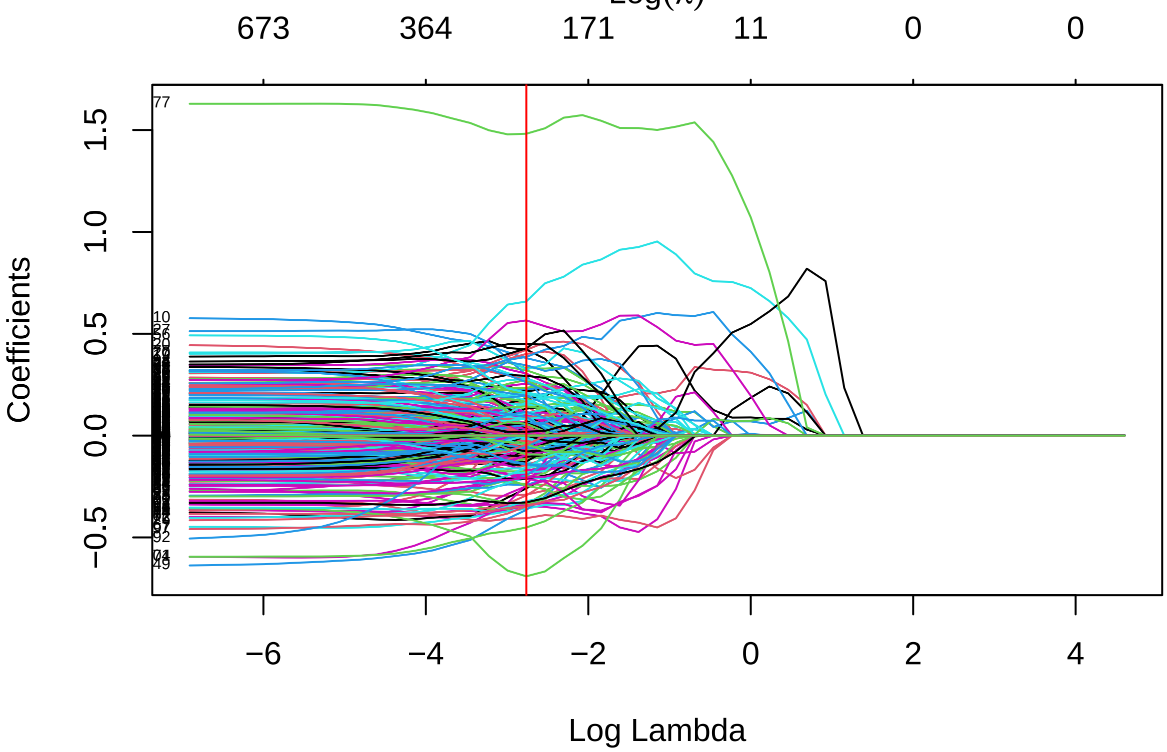
```
##   [1] 7.629295 8.104112 7.754788 7.879113 7.448450 7.201397 5.501333 8.611833
##   [9] 7.381283 7.106920 6.192088 7.936326 7.810007 6.984990 6.713812 6.576304
```

```
##  [17] 7.653476 8.216624 7.716089 6.804034 6.249390 6.288988 7.482934 8.855075
##  [25] 8.522792 8.393149 6.766369 7.550849 7.995873 7.853574 7.650215 7.145010
##  [33] 7.314400 5.565086 6.856283 4.734888 8.274362 7.209236 6.033836 6.517260
##  [41] 7.681327 8.145960 7.514836 7.362256 8.129135 6.783580 7.808237 7.216043
##  [49] 8.237871 5.374365 7.907379 5.413012 7.045548 6.979988 7.068637 6.837612
##  [57] 8.269868 7.586485 8.493679 7.588357 5.668250 6.369762 8.518521 4.497620
##  [65] 7.579161 4.867118 7.463600 5.893944 7.802785 7.474602 8.000153 7.299880
##  [73] 7.465042 7.046953 6.382538 6.821104 8.225907 8.106179 5.746036 8.085414
##  [81] 7.915060 6.346003 6.897764 8.238550 8.419693 7.328571 6.528919 8.199489
##  [89] 7.949185 8.284254 5.149757 7.444070 7.546341 8.547882 7.456473 6.362823
##  [97] 8.460601 6.278914 8.087675 6.929681 7.941992 7.172072 6.032774 7.602823
## [105] 7.593301 7.828813 5.429099 6.222476 7.560763 8.551735 7.628958 6.883120
## [113] 8.327943 8.611061 6.145016 6.087245 7.188182 7.671455 7.002604 5.004154
## [121] 7.455776 6.102538 6.292923 6.794466 8.091730 7.833768 5.448453 6.399405
## [129] 8.260999 8.240994 7.823481 7.770996 5.749220 6.812752 6.767095 7.449658
## [137] 7.443360 7.441810 7.747683 7.969909 5.671705 8.540609 8.060119 7.358253
## [145] 5.033162 6.155805 7.784059 6.965940 7.681582 7.714356 6.088390 5.348122
## [153] 6.380429 7.546600 7.455326 6.227388 5.910346 6.458822 5.820939 6.111185
## [161] 8.005602 6.687984 6.000340 6.594773 8.192322 7.249413 6.622344 5.070568
## [169] 5.921387 7.333112 7.177954 7.079885 7.324319 5.750434 6.717910 7.207637
## [177] 7.979922 7.344849 5.336895 5.297882 5.455055 6.735311 6.756182 