Reproduciblity: Bayesian Regression for Directional Data

# 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

## 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=750, with p=10 (10 covariates) and the responses are circular,i.e., d=2. Then we fit the EM algorithm to estimate the regression coefficients. True value of the regression coefficients, and its estimates are printed.

Start\_time= Sys.time()  
Set\_MCSamplerSize=1000  
  
n=750 # NUmber of the samples  
p=10 # NUmber of the regression covariates  
d=2 # Number of direcions in the direcional data  
#### bbeta is a matrix of dimension p\times d  
#bbeta=matrix( rnorm(p\*d), nrow=p, ncol=d)  
sigma\_square=1  
tau\_square=10000  
data\_lst = Data\_generator\_vnf\_reg(n=n, p=p, d=d, concentration\_factor = 1, beta\_factor = 10)  
Y = data\_lst$Y;X=data\_lst$X;  
  
# Fitting the EM algorithm for the Standard Regresion for directional responses: This takes less than a minute.   
beta\_EM=EM\_Dir\_regression\_optimizer\_V1(Y=Y, X=X, prior=NULL, beta\_init = NULL, 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  
## [1] 61  
## [1] 62  
## [1] 63  
## [1] 64  
## [1] 65  
## [1] 66  
## [1] 67  
## [1] 68  
## [1] 69  
## [1] 70  
## [1] 71  
## [1] 72  
## [1] 73  
## [1] 74  
## [1] 75  
## [1] 76  
## [1] 77  
## [1] 78  
## [1] 79  
## [1] 80  
## [1] 81  
## [1] 82  
## [1] 83  
## [1] 84  
## [1] 85  
## [1] 86  
## [1] 87  
## [1] 88  
## [1] 89  
## [1] 90  
## [1] 91  
## [1] 92  
## [1] 93  
## [1] 94  
## [1] 95  
## [1] 96  
## [1] 97  
## [1] 98  
## [1] 99  
## [1] 100  
## [1] 101  
## [1] 102  
## [1] 103  
## [1] 104  
## [1] 105  
## [1] 106  
## [1] 107  
## [1] 108  
## [1] 109  
## [1] 110  
## [1] 111  
## [1] 112  
## [1] 113  
## [1] 114  
## [1] 115  
## [1] 116  
## [1] 117  
## [1] 118  
## [1] 119  
## [1] 120  
## [1] 121  
## [1] 122  
## [1] 123  
## [1] 124  
## [1] 125  
## [1] 126  
## [1] 127  
## [1] 128  
## [1] 129  
## [1] 130  
## [1] 131  
## [1] 132  
## [1] 133  
## [1] 134  
## [1] 135  
## [1] 136  
## [1] 137  
## [1] 138  
## [1] 139  
## [1] 140  
## [1] 141  
## [1] 142  
## [1] 143  
## [1] 144  
## [1] 145  
## [1] 146  
## [1] 147  
## [1] 148  
## [1] 149  
## [1] 150  
## [1] 151  
## [1] 152  
## [1] 153  
## [1] 154  
## [1] 155  
## [1] 156  
## [1] 157  
## [1] 158  
## [1] 159  
## [1] 160  
## [1] 161  
## [1] 162  
## [1] 163  
## [1] 164  
## [1] 165  
## [1] 166  
## [1] 167  
## [1] 168  
## [1] 169  
## [1] 170  
## [1] 171  
## [1] 172  
## [1] 173  
## [1] 174  
## [1] 175  
## [1] 176  
## [1] 177  
## [1] 178  
## [1] 179  
## [1] 180  
## [1] 181  
## [1] 182  
## [1] 183  
## [1] 184  
## [1] 185  
## [1] 186  
## [1] 187  
## [1] 188  
## [1] 189  
## [1] 190  
## [1] 191  
## [1] 192  
## [1] 193  
## [1] 194  
## [1] 195  
## [1] 196  
## [1] 197  
## [1] 198  
## [1] 199  
## [1] 200  
## [1] 201  
## [1] 202  
## [1] 203  
## [1] 204  
## [1] 205  
## [1] 206  
## [1] 207  
## [1] 208  
## [1] 209  
## [1] 210  
## [1] 211  
## [1] 212  
## [1] 213  
## [1] 214  
## [1] 215  
## [1] 216  
## [1] 217  
## [1] 218  
## [1] 219  
## [1] 220  
## [1] 221  
## [1] 222  
## [1] 223  
## [1] 224  
## [1] 225  
## [1] 226  
## [1] 227  
## [1] 228  
## [1] 229  
## [1] 230  
## [1] 231  
## [1] 232  
## [1] 233  
## [1] 234  
## [1] 235  
## [1] 236  
## [1] 237  
## [1] 238  
## [1] 239  
## [1] 240  
## [1] 241  
## [1] 242  
## [1] 243  
## [1] 244  
## [1] 245  
## [1] 246  
## [1] 247  
## [1] 248  
## [1] 249  
## [1] 250  
## [1] 251  
## [1] 252  
## [1] 253  
## [1] 254  
## [1] 255  
## [1] 256  
## [1] 257  
## [1] 258  
## [1] 259  
## [1] 260  
## [1] 261  
## [1] 262  
## [1] 263  
## [1] 264  
## [1] 265  
## [1] 266  
## [1] 267  
## [1] 268  
## [1] 269  
## [1] 270  
## [1] 271  
## [1] 272  
## [1] 273  
## [1] 274  
## [1] 275  
## [1] 276  
## [1] 277  
## [1] 278  
## [1] 279  
## [1] 280  
## [1] 281  
## [1] 282  
## [1] 283  
## [1] 284  
## [1] 285  
## [1] 286  
## [1] 287  
## [1] 288  
## [1] 289  
## [1] 290  
## [1] 291  
## [1] 292  
## [1] 293  
## [1] 294  
## [1] 295  
## [1] 296  
## [1] 297  
## [1] 298  
## [1] 299  
## [1] 300  
## [1] 301  
## [1] 302  
## [1] 303  
## [1] 304  
## [1] 305  
## [1] 306  
## [1] 307  
## [1] 308  
## [1] 309  
## [1] 310  
## [1] 311  
## [1] 312  
## [1] 313  
## [1] 314  
## [1] 315  
## [1] 316  
## [1] 317  
## [1] 318  
## [1] 319  
## [1] 320  
## [1] 321  
## [1] 322  
## [1] 323  
## [1] 324  
## [1] 325  
## [1] 326  
## [1] 327  
## [1] 328  
## [1] 329  
## [1] 330  
## [1] 331  
## [1] 332  
## [1] 333  
## [1] 334  
## [1] 335  
## [1] 336  
## [1] 337  
## [1] 338  
## [1] 339  
## [1] 340  
## [1] 341  
## [1] 342  
## [1] 343  
## [1] 344  
## [1] 345  
## [1] 346  
## [1] 347  
## [1] 348  
## [1] 349  
## [1] 350  
## [1] 351  
## [1] 352  
## [1] 353  
## [1] 354  
## [1] 355  
## [1] 356  
## [1] 357  
## [1] 358  
## [1] 359  
## [1] 360  
## [1] 361  
## [1] 362  
## [1] 363  
## [1] 364  
## [1] 365  
## [1] 366  
## [1] 367

