Reproduciblity LASSO and Bayesian LASSO Regression for Directional Response (d=2)

# Loading the R Package

library(RBVNF)

##   
## Attaching package: 'RBVNF'

## The following object is masked from 'package:base':  
##   
## norm

load\_packages()

## Loading required package: numDeriv

## Loading required package: MASS

## Loading required package: Rcpp

## Loading required package: RcppZiggurat

## Loading required package: RcppParallel

##   
## Attaching package: 'RcppParallel'

## The following object is masked from 'package:Rcpp':  
##   
## LdFlags

##   
## Rfast: 2.1.0

## \_\_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_ \_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_ \_\_   
## | \_\_ \_\_ \_\_ \_\_ | | \_\_ \_\_ \_\_ \_\_ \_/ / \ | \_\_ \_\_ \_\_ \_\_ / /\_\_ \_\_ \_ \_ \_\_ \_\_\   
## | | | | | | / \_ \ | | / /   
## | | | | | | / / \ \ | | / /   
## | | | | | | / / \ \ | | / /   
## | |\_\_ \_\_ \_\_ \_\_| | | |\_\_ \_\_ \_\_ \_\_ / / \ \ | |\_\_ \_\_ \_\_ \_\_ \_ / /\_\_/\   
## | \_\_ \_\_ \_\_ \_\_| | \_\_ \_\_ \_\_ \_\_| / /\_\_ \_ \_\_\ \ |\_ \_\_ \_\_ \_\_ \_ | / \_\_\_ /   
## | \ | | / \_ \_ \_ \_ \_ \_ \ | | \/ / /   
## | |\ \ | | / / \ \ | | / /   
## | | \ \ | | / / \ \ | | / /   
## | | \ \ | | / / \ \ | | / /   
## | | \ \\_\_ \_\_ \_ | | / / \ \ \_ \_\_ \_\_ \_\_ \_| | / /   
## |\_| \\_\_ \_\_ \_\_\ |\_| /\_/ \\_\ /\_ \_\_ \_\_ \_\_ \_\_\_| \/ team

## Loading required package: cowplot

##   
## Attaching package: 'mvtnorm'

## The following objects are masked from 'package:Rfast':  
##   
## Crossprod, dmvnorm, dmvt, rmvnorm, rmvt, Tcrossprod

## Loading required package: Matrix

## Loaded glmnet 4.1-8

load\_additional\_packages()

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:MASS':  
##   
## select

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

## The following object is masked from 'package:graphics':  
##   
## layout

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':  
##   
## combine

## The following object is masked from 'package:Rfast':  
##   
## nth

## The following object is masked from 'package:MASS':  
##   
## select

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

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

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.0  
## ✔ readr 2.1.4   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::combine() masks gridExtra::combine()  
## ✖ tidyr::expand() masks Matrix::expand()  
## ✖ dplyr::filter() masks plotly::filter(), stats::filter()  
## ✖ purrr::is\_integer() masks Rfast::is\_integer()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ dplyr::nth() masks Rfast::nth()  
## ✖ tidyr::pack() masks Matrix::pack()  
## ✖ dplyr::select() masks plotly::select(), MASS::select()  
## ✖ lubridate::stamp() masks cowplot::stamp()  
## ✖ purrr::transpose() masks Rfast::transpose()  
## ✖ tidyr::unpack() masks Matrix::unpack()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(glmnet)

## Simulated Data Generation and EM for Posterior Mode (d=2, Circular Data):

In this part of the demonstration, we generate a dataset of size n=500, with p=20 (20 covariates) and the responses are circular,i.e., d=2. Then we Bayesian-LASSO algorithm to estimate the regression coefficients, where we consider the inverse Gamma Prior and sample the tuning parameter.

# LASSO Regression For Directional Data (d=2):

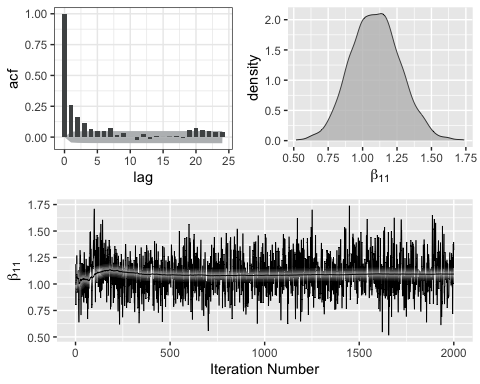
Start\_time= Sys.time()  
Set\_MCSamplerSize=1000  
  
n=500 # NUmber of the samples  
p=20 # NUmber of the regression covariates  
d=2 # Number of direcions in the direcional data  
  