5.843436
## [185] 5.540418 7.691520 6.579806 6.117754 7.187091 5.498909 6.047093 6.359447
## [193] 5.931474 5.337417 6.534829 8.117020 6.353076 6.364771 6.145030 5.348729
## [201] 5.898366 7.108031 5.450103 7.341450 6.178474 6.313661 6.069292 6.964876
## [209] 5.443267 6.196095 5.857597 7.594136 7.556903 5.625116 6.120712 7.169557
## [217] 7.274712 7.612522 7.217219 7.260451 7.147113 6.924302 7.475490 7.108247
## [225] 5.972624 5.150950 6.999695 7.137143 7.892998 8.085525 8.114795 5.970894
## [233] 7.896374 7.000089 7.821162 6.492876 8.364385 6.367919 6.810373 5.728866
## [241] 8.022622 7.337414 7.022535 7.088528 6.974200 6.240289 6.154902 5.972139
## [249] 6.707944 7.852313 7.044204 7.580889 7.559905 5.811695 7.646970 7.218587
## [257] 7.815793 6.046096 7.236341 8.364149 6.992481 5.350644 7.441857 7.634546
## [265] 6.607533 7.744383 5.800408 7.587225 5.932139 6.619184 4.116771 6.359882
## [273] 4.350033 7.157374 7.281271 7.465348 7.907748 6.514871 7.625396 6.142039
## [281] 6.914387 7.440975 5.626282 6.832369 6.705185 7.156972 7.665154 8.077627
## [289] 7.331726 6.352074 6.602576 6.505664 7.511877 5.731353 6.692300 5.980040
## [297] 5.768520 5.675022 6.394292 6.899265 5.388646 7.851779 7.657292 7.772334
## [305] 5.680409 4.078210 6.363805 6.943768 6.128578 7.384276 7.105253 5.414674
## [313] 6.445445 6.957166 7.013799 8.101953 6.969948 6.962341 7.822454 6.452173
## [321] 5.558170 7.157382 7.225580 4.618034 4.554085 8.260551 7.951622 5.760563
## [329] 5.458055 8.444757 6.565202 7.509648 5.953194 5.449862 7.180507 7.796836
## [337] 7.946357 7.294526 7.966349 7.096848 7.681665 5.261282 8.131855 7.559446
## [345] 7.820086 7.516147 7.141689 6.322003 6.901747 6.036378 7.617111 8.111855
## [353] 8.135892 5.805699 7.315853 7.859668 6.234616 7.566938 7.630809 6.869696
## [361] 5.661874 6.528566 7.421976 6.511596 7.738843 7.319221 7.871478 6.101400
## [369] 7.569614 6.623290 6.031562 