colnames(beta\_EM)= gsub("Y","Beta", colnames(beta\_EM))  
print(paste("Estimated Beta=", beta\_EM))

## [1] "Estimated Beta= -0.650770724485358" "Estimated Beta= 4.04252513196771"   
## [3] "Estimated Beta= 1.36945931747161" "Estimated Beta= -4.67916701372806"   
## [5] "Estimated Beta= -2.29849751461967" "Estimated Beta= -8.71541384111557"   
## [7] "Estimated Beta= 7.52221202001526" "Estimated Beta= -0.832673234543997"  
## [9] "Estimated Beta= -1.48681791438682" "Estimated Beta= 5.71250057824619"   
## [11] "Estimated Beta= -1.80861931949153" "Estimated Beta= 4.66008024998899"   
## [13] "Estimated Beta= 4.0921697024081" "Estimated Beta= 0.678595434937384"   
## [15] "Estimated Beta= -5.28190946011421" "Estimated Beta= 6.42606578413363"   
## [17] "Estimated Beta= -6.307433121304" "Estimated Beta= -6.60939441409675"   
## [19] "Estimated Beta= 9.02117414835791" "Estimated Beta= -4.76884168636692"

print(cbind(EstimatedValue=c(t(beta\_EM)),TrueValue=c(t(data\_lst$beta))))

## EstimatedValue TrueValue  
## [1,] -0.6507707 -0.9247039  
## [2,] -1.8086193 -1.5444272  
## [3,] 4.0425251 4.6009674  
## [4,] 4.6600802 5.4955359  
## [5,] 1.3694593 1.6466148  
## [6,] 4.0921697 4.5192532  
## [7,] -4.6791670 -5.1261019  
## [8,] 0.6785954 0.9156271  
## [9,] -2.2984975 -2.5608955  
## [10,] -5.2819095 -5.2779709  
## [11,] -8.7154138 -9.0469088  
## [12,] 6.4260658 6.5120942  
## [13,] 7.5222120 7.3960711  
## [14,] -6.3074331 -6.2419567  
## [15,] -0.8326732 -0.9617236  
## [16,] -6.6093944 -6.6669372  
## [17,] -1.4868179 -1.4231486  
## [18,] 9.0211741 9.3063619  
## [19,] 5.7125006 5.8621506  
## [20,] -4.7688417 -5.1951071

## Bayesian MCMC Algorithm (d=2):

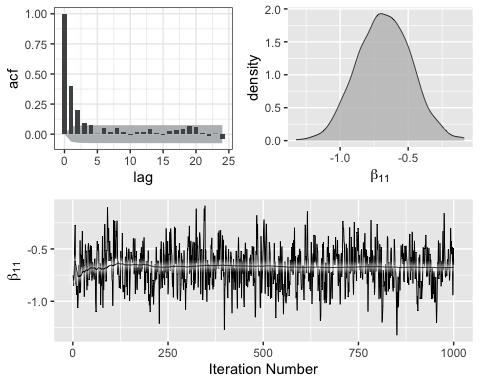
Here we obtain the posterior samples of the regression coefficients using the MCMC algorithm.