Num\_of\_nonzero\_beta= round(p\*.10)  
Min\_Non\_Zero\_beta= 1  
Max\_Non\_Zero\_beta=10  
data\_lst<-Data\_generator\_vnf\_reg\_sparse(n=n,  
 p=p,  
 d=d,  
 SetUp = 3,  
 NumOfNonZeroBeta=c(Num\_of\_nonzero\_beta,  
 Min\_Non\_Zero\_beta,  
 Max\_Non\_Zero\_beta  
 )  
 )

## [1] " Currently Using Default SetUp=3 "

MCMC\_Sample\_Size=2000#   
 lst\_BLASSO\_Beta\_MCMC = MCMC\_BLASSO\_Dir\_regression\_sampler\_V1( Y=data\_lst$Y,  
 X=data\_lst$X,  
 prior=NULL,  
 beta\_init = NULL,  
 MCSamplerSize =MCMC\_Sample\_Size,  
 lasso\_lambda\_spec = list(  
 Type="SAMPLE",  
 lasso\_lambda=0.01,  
 hyper\_lambda\_selector= NULL  
 ))

## [1] "Default Procedure using EM is being used to obtain initial value of the regression coefficients that will be used to start the MCMC Data Augmentation Algorithm. Iteration number of EM algorithm is being printed untill convergence."  
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6  
## [1] 7  
## [1] 8  
## [1] 9  
## [1] 10  
## [1] 11  
## [1] 12  
## [1] 13  
## [1] 14  
## [1] 15  
## [1] 16  
## [1] 17  
## [1] 18  
## [1] 19  
## [1] 20  
## [1] 21  
## [1] 22  
## [1] 23  
## [1] 24  
## [1] 25  
## [1] 26  
## [1] 27  
## [1] 28  
## [1] 29  
## [1] 30  
## [1] 31  
## [1] 32  
## [1] 33  
## [1] 34  
## [1] 35  
## [1] 36  
## [1] 37  
## [1] 38  
## [1] 39  
## [1] 40  
## [1] 41  
## [1] 42  
## [1] 43  
## [1] 44  
## [1] 45  
## [1] 46  
## [1] 47  
## [1] 48  
## [1] 49  
## [1] 50  
## [1] 51  
## [1] 52  
## [1] 53  
## [1] 54  
## [1] 55  
## [1] 56  
## [1] 57  
## [1] 58  
## [1] 59  
## [1] 60  
## [1] 61  
## [1] 62  
## [1] 63  
## [1] 64  
## [1] " Initial value and prior information obtained successfully. The MCMC samples are being generated. This step may take significnt amount of time depending on the MCMC sample size to be Generated. "  
## [1] "MC\_Iter=100completed"  
## [1] "MC\_Iter=200completed"  
## [1] "MC\_Iter=300completed"  
## [1] "MC\_Iter=400completed"  
## [1] "MC\_Iter=500completed"  
## [1] "MC\_Iter=600completed"  
## [1] "MC\_Iter=700completed"  
## [1] "MC\_Iter=800completed"  
## [1] "MC\_Iter=900completed"  
## [1] "MC\_Iter=1000completed"  
## [1] "MC\_Iter=1100completed"  
## [1] "MC\_Iter=1200completed"  
## [1] "MC\_Iter=1300completed"  
## [1] "MC\_Iter=1400completed"  
## [1] "MC\_Iter=1500completed"  
## [1] "MC\_Iter=1600completed"  
## [1] "MC\_Iter=1700completed"  
## [1] "MC\_Iter=1800completed"  
## [1] "MC\_Iter=1900completed"  
## [1] "MC\_Iter=2000completed"

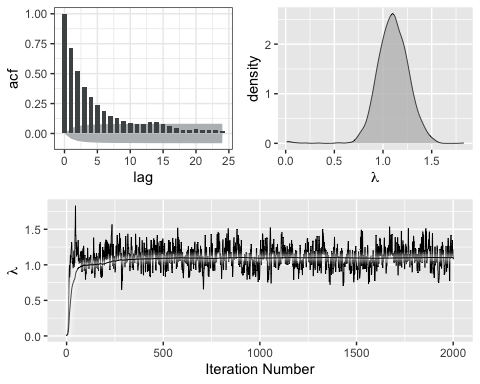
lst=lst\_BLASSO\_Beta\_MCMC  
 i=1;j= 1  
 Plot\_MCMC\_Diag\_Triplet(lst$MC$Mc\_Beta[,i,j],y\_lab\_text = bquote(beta[.(i)][.(j)]))