7.697933 6.929961 5.096603 7.027866 7.231154
## [377] 6.960730 7.272952 7.148383 6.898755 7.165968 7.087479 8.092915 7.631338
## [385] 6.081395 6.033920 7.771746 7.909727 6.719353 7.569491 5.965714 6.822972
## [393] 5.634779 7.229498 7.705357 6.516097 5.999131 6.389498 6.626338 7.827590
## [401] 6.784335 6.906733 5.462490 5.879159 7.270019 7.389466 7.541452 7.095696
## [409] 6.481578 5.611647 5.762168 5.769104 5.125988
```

```r
# Lasso
Y3 = pheno.df$Amylose.content
Y3[is.na(Y3)] = mean(Y3, na.rm=TRUE)
```

```
Y3_train = Y3[random_sample]
Y3_test = Y3[-random_sample]
processPenalization(X_train, Y3_train, X_test, Y3_test, alphaPen = 1)
```



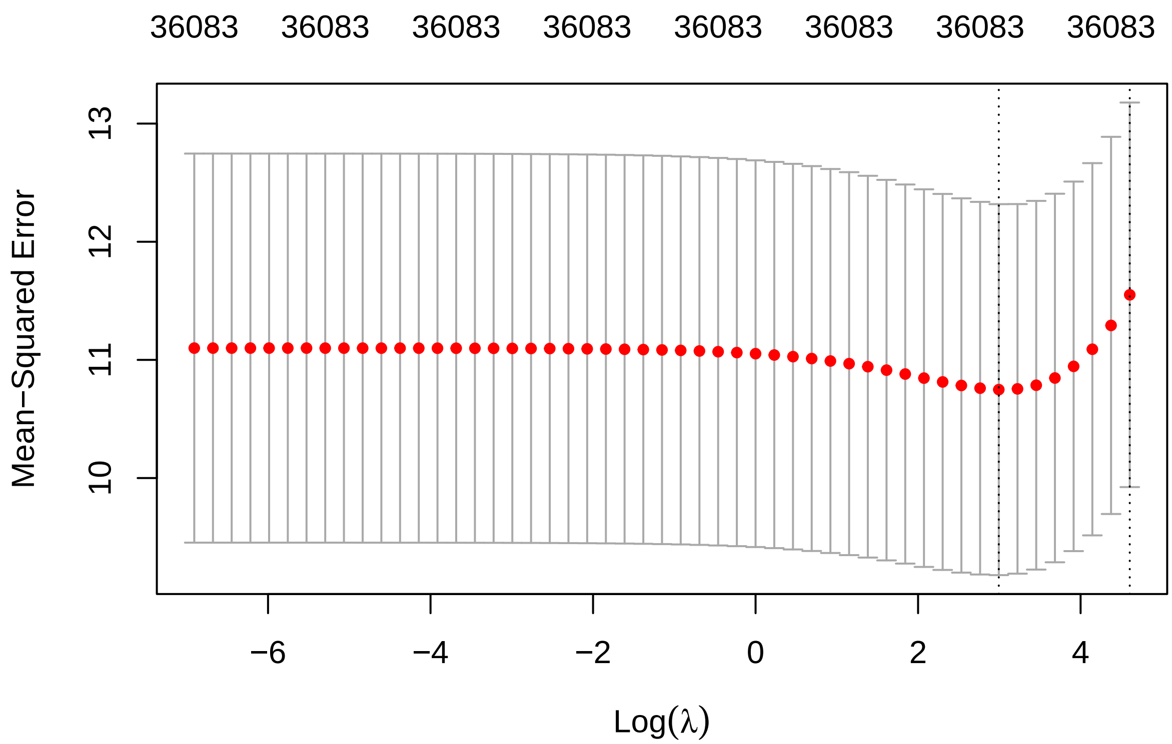


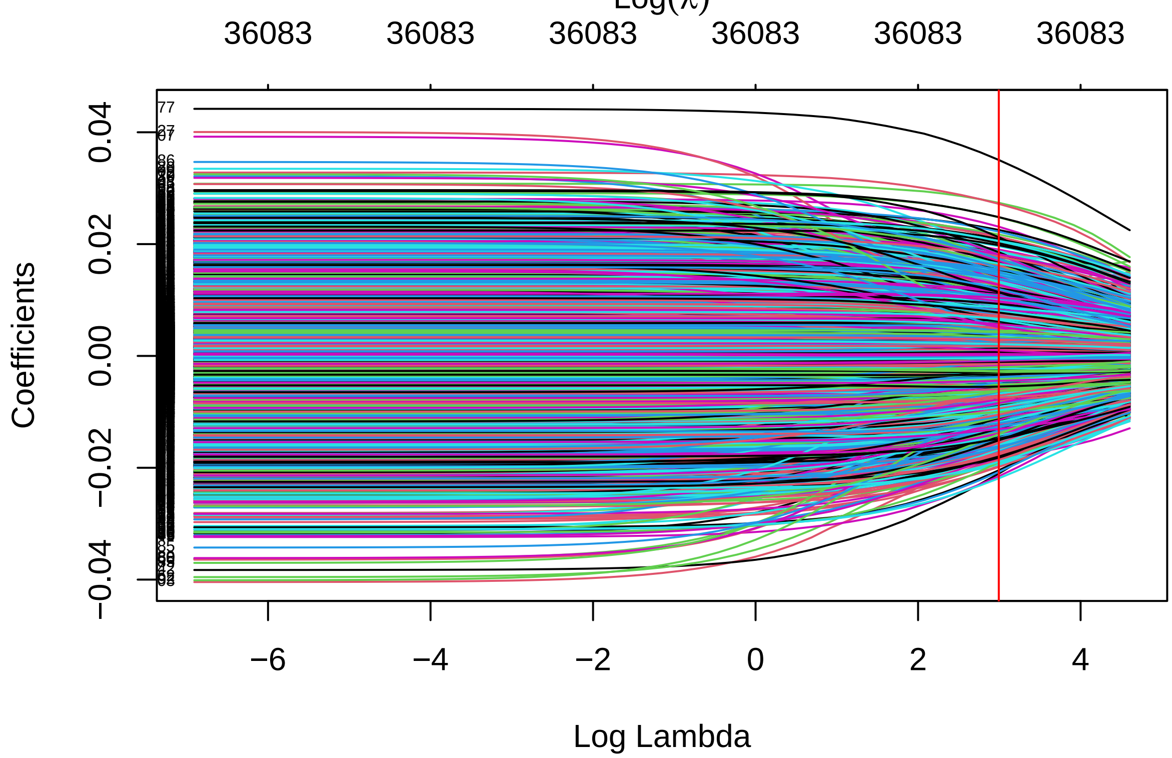

```
##          Model Name        R2      RMSE       MAE
## 1 Lasso (on train) 0.9723593 0.9795942 0.7071624
## 2  Lasso (on test) 0.6596715 3.2218230 2.2358588
```

```
##   [1] 23.206285 20.051602 22.781196 20.959234 21.348761 21.674691 15.149160
##   [8] 22.355843 25.423054 18.897630 17.564708 20.682331 14.746267 25.852838
##  [15] 17.214781 22.557898 21.799848 21.063477 20.443881 13.461612 19.105456
##  [22] 17.583708 22.863008 22.159203 23.964842 22.919972  2.787933 21.746873
##  [29] 24.484702 18.985224 20.924650 26.367677 24.254245 18.171481 17.580450
##  [36] 12.607596 23.934561 18.978006 19.186050  5.616963 21.040510 23.666263
##  [43] 19.906135 23.535068 22.980022 14.163333 24.331076 20.031881 24.426401
##  [50] 11.699117 25.398324 12.028526 20.705010 26.091880  5.347217  2.296498
##  [57] 24.456700 21.435553 18.813833 19.603000 15.174099 11.993849 22.634217
##  [64] 24.266933 21.742510 20.868388 20.504543 17.665941 24.909752 19.768976
##  [71] 23.384342 24.987741 24.537348 19.976379 16.099945 20.706739 20.047432
##  [78]  3.389795 13.921527 22.613521 20.932206 16.217402 21.762079 24.394327
##  [85] 22.367841 24.831398 15.600842 24.418237 