# Change Sample Size to get the full MCMC. MCMC step takes time depending on the sample size. This step can take 20 to 30 minutes. Prints output after every 100 samples are generated.  
lst=MCMC\_Dir\_regression\_sampler\_V1(Y=data\_lst$Y, X=data\_lst$X, prior=NULL, beta\_init = NULL, MCSamplerSize =Set\_MCSamplerSize)

## [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] 65  
## [1] 66  
## [1] 67  
## [1] 68  
## [1] 69  
## [1] 70  
## [1] 71  
## [1] 72  
## [1] 73  
## [1] 74  
## [1] 75  
## [1] 76  
## [1] 77  
## [1] 78  
## [1] 79  
## [1] 80  
## [1] 81  
## [1] 82  
## [1] 83  
## [1] 84  
## [1] 85  
## [1] 86  
## [1] 87  
## [1] 88  
## [1] 89  
## [1] 90  
## [1] 91  
## [1] 92  
## [1] 93  
## [1] 94  
## [1] 95  
## [1] 96  
## [1] 97  
## [1] 98  
## [1] 99  
## [1] 100  
## [1] 101  
## [1] 102  
## [1] 103  
## [1] 104  
## [1] 105  
## [1] 106  
## [1] 107  
## [1] 108  
## [1] 109  
## [1] 110  
## [1] 111  
## [1] 112  
## [1] 113  
## [1] 114  
## [1] 115  
## [1] 116  
## [1] 117  
## [1] 118  
## [1] 119  
## [1] 120  
## [1] 121  
## [1] 122  
## [1] 123  
## [1] 124  
## [1] 125  
## [1] 126  
## [1] 127  
## [1] 128  
## [1] 129  
## [1] 130  
## [1] 131  
## [1] 132  
## [1] 133  
## [1] 134  
## [1] 135  
## [1] 136  
## [1] 137  
## [1] 138  
## [1] 139  
## [1] 140  
## [1] 141  
## [1] 142  
## [1] 143  
## [1] 144  
## [1] 145  
## [1] 146  
## [1] 147  
## [1] 148  
## [1] 149  
## [1] 150  
## [1] 151  
## [1] 152  
## [1] 153  
## [1] 154  
## [1] 155  
## [1] 156  
## [1] 157  
## [1] 158  
## [1] 159  
## [1] 160  
## [1] 161  
## [1] 162  
## [1] 163  
## [1] 164  
## [1] 165  
## [1] 166  
## [1] 167  
## [1] 168  
## [1] 169  
## [1] 170  
## [1] 171  
## [1] 172  
## [1] 173  
## [1] 174  
## [1] 175  
## [1] 176  
## [1] 177  
## [1] 178  
## [1] 179  
## [1] 180  
## [1] 181  
## [1] 182  
## [1] 183  
## [1] 184  
## [1] 185  
## [1] 186  
## [1] 187  
## [1] 188  
## [1] 189  
## [1] 190  
## [1] 191  
## [1] 192  
## [1] 193  
## [1] 194  
## [1] 195  
## [1] 196  
## [1] 197  
## [1] 198  
## [1] 199  
## [1] 200  
## [1] 201  
## [1] 202  
## [1] 203  
## [1] 204  
## [1] 205  
## [1] 206  
## [1] 207  
## [1] 208  
## [1] 209  
## [1] 210  
## [1] 211  
## [1] 212  
## [1] 213  
## [1] 214  
## [1] 215  
## [1] 216  
## [1] 217  
## [1] 218  
## [1] 219  
## [1] 220  
## [1] 221  
## [1] 222  
## [1] 223  
## [1] 224  
## [1] 225  
## [1] 226  
## [1] 227  
## [1] 228  
## [1] 229  
## [1] 230  
## [1] 231  
## [1] 232  
## [1] 233  
## [1] 234  
## [1] 235  
## [1] 236  
## [1] 237  
## [1] 238  
## [1] 239  
## [1] 240  
## [1] 241  
## [1] 242  
## [1] 243  
## [1] 244  
## [1] 245  
## [1] 246  
## [1] 247  
## [1] 248  
## [1] 249  
## [1] 250  
## [1] 251  
## [1] 252  
## [1] 253  
## [1] 254  
## [1] 255  
## [1] 256  
## [1] 257  
## [1] 258  
## [1] 259  
## [1] 260  
## [1] 261  
## [1] 262  
## [1] 263  
## [1] 264  
## [1] 265  
## [1] 266  
## [1] 267  
## [1] 268  
## [1] 269  
## [1] 270  
## [1] 271  
## [1] 272  
## [1] 273  
## [1] 274  
## [1] 275  
## [1] 276  
## [1] 277  
## [1] 278  
## [1] 279  
## [1] 280  
## [1] 281  
## [1] 282  
## [1] 283  
## [1] 284  
## [1] 285  
## [1] 286  
## [1] 287  
## [1] 288  
## [1] 289  
## [1] 290  
## [1] 291  
## [1] 292  
## [1] 293  
## [1] 294  
## [1] 295  
## [1] 296  
## [1] 297  
## [1] 298  
## [1] 299  
## [1] 300  
## [1] 301  
## [1] 302  
## [1] 303  
## [1] 304  
## [1] 305  
## [1] 306  
## [1] 307  
## [1] 308  
## [1] 309  
## [1] 310  
## [1] 311  
## [1] 312  
## [1] 313  
## [1] 314  
## [1] 315  
## [1] 316  
## [1] 317  
## [1] 318  
## [1] 319  
## [1] 320  
## [1] 321  
## [1] 322  
## [1] 323  
## [1] 324  
## [1] 325  
## [1] 326  
## [1] 327  
## [1] 328  
## [1] 329  
## [1] 330  
## [1] 331  
## [1] 332  
## [1] 333  
## [1] 334  
## [1] 335  
## [1] 336  
## [1] 337  
## [1] 338  
## [1] 339  
## [1] 340  
## [1] 341  
## [1] 342  
## [1] 343  
## [1] 344  
## [1] 345  
## [1] 346  
## [1] 347  
## [1] 348  
## [1] 349  
## [1] 350  
## [1] 351  
## [1] 352  
## [1] 353  
## [1] 354  
## [1] 355  
## [1] 356  
## [1] 357  
## [1] 358  
## [1] 359  
## [1] 360  
## [1] 361  
## [1] 362  
## [1] 363  
## [1] 364  
## [1] 365  
## [1] 366  
## [1] 367  
## [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"