Posterior\_mean=apply(lst$MC$Mc\_Beta, MARGIN = c(2,3), FUN = mean)  
 Posterior\_Q\_Lower=apply(lst$MC$Mc\_Beta, MARGIN = c(2,3), FUN = function(x){return(c(quantile(x, probs = .025)))} )  
 Posterior\_Q\_Upper=apply(lst$MC$Mc\_Beta, MARGIN = c(2,3), FUN = function(x){return(c(quantile(x, probs = .975)))})  
print(cbind(TrueValue=c(t(data\_lst$beta)), Posterior\_mean=c(t(Posterior\_mean)),Lower\_Bound=c(t(Posterior\_Q\_Lower)), Upper\_Bound=c(t(Posterior\_Q\_Upper) )))

## TrueValue Posterior\_mean Lower\_Bound Upper\_Bound  
## [1,] 1.000690 1.0937117963 0.75903410 1.44614545  
## [2,] 7.544448 7.7601108257 6.92544266 8.78011422  
## [3,] -1.735633 -2.1367558293 -2.48888389 -1.80777738  
## [4,] 3.517584 3.5434832472 3.07513353 4.05356220  
## [5,] 0.000000 0.1164170262 -0.11254914 0.35706218  
## [6,] 0.000000 -0.0292287176 -0.30393083 0.25134110  
## [7,] 0.000000 0.0398638833 -0.17284036 0.25469633  
## [8,] 0.000000 -0.0475216761 -0.32277399 0.23034170  
## [9,] 0.000000 0.0270695049 -0.19508019 0.23771890  
## [10,] 0.000000 0.0733202503 -0.18004598 0.35131847  
## [11,] 0.000000 -0.0201580410 -0.25031431 0.19105388  
## [12,] 0.000000 0.0337256082 -0.23734010 0.30819188  
## [13,] 0.000000 0.0547731411 -0.18533870 0.30757973  
## [14,] 0.000000 0.1990572107 -0.09713485 0.48766902  
## [15,] 0.000000 -0.0384051473 -0.29012676 0.21907566  
## [16,] 0.000000 0.0536302524 -0.24290463 0.35372754  
## [17,] 0.000000 -0.0847695795 -0.31661323 0.13473133  
## [18,] 0.000000 -0.0506947773 -0.33817484 0.22573505  
## [19,] 0.000000 0.2165329563 -0.01740828 0.46194807  
## [20,] 0.000000 0.0233574536 -0.28610712 0.29491911  
## [21,] 0.000000 -0.0832181452 -0.32192408 0.13538333  
## [22,] 0.000000 -0.1534107967 -0.46531830 0.13080963  
## [23,] 0.000000 -0.0832037428 -0.33569632 0.13934172  
## [24,] 0.000000 0.2374707801 -0.05571193 0.53776173  
## [25,] 0.000000 0.0019828505 -0.21752475 0.23797615  
## [26,] 0.000000 0.0919644313 -0.17300219 0.35149219  
## [27,] 0.000000 -0.0749124944 -0.29239149 0.13911351  
## [28,] 0.000000 0.0491888419 -0.20957779 0.30112068  
## [29,] 0.000000 0.0003324116 -0.24712338 0.24482799  
## [30,] 0.000000 -0.0558489202 -0.36477109 0.24453597  
## [31,] 0.000000 -0.1182207165 -0.36453642 0.10810460  
## [32,] 0.000000 -0.1132890238 -0.43951638 0.18789267  
## [33,] 0.000000 -0.0616228378 -0.28410942 0.15805886  
## [34,] 0.000000 -0.1253283099 -0.40366048 0.13347147  
## [35,] 0.000000 -0.0831367035 -0.30617400 0.13371415  
## [36,] 0.000000 0.0249735348 -0.24328400 0.30059533  
## [37,] 0.000000 -0.1690904426 -0.40128371 0.08017753  
## [38,] 0.000000 -0.1359865601 -0.45947162 0.15518345  
## [39,] 0.000000 -0.0059128870 -0.23324711 0.21795010  
## [40,] 0.000000 0.2636016486 -0.01916867 0.55145929