26.558618 26.026102 15.089227
##  [92] 20.295830 22.531927 24.534046 21.414386 12.965978 22.873963 16.179740
##  [99] 21.209770 21.899428 22.929419 15.006422 15.389932 26.286356 27.215655
## [106] 23.951709 18.741936 19.251058 19.348178 24.994722 26.022718 20.272337
## [113] 26.971931 27.185075 24.778917 17.089300 22.419433 20.111009 21.799889
## [120] 13.724242 26.836450 21.337067 17.385228 17.088614 21.810067 21.254163
## [127] 17.013801 19.772101 23.281131 23.124644 22.545957 19.334969 18.420710
## [134] 20.907341 14.267805 21.158560 19.257291 27.075499 22.824213 22.516973
## [141] 14.397469 26.795305 23.679707 24.066484 16.661253  1.627419 19.703667
## [148] 14.339714 21.976025 21.514887 13.113266 12.793098 17.848408 16.614063
## [155] 19.595344 17.276865 13.655872 12.599633 15.076649 22.824996 22.905914
## [162]  2.559962 12.667479 13.321317 24.941688 20.324862 18.907965 14.675520
## [169] 16.090404 20.458665 25.894509 24.576915 25.621108 17.953860 21.748889
## [176] 26.912041 24.780592 24.749077 15.032898 13.041796 14.584301 17.879833
## [183] 16.527496 13.737876 15.364945 25.492766 15.830131 15.765014 25.053770
## [190] 17.142389 14.785299 17.942023 15.706202 15.407959 21.109441 15.438918
## [197] 15.791069 19.328563 18.178558 13.946347 14.074288 26.136088 16.425342
## [204] 25.464528 23.072894 19.095239 16.053665 15.480099 15.720907 14.096396
## [211] 16.908016 19.889205 18.888271 11.458986 15.220411 24.856682 25.482299
## [218] 25.319903 23.897183 24.046553 25.055851 24.094840 23.049286 25.066920
## [225] 14.096939 17.561614 11.442666 24.202217 22.784719 22.858898 25.190181
## [232] 15.756973 20.279964 24.588195 21.176947 24.378710 23.425719 18.253347
## [239] 24.932585 14.455196 22.912593 23.829916 20.139501 24.649731 24.410435
## [246] 17.631280 12.258062  1.979075 12.695181 25.465885 23.728988 25.207185
## [253] 20.710534 18.633218 20.917907 23.710143 22.641230 