The triplet plot (autocorrelation, traceplot and density plot) for some of the regression coefficient is plotted (d=2)

# Summary from MCMC output  
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\_SD=apply(lst$MC$Mc\_Beta, MARGIN = c(2,3), FUN = sd)  
 print(cbind(Posterior\_mean=c(t(Posterior\_mean)),TrueValue=c(t(data\_lst$beta))))

## Posterior\_mean TrueValue  
## [1,] -0.6765258 -0.9247039  
## [2,] -1.8627653 -1.5444272  
## [3,] 4.1980326 4.6009674  
## [4,] 4.8340897 5.4955359  
## [5,] 1.4227667 1.6466148  
## [6,] 4.2349652 4.5192532  
## [7,] -4.8584229 -5.1261019  
## [8,] 0.7077420 0.9156271  
## [9,] -2.3683394 -2.5608955  
## [10,] -5.4801222 -5.2779709  
## [11,] -9.0461421 -9.0469088  
## [12,] 6.6584729 6.5120942  
## [13,] 7.7938162 7.3960711  
## [14,] -6.5379800 -6.2419567  
## [15,] -0.8709495 -0.9617236  
## [16,] -6.8748198 -6.6669372  
## [17,] -1.5390838 -1.4231486  
## [18,] 9.3717328 9.3063619  
## [19,] 5.9296691 5.8621506  
## [20,] -4.9406651 -5.1951071

## Simulated Data Generation and EM for Posterior Mode (d=3, Spherical Data):

In this part of the demonstration, we generate a dataset of size n=750, with p=10 (10 covariates) and the responses are circular,i.e., d=2. Then we fit the EM algorithm to estimate the regression coefficients. True value of the regression coefficients, and its estimates are printed.

n=750 # NUmber of the samples  
p=10 # NUmber of the regression covariates  
d=3 # Number of direcions in the direcional data  
#### bbeta is a matrix of dimension p\times d  
#bbeta=matrix( rnorm(p\*d), nrow=p, ncol=d)  
sigma\_square=1  
tau\_square=10000  
data\_lst = Data\_generator\_vnf\_reg(n=n, p=p, d=d, concentration\_factor = 1, beta\_factor = 10)  
Y = data\_lst$Y;X=data\_lst$X;  
  
# Fitting the EM algorithm for the Standard Regresion for directional responses: This takes less than a minute.   
beta\_EM=EM\_Dir\_regression\_optimizer\_V1(Y=Y, X=X, prior=NULL, beta\_init = NULL, 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  
## [1] 61  
## [1] 62  
## [1] 63  
## [1] 64  
## [1] 65  
## [1] 66  
## [1] 67  
## [1] 68  
## [1] 69  
## [1] 70  
## [1] 71  
## [1] 72  
## [1] 73  
## [1] 74  
## [1] 75  
## [1] 76  
## [1] 77  
## [1] 78  
## [1] 79  
## [1] 80  
## [1] 81  
## [1] 82  
## [1] 83  
## [1] 84  
## [1] 85  
## [1] 86  
## [1] 87  
## [1] 88  
## [1] 89  
## [1] 90  
## [1] 91  
## [1] 92  
## [1] 93  
## [1] 94  
## [1] 95  
## [1] 96  
## [1] 97  
## [1] 98  
## [1] 99  
## [1] 100  
## [1] 101  
## [1] 102  
## [1] 103  
## [1] 104  
## [1] 105  
## [1] 106  
## [1] 107  
## [1] 108  
## [1] 109  
## [1] 110  
## [1] 111  
## [1] 112  
## [1] 113  
## [1] 114  
## [1] 115  
## [1] 116  
## [1] 117  
## [1] 118  
## [1] 119  
## [1] 120  
## [1] 121  
## [1] 122  
## [1] 123  
## [1] 124  
## [1] 125  
## [1] 126  
## [1] 127  
## [1] 128  
## [1] 129  
## [1] 130  
## [1] 131  
## [1] 132  
## [1] 133  
## [1] 134  
## [1] 135  
## [1] 136  
## [1] 137  
## [1] 138  
## [1] 139  
## [1] 140  
## [1] 141  
## [1] 142  
## [1] 143  
## [1] 144  
## [1] 145  
## [1] 146  
## [1] 147  
## [1] 148  
## [1] 149  
## [1] 150  
## [1] 151  
## [1] 152  
## [1] 153  
## [1] 154  
## [1] 155  
## [1] 156  
## [1] 157  
## [1] 158  
## [1] 159  
## [1] 160  
## [1] 161  
## [1] 162  
## [1] 163  
## [1] 164  
## [1] 165  
## [1] 166  
## [1] 167  
## [1] 168  
## [1] 169  
## [1] 170  
## [1] 171  
## [1] 172  
## [1] 173  
## [1] 174  
## [1] 175  
## [1] 176  
## [1] 177  
## [1] 178  
## [1] 179  
## [1] 180  
## [1] 181  
## [1] 182  
## [1] 183  
## [1] 184  
## [1] 185  
## [1] 186  
## [1] 187  
## [1] 188  
## [1] 189  
## [1] 190  
## [1] 191  
## [1] 192  
## [1] 193  
## [1] 194  
## [1] 195  
## [1] 196  
## [1] 197  
## [1] 198  
## [1] 199  
## [1] 200  
## [1] 201  
## [1] 202  
## [1] 203  
## [1] 204  
## [1] 205  
## [1] 206  
## [1] 207  
## [1] 208  
## [1] 209  
## [1] 210  
## [1] 211  
## [1] 212  
## [1] 213  
## [1] 214  
## [1] 215  
## [1] 216  
## [1] 217  
## [1] 218  
## [1] 219  
## [1] 220  
## [1] 221