# Samples of the tuning parameter  
 Plot\_MCMC\_Diag\_Triplet(lst$MC$lasso\_lambda\_all,y\_lab\_text = bquote(lambda))



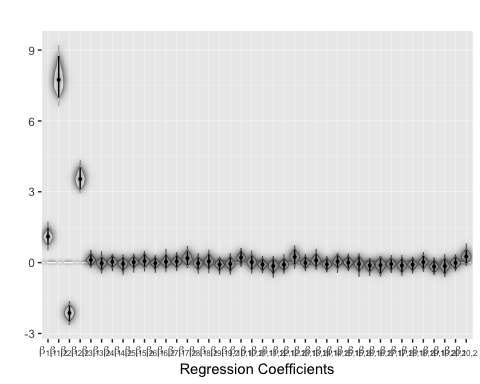
plot\_beta\_p\_20\_d\_2<-function(Mc\_obj=Mc\_obj\_lso, burnIN=1000, PlotType="vio", zero\_marking\_linetype=6, zero\_marking\_linewidth=.8){  
  
  
 #location="/Users/subhadippal/Desktop/Lasso\_Simulation\_RBVNF/"  
 #fileName= paste0("MC\_SIM\_BLASSO\_Reg\_Dir\_Data\_d\_eq\_",2,"\_SimNUmber\_",5,".RData")  
 #file\_with\_path= paste0(location,fileName)  
 #assign('Mc\_obj\_lso', get( load(file=file\_with\_path ) ))  
  
 #plot\_beta\_p\_20\_d\_2(Mc\_obj\_lso)  
  
 data\_summary <- function(x) {  
 m <- median(x)  
 ymin <- as.numeric(quantile(x,.025))#m-sd(x)  
 ymax <- as.numeric(quantile(x, 0.975))#m+sd(x)  
 return(c(y=m,ymin=ymin,ymax=ymax))  
 }  
  
 Mc\_Beta\_burnin<-((Mc\_obj$MC\_lst$MC$Mc\_Beta[-(1:burnIN), , ]))  
 mc\_samples= Mc\_Beta\_burnin  
 mc\_len=dim(mc\_samples)[1]  
 value1 <- ( matrix(mc\_samples, nrow=mc\_len ))  
  
index\_rearrange=0\*1:40  
 for(i in 1:20){  
 index\_rearrange[2\*i-1]= i; index\_rearrange[2\*i]= 20+i  
 }  
value= value1[,index\_rearrange]  
#names <- c(rep("A", mc\_len) , rep("B", 5) , rep("C", 30), rep("D", 100))  
beta\_names<-c(t(replicate(1, paste0("beta",paste0('[paste(',row(mc\_samples[1, , ]), ",',',",col(mc\_samples[1, , ]), ')]')))))[index\_rearrange]  
beta= c(t(replicate(mc\_len, paste0("beta", 111:150))))  
#beta<-c(t(replicate(mc\_len, paste0("beta",paste0('[paste(',row(mc\_samples[1, , ]), ",',',",col(mc\_samples[1, , ]), ')]')))))  
  
data <- data.frame(Names=c(beta),Value=c(value))  
  
# prepare a special xlab with the number of obs for each group  
my\_xlab <- paste0("c(", ((paste("expression(",(beta\_names),")",collapse=","))), ")")  
  
# plot  
  
  
p<-ggplot(data, aes(x=Names, y=Value,fill=Names ))  
if(PlotType=='vio'){  
 p<-p+with\_shadow(geom\_hline(yintercept=0, col="white",linewidth=zero\_marking\_linewidth, linetype=zero\_marking\_linetype ), sigma = 1,  
 x\_offset = 0,  
 y\_offset = 0,  
 colour = "black" ) #   
 p<-p+ with\_shadow(geom\_violin(alpha=0.9,  
 linewidth = .05, scale = 'width'),  
 sigma = 3,  
 x\_offset = 0,  
 y\_offset = 0,  
 colour = "black" )#  
 #p<-p+stat\_summary(fun.data=mean\_sdl, mult=1, geom="pointrange", color="red")  
 p<- p + with\_shadow(stat\_summary(fun.data=data\_summary, size=.01, col="black"), sigma = 2,  
 x\_offset = 0,  
 y\_offset = 0,  
 colour = "black" )#   
  