17.281999 21.119231
## [260] 23.790250 16.349770 10.008712 15.764650 14.774355 16.504824 20.977457
## [267] 23.191838 17.377861 21.719052 24.870266 21.508867 23.145281 21.409774
## [274] 22.865346 24.129641 22.434787 23.315530 18.072934 20.147205 18.221765
## [281] 17.060014 22.728634 23.955624 25.105470 21.352055 24.325252 19.975244
## [288] 19.000924 22.840492 17.151357  4.155307 20.270100 21.004931 22.880753
## [295] 19.002146 15.335964 15.494669 15.796904 15.130749 25.128116 18.023970
## [302] 16.350098 21.704174 18.324361 21.928504 20.537067 21.838468 25.230930
## [309] 21.096801 19.727666 24.378667 21.017235 17.977273 16.580467 23.939128
## [316]  3.082767 20.984105 25.200489 22.560257 24.066424 21.766185 18.956892
## [323] 23.615000 22.879514 15.898978 21.167861 22.128108 16.499339 15.941663
## [330] 24.228423 17.995579 23.168286 18.532868 14.961828 24.097709 20.608714
## [337] 20.318959 26.300812 20.547168 21.952108 24.194834 16.869237 21.336262
## [344] 21.873947 23.655730 27.716115 27.636026 16.048516 17.830154 14.411701
## [351] 24.682819 22.683924 20.717011 17.643298 24.541587 25.888569 13.982000
## [358] 15.233731 21.187246 19.031731 15.304215 18.841078 22.552054 22.872714
## [365] 21.187208 20.715578 24.207385 19.114584 21.856699 26.831102 15.461412
## [372] 24.782724 17.272339 15.716775 24.769968 24.826596 25.027424 26.847459
```

```
## [379] 23.460788 24.678691 23.960170 25.115802 21.981676 25.646677 13.261967
## [386] 17.831928 25.117224 19.316324 18.814512 18.944320 16.484147 25.560334
## [393]  9.576147 23.205456 24.623551 11.343139 14.914516 13.567818 25.246516
## [400] 19.861805 20.578344 25.573539 19.884213 19.065494 23.396623 24.136839
## [407] 19.496557 25.233114 13.788827 19.060415 13.759263 12.252934 11.250860
```

```
# Ridge
processPenalization(X_train, Y3_train, X_test, Y3_test, alphaPen = 0)
```



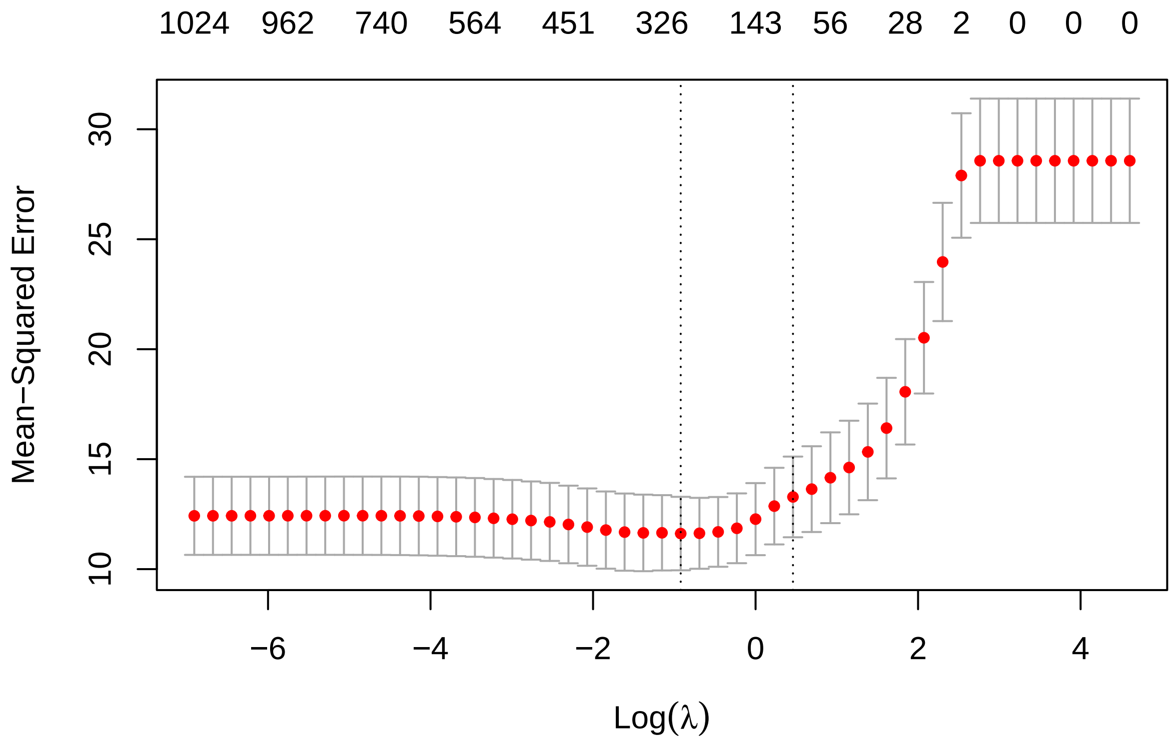