colnames(beta\_EM)= gsub("Y","Beta", colnames(beta\_EM))  
print(paste("Estimated Beta=", beta\_EM))

## [1] "Estimated Beta= -6.67891967895741" "Estimated Beta= 0.578084343496613"   
## [3] "Estimated Beta= 8.35164819952728" "Estimated Beta= 1.61482271586225"   
## [5] "Estimated Beta= -5.525250089067" "Estimated Beta= -0.461658089374925"   
## [7] "Estimated Beta= 5.63114043962696" "Estimated Beta= 7.77897918499387"   
## [9] "Estimated Beta= 4.08274723535059" "Estimated Beta= -1.19267268497901"   
## [11] "Estimated Beta= -0.0677524814887373" "Estimated Beta= -2.0255633992556"   
## [13] "Estimated Beta= -3.61962836203075" "Estimated Beta= -6.72080745687412"   
## [15] "Estimated Beta= 6.32452146759976" "Estimated Beta= 0.878880559033346"   
## [17] "Estimated Beta= 1.20769634266936" "Estimated Beta= -8.36891072992992"   
## [19] "Estimated Beta= -5.3483566334213" "Estimated Beta= 1.97649137745432"   
## [21] "Estimated Beta= -1.67835409865876" "Estimated Beta= -3.81227832935702"   
## [23] "Estimated Beta= -2.17978238051758" "Estimated Beta= -5.00007015743628"   
## [25] "Estimated Beta= 4.50994400499595" "Estimated Beta= 3.35006271339294"   
## [27] "Estimated Beta= 4.28843887150907" "Estimated Beta= 8.67031833954333"   
## [29] "Estimated Beta= -0.834322703636711" "Estimated Beta= 3.80065985117203"

print(cbind(EstimatedValue=c(t(beta\_EM)),TrueValue=c(t(data\_lst$beta))))

## EstimatedValue TrueValue  
## [1,] -6.67891968 -6.6178593  
## [2,] -0.06775248 -0.3753387  
## [3,] -1.67835410 -1.4738338  
## [4,] 0.57808434 1.0952580  
## [5,] -2.02556340 -2.3403822  
## [6,] -3.81227833 -3.5314326  
## [7,] 8.35164820 7.9773300  
## [8,] -3.61962836 -3.9077075  
## [9,] -2.17978238 -2.3283126  
## [10,] 1.61482272 1.7843416  
## [11,] -6.72080746 -6.8717300  
## [12,] -5.00007016 -4.8753227  
## [13,] -5.52525009 -5.2795257  
## [14,] 6.32452147 6.1954139  
## [15,] 4.50994400 4.2782677  
## [16,] -0.46165809 -0.5396361  
## [17,] 0.87888056 1.1080453  
## [18,] 3.35006271 3.3945416  
## [19,] 5.63114044 5.3452597  
## [20,] 1.20769634 1.2814337  
## [21,] 4.28843887 4.0664180  
## [22,] 7.77897918 7.5886772  
## [23,] -8.36891073 -7.9460268  
## [24,] 8.67031834 8.7634763  
## [25,] 4.08274724 3.8685602  
## [26,] -5.34835663 -4.8139629  
## [27,] -0.83432270 -1.0107075  
## [28,] -1.19267268 -1.3235758  
## [29,] 1.97649138 2.2100069  
## [30,] 3.80065985 3.9251089

## Bayesian MCMC Algorithm(d=3):

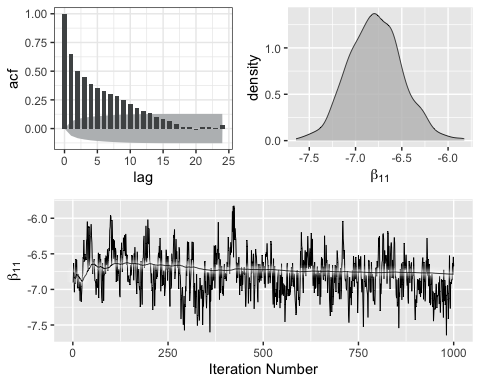
Here we obtain the posterior samples of the regression coefficients using the MCMC algorithm.