}  
if(PlotType!='vio'){  
p<-p+ with\_shadow(geom\_boxplot(alpha=0.85,  
 linewidth = .15,  
 outlier.size = .05, outlier.colour = "black"),  
 sigma = 2,  
 x\_offset = 0,  
 y\_offset = 0,  
 colour = "black" )  
}  
  
 p<-p+theme(legend.position="none") +scale\_x\_discrete(labels=eval(parse(text=my\_xlab)))  
p<-p+theme(panel.grid = element\_line(color = "white",  
 size = 0.15,  
 linetype = 1))  
  
p<-p+scale\_fill\_manual(values=c(replicate(4, "white"), replicate(36, "gray")))  
p<-p+xlab("Regression Coefficients")+ ylab(" ")+ ggtitle(" ")  
return(p)  
}

Estimated Regression coefficients:

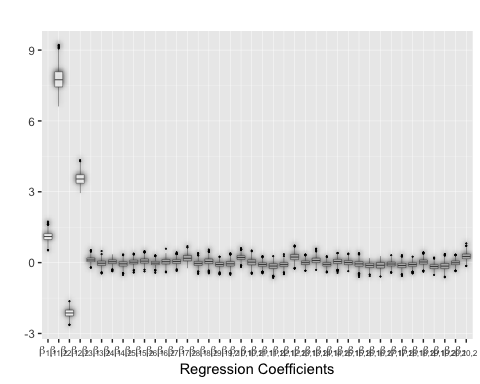
Mc\_obj=list(MC\_lst=lst\_BLASSO\_Beta\_MCMC)  
plt\_vio<-plot\_beta\_p\_20\_d\_2(Mc\_obj, PlotType = 'vio', zero\_marking\_linetype=6, zero\_marking\_linewidth=.5)

## Warning: The `size` argument of `element\_line()` is deprecated as of ggplot2 3.4.0.  
## ℹ Please use the `linewidth` argument instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

plt\_vio



plt\_box<-plot\_beta\_p\_20\_d\_2(Mc\_obj, PlotType = 'box', zero\_marking\_linetype=6, zero\_marking\_linewidth=.5)  
plt\_box



# Cross Validation and LASSO Estimate:

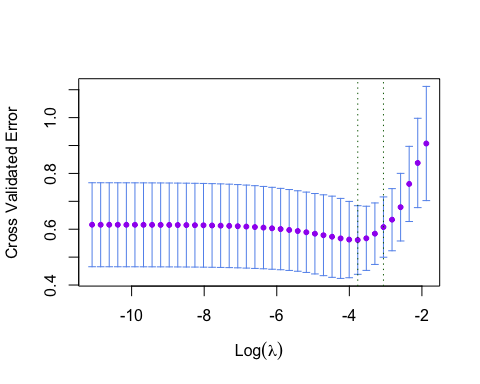
library(glmnet)  
n=200 # Number of the samples  
p=30 # Number of the regression covariates  
d=5 # Number of directions in the directional data  
  
#Test Lasso  
data\_lst <- Data\_generator\_vnf\_reg\_sparse(n=200, p=p, d=d,SetUp = 2, NumOfNonZeroBeta=c(4, 1, 10))

## [1] "Currently SetUp can be wither 1 or 2. Currently Using Default SetUp=2 "

Y = data\_lst$Y;X=data\_lst$X;  
  
  
  
cv\_LASSO\_output<-EM\_BLASSO\_Dir\_regression\_optimizer\_V1.cv(Y=data\_lst$Y,  
 X=data\_lst$X,  
 beta\_init = NULL,  
 Max\_EM\_iter=1000,  
 cv\_k\_fold = 8,  
 cv\_lambda\_n = 40,  
 epsilon\_lambda\_range\_min = .0001,  
 lambda\_Range\_Type = 2  
 )