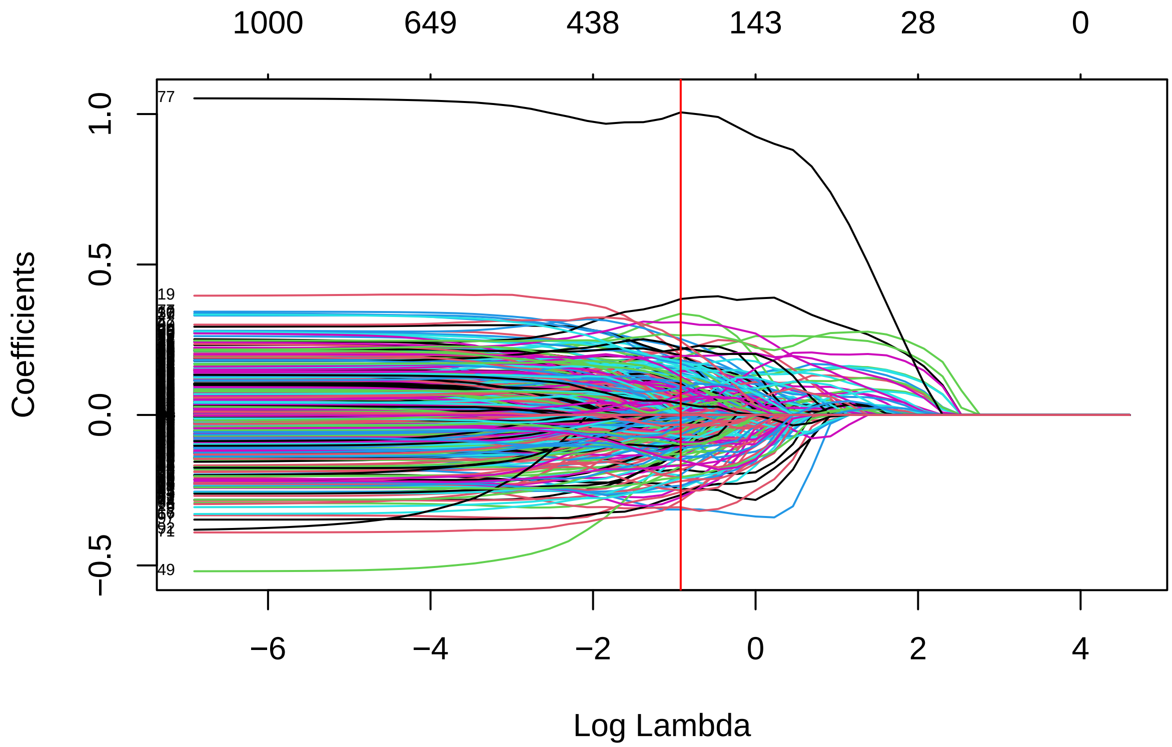
```
##         Model Name        R2      RMSE       MAE
## 1 Ridge (on train) 0.9829524 0.7857659 0.5202778
## 2  Ridge (on test) 0.7184563 2.9889851 1.8828471
```

```
##   [1] 23.306460 19.514892 23.271931 20.529744 21.969894 21.770970 15.336479
##   [8] 22.819981 25.427071 18.476782 17.038522 21.096204 14.333817 26.133808
##  [15] 17.474308 22.278076 22.413297 21.093182 19.904915 13.805769 18.460835
##  [22] 16.713778 22.793127 21.631417 24.097371 22.903628  2.009274 21.541796
##  [29] 24.529950 19.256190 20.940350 26.112637 24.290383 18.008175 17.065323
##  [36] 13.723174 23.852762 18.947001 18.420362  5.975100 20.928329 23.825912
##  [43] 19.688733 23.909963 23.054073 14.500098 24.392885 19.708019 24.535153
##  [50] 11.914493 25.351435 12.218268 20.701807 26.396340  3.825422  1.746587
##  [57] 24.585159 21.988619 18.601814 18.741020 15.337869 12.151067 23.379264
##  [64] 25.399208 22.096452 20.901884 21.273570 17.408597 24.591169 20.159012
##  [71] 22.876392 24.597391 24.709519 19.209853 16.058958 20.757162 19.904518
##  [78]  2.142520 13.914362 21.960863 19.994700 15.525819 22.078389 24.318295
##  [85] 23.896500 25.041194 16.313833 24.741444 26.932445 25.852967 14.975918
##  [92] 20.215847 22.636331 25.076055 21.648597 13.356537 23.286421 16.928850
##  [99] 20.264904 22.013059 22.879284 15.602941 16.166484 26.295644 27.160635
## [106] 23.880386 18.159176 18.622878 18.669678 24.803332 26.299973 20.085502
## [113] 27.450623 27.608665 24.972051 17.341406 22.532500 19.506206 21.448796
## [120] 13.504263 27.003693 19.355060 18.361390 18.158377 22.437263 21.322064
## [127] 16.950877 20.503678 22.813997 23.252853 22.547512 19.603432 18.520322
## [134] 21.524320 15.101888 20.248136 19.668505 27.466766 23.278643 22.891877
## [141] 14.816861 27.113696 22.808209 24.676495 16.478719  1.841630 20.181448
## [148] 13.724315 22.570426 21.717748 