# Change Sample Size to get the full MCMC. MCMC step takes time depending on the sample size. This step can take 20 to 30 minutes. Prints output after every 100 samples are generated.  
lst=MCMC\_Dir\_regression\_sampler\_V1(Y=data\_lst$Y, X=data\_lst$X, prior=NULL, beta\_init = NULL, MCSamplerSize =Set\_MCSamplerSize)

## [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] 65  
## [1] 66  
## [1] 67  
## [1] 68  
## [1] 69  
## [1] 70  
## [1] 71  
## [1] 72  
## [1] 73  
## [1] 74  
## [1] 75  
## [1] 76  
## [1] 77  
## [1] 78  
## [1] 79  
## [1] 80  
## [1] 81  
## [1] 82  
## [1] 83  
## [1] 84  
## [1] 85  
## [1] 86  
## [1] 87  
## [1] 88  
## [1] 89  
## [1] 90  
## [1] 91  
## [1] 92  
## [1] 93  
## [1] 94  
## [1] 95  
## [1] 96  
## [1] 97  
## [1] 98  
## [1] 99  
## [1] 100  
## [1] 101  
## [1] 102  
## [1] 103  
## [1] 104  
## [1] 105  
## [1] 106  
## [1] 107  
## [1] 108  
## [1] 109  
## [1] 110  
## [1] 111  
## [1] 112  
## [1] 113  
## [1] 114  
## [1] 115  
## [1] 116  
## [1] 117  
## [1] 118  
## [1] 119  
## [1] 120  
## [1] 121  
## [1] 122  
## [1] 123  
## [1] 124  
## [1] 125  
## [1] 126  
## [1] 127  
## [1] 128  
## [1] 129  
## [1] 130  
## [1] 131  
## [1] 132  
## [1] 133  
## [1] 134  
## [1] 135  
## [1] 136  
## [1] 137  
## [1] 138  
## [1] 139  
## [1] 140  
## [1] 141  
## [1] 142  
## [1] 143  
## [1] 144  
## [1] 145  
## [1] 146  
## [1] 147  
## [1] 148  
## [1] 149  
## [1] 150  
## [1] 151  
## [1] 152  
## [1] 153  
## [1] 154  
## [1] 155  
## [1] 156  
## [1] 157  
## [1] 158  
## [1] 159  
## [1] 160  
## [1] 161  
## [1] 162  
## [1] 163  
## [1] 164  
## [1] 165  
## [1] 166  
## [1] 167  
## [1] 168  
## [1] 169  
## [1] 170  
## [1] 171  
## [1] 172  
## [1] 173  
## [1] 174  
## [1] 175  
## [1] 176  
## [1] 177  
## [1] 178  
## [1] 179  
## [1] 180  
## [1] 181  
## [1] 182  
## [1] 183  
## [1] 184  
## [1] 185  
## [1] 186  
## [1] 187  
## [1] 188  
## [1] 189  
## [1] 190  
## [1] 191  
## [1] 192  
## [1] 193  
## [1] 194  
## [1] 195  
## [1] 196  
## [1] 197  
## [1] 198  
## [1] 199  
## [1] 200  
## [1] 201  
## [1] 202  
## [1] 203  
## [1] 204  
## [1] 205  
## [1] 206  
## [1] 207  
## [1] 208  
## [1] 209  
## [1] 210  
## [1] 211  
## [1] 212  
## [1] 213  
## [1] 214  
## [1] 215  
## [1] 216  
## [1] 217  
## [1] 218  
## [1] 219  
## [1] 220  
## [1] 221  
## [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"

The triplet plot (autocorrelation, traceplot and density plot) for some of the regression coefficient is plotted (d=3)

# Summary from MCMC output  
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\_SD=apply( lst$MC$Mc\_Beta, MARGIN = c(2,3), FUN = sd)  
 print(cbind(Posterior\_mean=c(t(Posterior\_mean)),TrueValue=c(t(data\_lst$beta))))

## Posterior\_mean TrueValue  
## [1,] -6.78492616 -6.6178593  
## [2,] -0.07065325 -0.3753387  
## [3,] -1.71006030 -1.4738338  
## [4,] 0.59703459 1.0952580  
## [5,] -2.06178613 -2.3403822  
## [6,] -3.86783529 -3.5314326  
## [7,] 8.46893770 7.9773300  
## [8,] -3.65265844 -3.9077075  
## [9,] -2.22616904 -2.3283126  
## [10,] 1.63786163 1.7843416  
## [11,] -6.81129513 -6.8717300  
## [12,] -5.06147945 -4.8753227  
## [13,] -5.60012087 -5.2795257  
## [14,] 6.42534588 6.1954139  
## [15,] 4.57673176 4.2782677  
## [16,] -0.46434499 -0.5396361  
## [17,] 0.88339161 1.1080453  
## [18,] 3.40701937 3.3945416  
## [19,] 5.70940017 5.3452597  
## [20,] 1.23353493 1.2814337  
## [21,] 4.36296320 4.0664180  
## [22,] 7.88964814 7.5886772  
## [23,] -8.49823842 -7.9460268  
## [24,] 8.78694582 8.7634763  
## [25,] 4.14209925 3.8685602  
## [26,] -5.43421466 -4.8139629  
## [27,] -0.85166571 -1.0107075  
## [28,] -1.20215236 -1.3235758  
## [29,] 2.01097705 2.2100069  
## [30,] 3.86310525 3.9251089

## Simulated Data Generation and EM for Posterior Mode (d=10, Higher dimensional Spherical Data):

In this part of the demonstration, we generate a dataset of size n=750, with p=10 (10 covariates) and the responses are circular,i.e., d=2. Then we fit the EM algorithm to estimate the regression coefficients. True value of the regression coefficients, and its estimates are printed.