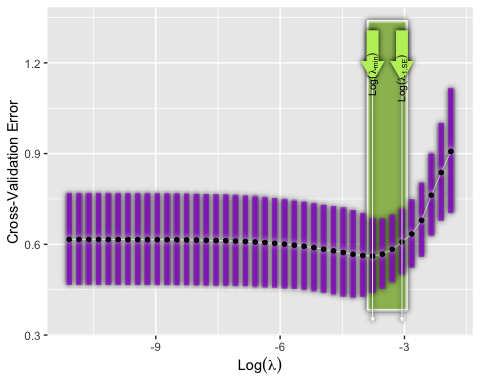
## [1] "Cross Validated Error computation is completed for all lambda and for the Testing Fold Number= 1"  
## [1] "Cross Validated Error computation is completed for all lambda and for the Testing Fold Number= 2"  
## [1] "Cross Validated Error computation is completed for all lambda and for the Testing Fold Number= 3"  
## [1] "Cross Validated Error computation is completed for all lambda and for the Testing Fold Number= 4"  
## [1] "Cross Validated Error computation is completed for all lambda and for the Testing Fold Number= 5"  
## [1] "Cross Validated Error computation is completed for all lambda and for the Testing Fold Number= 6"  
## [1] "Cross Validated Error computation is completed for all lambda and for the Testing Fold Number= 7"  
## [1] "Cross Validated Error computation is completed for all lambda and for the Testing Fold Number= 8"

plot.cv.Dir\_Lasso\_Reg(cv\_LASSO\_output)



library(ggfx)  
plt\_cv\_lambda<-plot.cv.Dir\_Lasso\_Reg\_gg(cv\_LASSO\_output, color\_theme = 2)  
plt\_cv\_lambda

## Warning in is.na(x): is.na() applied to non-(list or vector) of type  
## 'expression'  
  
## Warning in is.na(x): is.na() applied to non-(list or vector) of type  
## 'expression'



beta\_EM\_Lasso\_lambda\_min=EM\_BLASSO\_Dir\_regression\_optimizer\_V1(Y=data\_lst$Y, X=data\_lst$X, beta\_init = NULL, lasso\_lambda = max(cv\_LASSO\_output$lambda.min), EM\_tolerence = .00001)

## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6  
## [1] 7  
## [1] 8  
## [1] 9  
## [1] 10  
## [1] 11  
## [1] 12  
## [1] 13  
## [1] 14  
## [1] 15  
## [1] 16  
## [1] 17  
## [1] 18  
## [1] 19  
## [1] 20  
## [1] 21  
## [1] 22  
## [1] 23  
## [1] 24  
## [1] 25  
## [1] 26  
## [1] 27  
## [1] 28  
## [1] 29  
## [1] 30  
## [1] 31  
## [1] 32  
## [1] 33  
## [1] 34  
## [1] 35  
## [1] 36  
## [1] 37  
## [1] 38  
## [1] 39  
## [1] 40  
## [1] 41  
## [1] 42  
## [1] 43  
## [1] 44  
## [1] 45  
## [1] 46  
## [1] 47  
## [1] 48  
## [1] 49  
## [1] 50  
## [1] 51  
## [1] 52  
## [1] 53  
## [1] 54  
## [1] 55  
## [1] 56  
## [1] 57  
## [1] 58  
## [1] 59  
## [1] 60

beta\_EM\_Lasso\_lambda1.se=EM\_BLASSO\_Dir\_regression\_optimizer\_V1(Y=data\_lst$Y, X=data\_lst$X, beta\_init = NULL, lasso\_lambda = max(cv\_LASSO\_output$lambda.1se), EM\_tolerence = .00001)

## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6  
## [1] 7  
## [1] 8  
## [1] 9  
## [1] 10  
## [1] 11  
## [1] 12  
## [1] 13  
## [1] 14  
## [1] 15  
## [1] 16  
## [1] 17  
## [1] 18  
## [1] 19  
## [1] 20  
## [1] 21  
## [1] 22  
## [1] 23  
## [1] 24  
## [1] 25  
## [1] 26  
## [1] 27  
## [1] 28  
## [1] 29  
## [1] 30  
## [1] 31  
## [1] 32  
## [1] 33

print(cbind(TrueValue=c(t(data\_lst$beta)), beta\_EM\_Lasso\_lambda\_min=c(t(beta\_EM\_Lasso\_lambda\_min)), beta\_EM\_Lasso\_lambda1.se=c(t(beta\_EM\_Lasso\_lambda1.se)) ))