13.481045 12.995162 17.169365 16.660482
## [155] 18.889875 17.021147 13.909185 12.113378 13.845904 22.611860 23.274878
## [162]  2.379308 12.002170 13.413097 25.029166 20.145517 19.325474 15.311265
## [169] 16.067692 20.488314 25.154004 25.568175 25.496369 18.055914 22.236758
## [176] 27.146693 25.308732 25.049502 13.955399 13.476006 14.029375 18.112542
## [183] 16.154402 13.803709 15.671719 25.442470 16.361887 16.295423 25.007846
## [190] 16.789435 14.463658 18.598075 16.001825 15.195262 20.049518 14.922603
## [197] 16.359305 19.284138 17.170557 14.389508 14.452135 26.033843 16.675018
## [204] 25.973048 23.209983 18.725587 15.499804 15.520476 16.405840 14.754137
## [211] 16.664721 20.124500 18.971353 11.991716 15.514181 24.848916 25.164002
## [218] 25.020526 23.506086 24.786361 25.126663 24.283938 22.845562 25.276715
## [225] 14.386254 17.124663 10.764571 23.965286 23.169118 23.199437 25.576137
## [232] 15.186222 20.149449 24.192753 21.387495 23.790160 23.185740 17.932720
## [239] 25.082666 15.020589 23.496778 23.688487 20.292125 25.271122 24.594445
## [246] 18.259740 12.000490  1.167406 13.539059 25.081949 23.918571 24.932725
## [253] 20.584632 18.822647 20.714297 23.369138 22.802659 18.064663 21.320337
## [260] 23.565494 15.565316 10.177988 16.477820 14.825699 15.734743 20.877631
## [267] 24.233922 17.581097 21.576767 25.274582 21.856135 23.665484 20.592621
## [274] 22.580005 24.860216 20.857207 23.325500 17.990249 20.213657 17.906228
## [281] 17.864475 22.762472 23.022865 24.794573 21.332692 24.937011 20.073076
## [288] 19.375832 22.723138 16.456546  4.277103 19.680178 20.310084 21.851862
## [295] 18.911303 15.389912 15.375916 15.584225 16.127146 24.426239 17.424669
## [302] 16.667301 21.945060 17.575697 21.867272 19.982975 21.272127 25.613098
## [309] 21.807797 19.617261 24.513575 21.257616 17.896544 17.320011 23.480535
## [316]  1.667774 20.495332 25.445462 22.612516 