n=750 # NUmber of the samples  
p=10 # NUmber of the regression covariates  
d=10 # Number of direcions in the direcional data  
#### bbeta is a matrix of dimension p\times d  
#bbeta=matrix( rnorm(p\*d), nrow=p, ncol=d)  
sigma\_square=1  
tau\_square=10000  
data\_lst = Data\_generator\_vnf\_reg(n=n, p=p, d=d, concentration\_factor = 1, beta\_factor = 10)  
Y = data\_lst$Y;X=data\_lst$X;  
  
# Fitting the EM algorithm for the Standard Regresion for directional responses: This takes less than a minute.   
beta\_EM=EM\_Dir\_regression\_optimizer\_V1(Y=Y, X=X, prior=NULL, beta\_init = NULL, 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  
## [1] 61  
## [1] 62  
## [1] 63  
## [1] 64  
## [1] 65  
## [1] 66  
## [1] 67  
## [1] 68  
## [1] 69  
## [1] 70  
## [1] 71  
## [1] 72  
## [1] 73  
## [1] 74  
## [1] 75  
## [1] 76  
## [1] 77  
## [1] 78  
## [1] 79  
## [1] 80  
## [1] 81  
## [1] 82  
## [1] 83  
## [1] 84  
## [1] 85  
## [1] 86  
## [1] 87  
## [1] 88  
## [1] 89  
## [1] 90  
## [1] 91  
## [1] 92  
## [1] 93  
## [1] 94  
## [1] 95  
## [1] 96  
## [1] 97  
## [1] 98  
## [1] 99  
## [1] 100  
## [1] 101  
## [1] 102  
## [1] 103  
## [1] 104  
## [1] 105  
## [1] 106  
## [1] 107  
## [1] 108  
## [1] 109  
## [1] 110  
## [1] 111  
## [1] 112  
## [1] 113  
## [1] 114  
## [1] 115  
## [1] 116  
## [1] 117  
## [1] 118  
## [1] 119  
## [1] 120  
## [1] 121  
## [1] 122  
## [1] 123  
## [1] 124  
## [1] 125  
## [1] 126  
## [1] 127  
## [1] 128  
## [1] 129  
## [1] 130  
## [1] 131  
## [1] 132  
## [1] 133  
## [1] 134  
## [1] 135  
## [1] 136

colnames(beta\_EM)= gsub("Y","Beta", colnames(beta\_EM))  
print(paste("Estimated Beta=", beta\_EM))

## [1] "Estimated Beta= -1.81117847426189" "Estimated Beta= 0.686999394473401"   
## [3] "Estimated Beta= 0.193974811669485" "Estimated Beta= -1.87234545954101"   
## [5] "Estimated Beta= -1.58539280439524" "Estimated Beta= 7.73354056719241"   
## [7] "Estimated Beta= 3.34362051635042" "Estimated Beta= 5.70312562476308"   
## [9] "Estimated Beta= 8.53210596948993" "Estimated Beta= -5.6734246710189"   
## [11] "Estimated Beta= 3.01094603076965" "Estimated Beta= 5.73981464704966"   
## [13] "Estimated Beta= 7.86322091437474" "Estimated Beta= 1.19611548833422"   
## [15] "Estimated Beta= 2.37515505704142" "Estimated Beta= 5.78534310946627"   
## [17] "Estimated Beta= -1.92559204254612" "Estimated Beta= 6.58642338716844"   
## [19] "Estimated Beta= -3.95369865214388" "Estimated Beta= -2.49254817935345"   
## [21] "Estimated Beta= 3.91036609523622" "Estimated Beta= 7.23331121011117"   
## [23] "Estimated Beta= -3.33308496867639" "Estimated Beta= -3.33397433523189"   
## [25] "Estimated Beta= 1.09687891824249" "Estimated Beta= -6.76284761719998"   
## [27] "Estimated Beta= 4.23322954056419" "Estimated Beta= -6.58811481569782"   
## [29] "Estimated Beta= -8.85948945757705" "Estimated Beta= 0.166533089439158"   
## [31] "Estimated Beta= 6.2399799842558" "Estimated Beta= -5.96544640618299"   
## [33] "Estimated Beta= -9.67959473783857" "Estimated Beta= 9.28676658613405"   
## [35] "Estimated Beta= 7.88709469368994" "Estimated Beta= 5.64983742738945"   
## [37] "Estimated Beta= -8.63335573499706" "Estimated Beta= -0.847215633657721"  
## [39] "Estimated Beta= -5.42538391694987" "Estimated Beta= 0.794393691742267"   
## [41] "Estimated Beta= -4.0679364762166" "Estimated Beta= -2.35914731928433"   
## [43] "Estimated Beta= -8.08229612669511" "Estimated Beta= 9.63784168159801"   
## [45] "Estimated Beta= 8.24795304732821" "Estimated Beta= 8.5029773648228"   
## [47] "Estimated Beta= 2.00537480037031" "Estimated Beta= 6.93637053310389"   
## [49] "Estimated Beta= 4.82570346463644" "Estimated Beta= -2.41592009309803"   
## [51] "Estimated Beta= -8.93809691520147" "Estimated Beta= 8.65433197548452"   
## [53] "Estimated Beta= -0.132144741342933" "Estimated Beta= -6.22275538103519"   
## [55] "Estimated Beta= 1.335884560526" "Estimated Beta= 3.94246144327889"   
## [57] "Estimated Beta= -7.89968050100003" "Estimated Beta= 6.93098750195205"   
## [59] "Estimated Beta= 3.04731289796782" "Estimated Beta= 4.83517289514241"   
## [61] "Estimated Beta= -8.94570427397528" "Estimated Beta= -0.158096756604444"  
## [63] "Estimated Beta= -5.59004431075095" "Estimated Beta= -9.19363783160022"   
## [65] "Estimated Beta= 10.1813716282649" "Estimated Beta= 5.97305674805877"   
## [67] "Estimated Beta= -5.94541616385583" "Estimated Beta= 9.74328041241518"   
## [69] "Estimated Beta= -7.60382078230674" "Estimated Beta= 8.79009639862667"   
## [71] "Estimated Beta= 7.21710026530024" "Estimated Beta= -4.97024706190382"   
## [73] "Estimated Beta= 8.47557529556265" "Estimated Beta= -4.85191877385558"   
## [75] "Estimated Beta= 3.24289082558568" "Estimated Beta= -1.51759625979132"   
## [77] "Estimated Beta= -9.37504393583094" "Estimated Beta= 0.232677423156192"   
## [79] "Estimated Beta= -1.31073464831864" "Estimated Beta= 0.520371928698869"   
## [81] "Estimated Beta= -9.66794004554779" "Estimated Beta= 6.12502242762453"   
## [83] "Estimated Beta= 7.12159683821976" "Estimated Beta= 5.75559717142887"   
## [85] "Estimated Beta= 0.953191996752063" "Estimated Beta= -5.69649642965227"   
## [87] "Estimated Beta= 4.97229163985329" "Estimated Beta= -9.66629915908632"   
## [89] "Estimated Beta= -6.3882197795229" "Estimated Beta= -7.11940016491259"   
## [91] "Estimated Beta= -4.36998189356732" "Estimated Beta= -1.70502504744112"   
## [93] "Estimated Beta= -2.59907579985076" "Estimated Beta= 4.56903511646267"   
## [95] "Estimated Beta= 0.489484019403306" "Estimated Beta= 8.67571719842719"   
## [97] "Estimated Beta= 3.95043113084978" "Estimated Beta= -0.683956396732183"  
## [99] "Estimated Beta= -1.43989955935619" "Estimated Beta= -2.96081552570612"