## TrueValue beta\_EM\_Lasso\_lambda\_min beta\_EM\_Lasso\_lambda1.se  
## [1,] 7.019657 3.379397364 1.97388812  
## [2,] 4.720968 1.571784011 0.62198352  
## [3,] 9.655963 4.408121610 2.60994932  
## [4,] 2.825150 0.980554399 0.28018807  
## [5,] 4.052278 1.610233003 0.75616509  
## [6,] 4.904778 1.574209478 0.65485400  
## [7,] 7.837770 3.173626891 1.68737558  
## [8,] 6.891127 2.674224991 1.33379835  
## [9,] 2.938765 0.786263575 0.03907576  
## [10,] 4.494591 1.635923357 0.70910486  
## [11,] 2.113003 0.071441720 0.00000000  
## [12,] 7.604355 3.545638619 1.97397753  
## [13,] 5.763493 2.590313002 1.28690580  
## [14,] 4.964992 2.118997750 1.02593166  
## [15,] 4.016465 1.129486283 0.36091391  
## [16,] 4.096446 1.507882994 0.65314355  
## [17,] 4.842084 1.902175951 0.81331179  
## [18,] 3.011141 0.641725337 0.00000000  
## [19,] 6.187760 2.959942914 1.67121438  
## [20,] 2.664374 0.737348364 0.11029169  
## [21,] 0.000000 0.000000000 0.00000000  
## [22,] 0.000000 0.000000000 0.00000000  
## [23,] 0.000000 0.000000000 0.00000000  
## [24,] 0.000000 0.000000000 0.00000000  
## [25,] 0.000000 0.000000000 0.00000000  
## [26,] 0.000000 0.000000000 0.00000000  
## [27,] 0.000000 0.000000000 0.00000000  
## [28,] 0.000000 0.000000000 0.00000000  
## [29,] 0.000000 0.000000000 0.00000000  
## [30,] 0.000000 0.000000000 0.00000000  
## [31,] 0.000000 0.000000000 0.00000000  
## [32,] 0.000000 0.000000000 0.00000000  
## [33,] 0.000000 0.000000000 0.00000000  
## [34,] 0.000000 0.000000000 0.00000000  
## [35,] 0.000000 0.051455345 0.00000000  
## [36,] 0.000000 0.000000000 0.00000000  
## [37,] 0.000000 0.000000000 0.00000000  
## [38,] 0.000000 0.180194017 0.00000000  
## [39,] 0.000000 0.000000000 0.00000000  
## [40,] 0.000000 0.000000000 0.00000000  
## [41,] 0.000000 0.000000000 0.00000000  
## [42,] 0.000000 0.000000000 0.00000000  
## [43,] 0.000000 0.000000000 0.00000000  
## [44,] 0.000000 0.000000000 0.00000000  
## [45,] 0.000000 0.000000000 0.00000000  
## [46,] 0.000000 0.000000000 0.00000000  
## [47,] 0.000000 0.000000000 0.00000000  
## [48,] 0.000000 0.000000000 0.00000000  
## [49,] 0.000000 0.000000000 0.00000000  
## [50,] 0.000000 0.000000000 0.00000000  
## [51,] 0.000000 0.000000000 0.00000000  
## [52,] 0.000000 -0.084923701 0.00000000  
## [53,] 0.000000 0.000000000 0.00000000  
## [54,] 0.000000 0.000000000 0.00000000  
## [55,] 0.000000 0.000000000 0.00000000  
## [56,] 0.000000 0.000000000 0.00000000  
## [57,] 0.000000 0.000000000 0.00000000  
## [58,] 0.000000 0.000000000 0.00000000  
## [59,] 0.000000 0.000000000 0.00000000  
## [60,] 0.000000 0.000000000 0.00000000  
## [61,] 0.000000 0.000000000 0.00000000  
## [62,] 0.000000 0.000000000 0.00000000  
## [63,] 0.000000 0.000000000 0.00000000  
## [64,] 0.000000 0.000000000 0.00000000  
## [65,] 0.000000 0.000000000 0.00000000  
## [66,] 0.000000 0.000000000 0.00000000  
## [67,] 0.000000 0.000000000 0.00000000  
## [68,] 0.000000 0.000000000 0.00000000  
## [69,] 0.000000 -0.002402030 0.00000000  
## [70,] 0.000000 0.000000000 0.00000000  
## [71,] 0.000000 0.000000000 0.00000000  
## [72,] 0.000000 0.000000000 0.00000000  
## [73,] 0.000000 0.000000000 0.00000000  
## [74,] 0.000000 -0.150522246 0.00000000  
## [75,] 0.000000 0.000000000 0.00000000  
## [76,] 0.000000 0.000000000 0.00000000  
## [77,] 0.000000 0.000000000 