24.005357 21.866535 18.268963
## [323] 24.170671 22.582747 15.964664 21.302027 22.335152 16.520816 16.359890
## [330] 24.099229 18.241280 23.316004 18.168998 15.994444 23.425370 20.355254
## [337] 23.333359 24.154199 21.216048 25.108880 22.600509 15.854807 20.263060
## [344] 21.356002 22.146438 25.363587 25.583896 16.705734 18.324572 16.733863
```

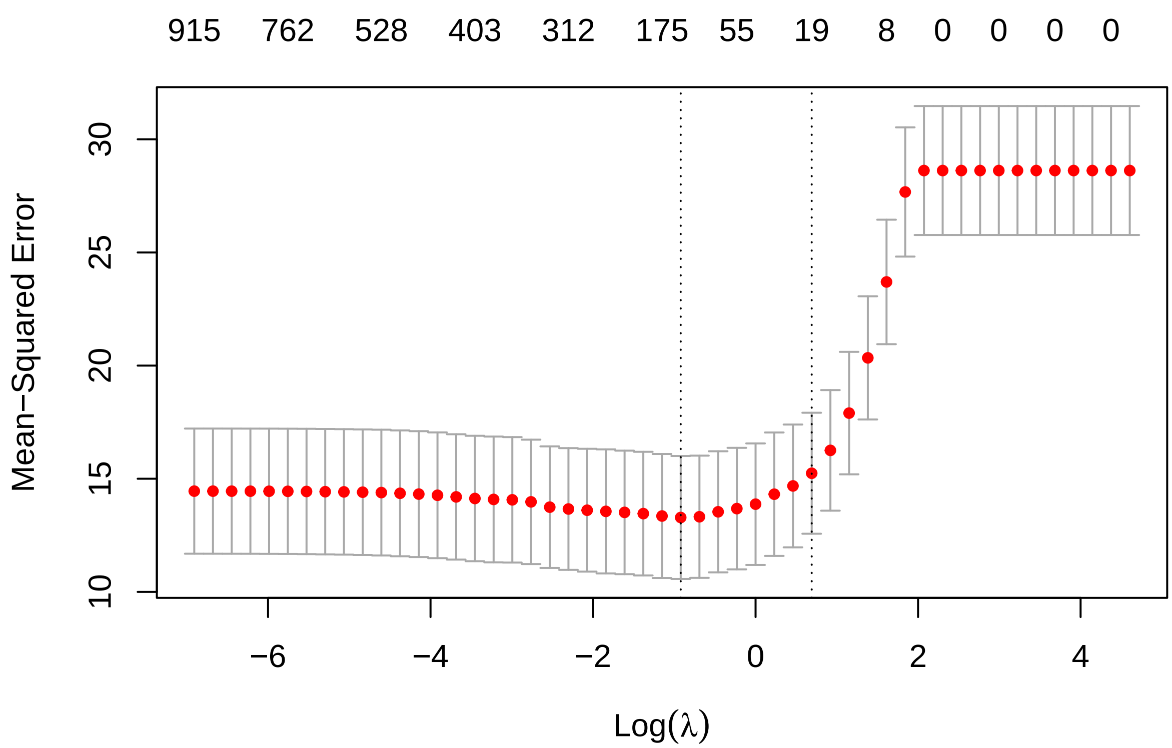
```
## [351] 22.344736 24.473275 23.252238 17.278523 22.868187 25.190874 16.394347
## [358] 17.244957 21.576711 20.580536 15.023352 17.302359 22.423533 19.690878
## [365] 20.329681 20.976318 22.802479 18.353532 24.636705 25.067687 14.484211
## [372] 22.237503 17.660881 15.623731 23.825506 22.856436 23.610995 26.557472
## [379] 25.176593 24.795427 25.080010 24.439741 23.787123 25.694009 14.478270
## [386] 17.058319 25.334220 19.867490 18.303281 19.254367 16.029300 23.302198
## [393] 10.698656 24.477488 24.636131 12.041776 14.288399 13.403396 22.867476
## [400] 20.287190 21.006677 24.261385 21.293734 20.926095 23.617784 25.115507
## [407] 19.368723 26.300380 13.632241 17.032976 14.026366 15.137094 10.613423
```

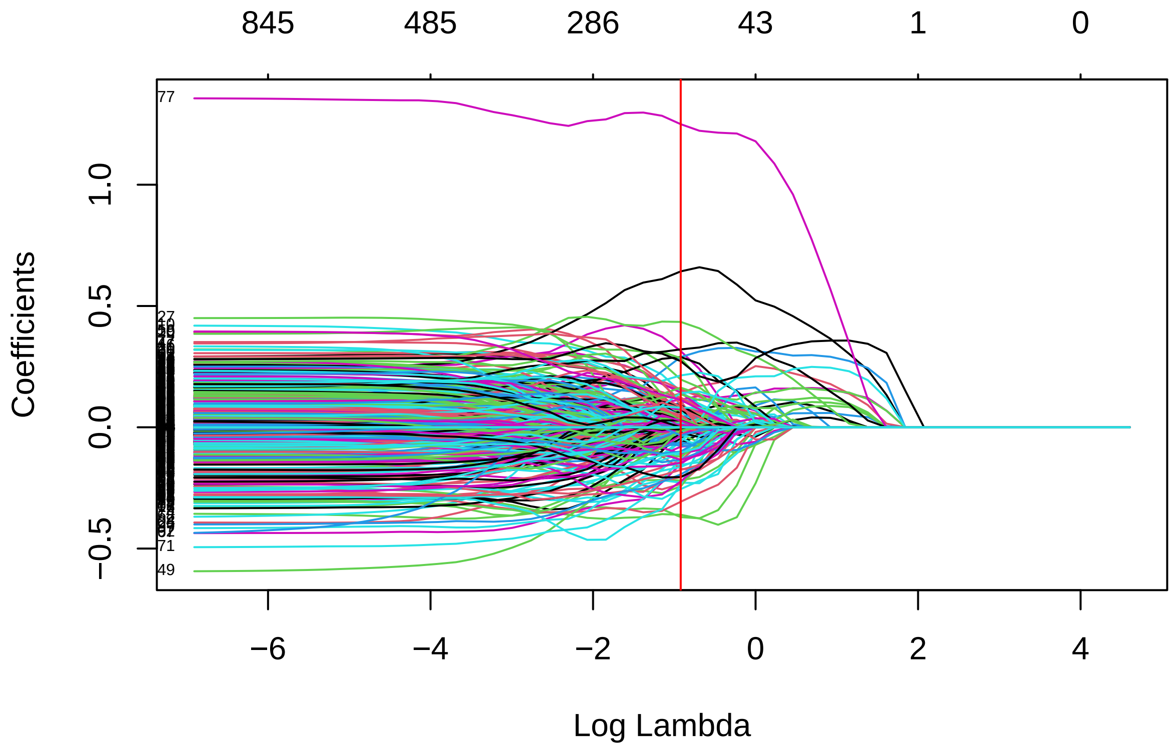
```
for (alpha in c(0.25,0.5,0.75))
  processPenalization(X_train, Y3_train, X_test, Y3_test, alphaPen = alpha)
```



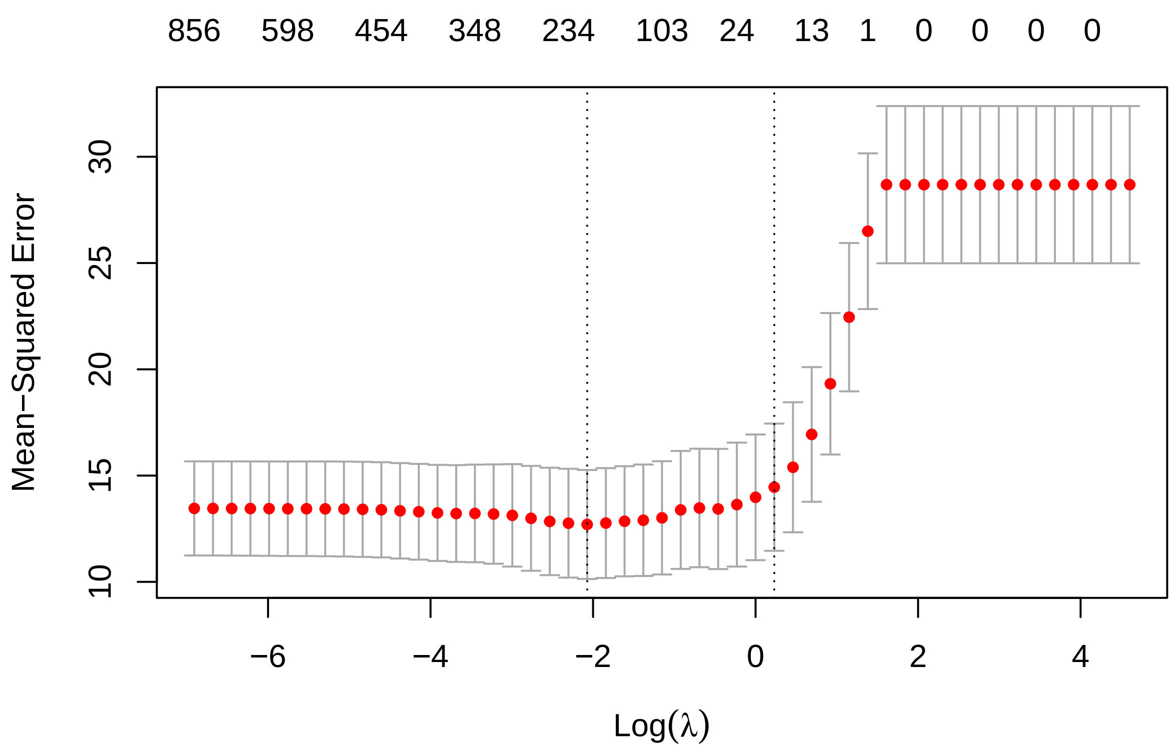


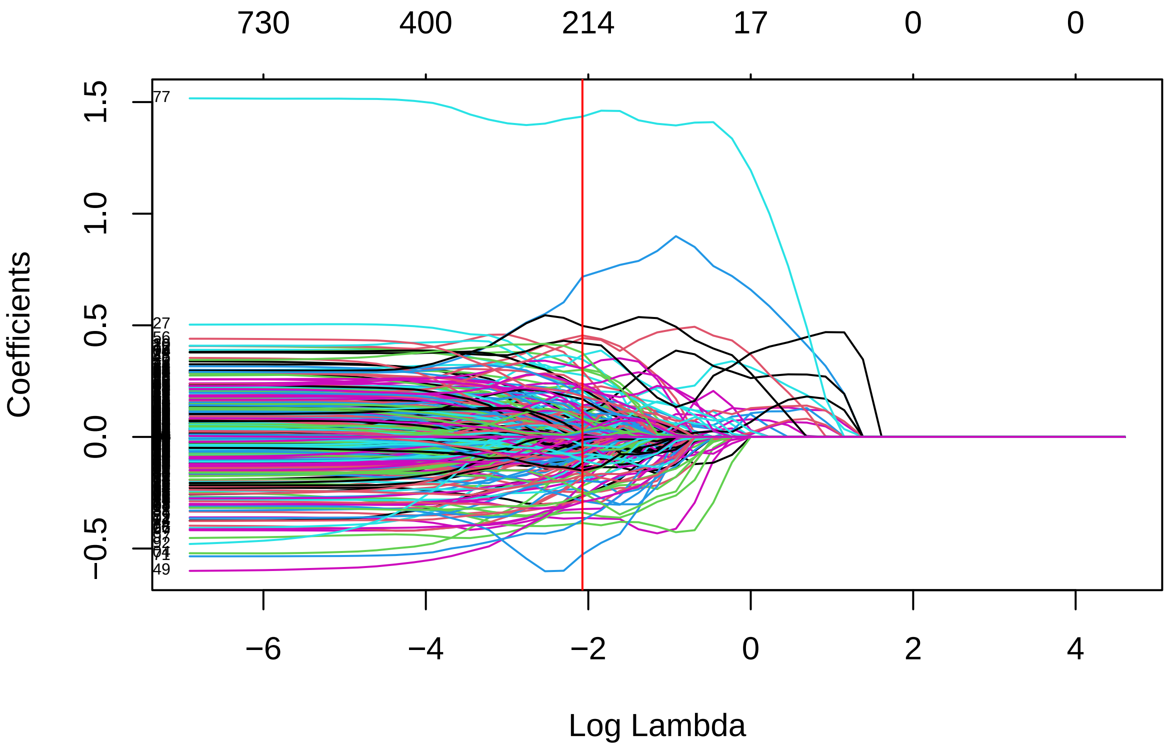

```
##                  Model Name        R2      RMSE       MAE
## 1 Elastic-Net (on train) 0.9407257 1.438213 1.024091
## 2  Elastic-Net (on test) 0.6723752 3.177460 2.148033
```






```
##                  Model Name        R2     RMSE      MAE
## 1 Elastic-Net (on train) 0.8461791 2.235796 1.535055
## 2  Elastic-Net (on test) 0.7239708 2.964531 1.952748
```






```
##                 Model Name        R2     RMSE       MAE
## 1 Elastic-Net (on train) 0.9478073 1.338034 0.9599673
## 2  Elastic-Net (on test) 0.6643932 3.202387 2.1496773
```