0.00000000  
## [78,] 0.000000 0.000000000 0.00000000  
## [79,] 0.000000 0.000000000 0.00000000  
## [80,] 0.000000 0.000000000 0.00000000  
## [81,] 0.000000 0.000000000 0.00000000  
## [82,] 0.000000 0.060507785 0.00000000  
## [83,] 0.000000 0.000000000 0.00000000  
## [84,] 0.000000 0.000000000 0.00000000  
## [85,] 0.000000 0.000000000 0.00000000  
## [86,] 0.000000 0.000000000 0.00000000  
## [87,] 0.000000 0.000000000 0.00000000  
## [88,] 0.000000 0.000000000 0.00000000  
## [89,] 0.000000 0.000000000 0.00000000  
## [90,] 0.000000 0.000000000 0.00000000  
## [91,] 0.000000 0.000000000 0.00000000  
## [92,] 0.000000 0.000000000 0.00000000  
## [93,] 0.000000 0.000000000 0.00000000  
## [94,] 0.000000 0.000000000 0.00000000  
## [95,] 0.000000 0.000000000 0.00000000  
## [96,] 0.000000 0.000000000 0.00000000  
## [97,] 0.000000 0.000000000 0.00000000  
## [98,] 0.000000 0.000000000 0.00000000  
## [99,] 0.000000 0.000000000 0.00000000  
## [100,] 0.000000 0.000000000 0.00000000  
## [101,] 0.000000 0.000000000 0.00000000  
## [102,] 0.000000 0.000000000 0.00000000  
## [103,] 0.000000 0.000000000 0.00000000  
## [104,] 0.000000 0.000000000 0.00000000  
## [105,] 0.000000 0.000000000 0.00000000  
## [106,] 0.000000 0.000000000 0.00000000  
## [107,] 0.000000 0.000000000 0.00000000  
## [108,] 0.000000 -0.057004770 0.00000000  
## [109,] 0.000000 -0.002491328 0.00000000  
## [110,] 0.000000 0.006613456 0.00000000  
## [111,] 0.000000 -0.380379273 -0.10034257  
## [112,] 0.000000 0.000000000 0.00000000  
## [113,] 0.000000 0.000000000 0.00000000  
## [114,] 0.000000 0.000000000 0.00000000  
## [115,] 0.000000 0.000000000 0.00000000  
## [116,] 0.000000 0.000000000 0.00000000  
## [117,] 0.000000 0.000000000 0.00000000  
## [118,] 0.000000 0.000000000 0.00000000  
## [119,] 0.000000 0.000000000 0.00000000  
## [120,] 0.000000 0.035110035 0.00000000  
## [121,] 0.000000 0.000000000 0.00000000  
## [122,] 0.000000 0.000000000 0.00000000  
## [123,] 0.000000 0.000000000 0.00000000  
## [124,] 0.000000 0.000000000 0.00000000  
## [125,] 0.000000 0.000000000 0.00000000  
## [126,] 0.000000 0.000000000 0.00000000  
## [127,] 0.000000 0.000000000 0.00000000  
## [128,] 0.000000 0.000000000 0.00000000  
## [129,] 0.000000 0.000000000 0.00000000  
## [130,] 0.000000 0.000000000 0.00000000  
## [131,] 0.000000 0.000000000 0.00000000  
## [132,] 0.000000 0.000000000 0.00000000  
## [133,] 0.000000 0.000000000 0.00000000  
## [134,] 0.000000 0.000000000 0.00000000  
## [135,] 0.000000 0.000000000 0.00000000  
## [136,] 0.000000 0.000000000 0.00000000  
## [137,] 0.000000 0.000000000 0.00000000  
## [138,] 0.000000 0.000000000 0.00000000  
## [139,] 0.000000 -0.083540309 0.00000000  
## [140,] 0.000000 0.000000000 0.00000000  
## [141,] 0.000000 0.000000000 0.00000000  
## [142,] 0.000000 -0.054270937 0.00000000  
## [143,] 0.000000 0.000000000 0.00000000  
## [144,] 0.000000 0.000000000 0.00000000  
## [145,] 0.000000 0.000000000 0.00000000  
## [146,] 0.000000 0.186967076 0.00000000  
## [147,] 0.000000 0.000000000 0.00000000  
## [148,] 0.000000 0.000000000 0.00000000  
## [149,] 0.000000 0.000000000 0.00000000  
## [150,] 0.000000 -0.061153568 0.00000000

End\_time= Sys.time()  
  
print(paste0(" Total Run Time: ",End\_time-Start\_time, "minutes" ))

## [1] " Total Run Time: 10.1090165336927minutes"