print(cbind(EstimatedValue=c(t(beta\_EM)),TrueValue=c(t(data\_lst$beta))))

## EstimatedValue TrueValue  
## [1,] -1.8111785 -1.6698773  
## [2,] 3.0109460 3.5166189  
## [3,] 3.9103661 4.2548866  
## [4,] 6.2399800 5.9123033  
## [5,] -4.0679365 -4.1660718  
## [6,] -8.9380969 -9.3179983  
## [7,] -8.9457043 -9.4273595  
## [8,] 7.2171003 6.8788346  
## [9,] -9.6679400 -9.7381236  
## [10,] -4.3699819 -3.8936974  
## [11,] 0.6869994 0.7380284  
## [12,] 5.7398146 6.0411335  
## [13,] 7.2333112 7.1083025  
## [14,] -5.9654464 -5.8658903  
## [15,] -2.3591473 -2.2136283  
## [16,] 8.6543320 8.4367928  
## [17,] -0.1580968 -0.3551959  
## [18,] -4.9702471 -4.8945231  
## [19,] 6.1250224 5.9712232  
## [20,] -1.7050250 -1.8597362  
## [21,] 0.1939748 0.1675250  
## [22,] 7.8632209 7.8608981  
## [23,] -3.3330850 -3.1848677  
## [24,] -9.6795947 -9.6033902  
## [25,] -8.0822961 -7.6909486  
## [26,] -0.1321447 0.1658053  
## [27,] -5.5900443 -5.8948960  
## [28,] 8.4755753 8.4279143  
## [29,] 7.1215968 6.3521159  
## [30,] -2.5990758 -3.3002317  
## [31,] -1.8723455 -1.9484992  
## [32,] 1.1961155 1.0493206  
## [33,] -3.3339743 -3.7147376  
## [34,] 9.2867666 8.7291110  
## [35,] 9.6378417 9.6212789  
## [36,] -6.2227554 -6.3388244  
## [37,] -9.1936378 -9.0559942  
## [38,] -4.8519188 -4.4323506  
## [39,] 5.7555972 5.6481174  
## [40,] 4.5690351 4.6877488  
## [41,] -1.5853928 -1.3840989  
## [42,] 2.3751551 2.1877198  
## [43,] 1.0968789 0.9553325  
## [44,] 7.8870947 7.8349960  
## [45,] 8.2479530 8.0653042  
## [46,] 1.3358846 1.5011557  
## [47,] 10.1813716 9.9877867  
## [48,] 3.2428908 3.1901197  
## [49,] 0.9531920 0.9961095  
## [50,] 0.4894840 0.4026503  
## [51,] 7.7335406 7.2653239  
## [52,] 5.7853431 5.3569360  
## [53,] -6.7628476 -6.3511406  
## [54,] 5.6498374 5.7366700  
## [55,] 8.5029774 8.2418984  
## [56,] 3.9424614 3.7340688  
## [57,] 5.9730567 5.8368395  
## [58,] -1.5175963 -1.1193659  
## [59,] -5.6964964 -5.7101099  
## [60,] 8.6757172 8.2333657  
## [61,] 3.3436205 3.6809960  
## [62,] -1.9255920 -1.6520009  
## [63,] 4.2332295 4.2710378  
## [64,] -8.6333557 -8.2018146  
## [65,] 2.0053748 2.2482077  
## [66,] -7.8996805 -8.0248900  
## [67,] -5.9454162 -6.2213572  
## [68,] -9.3750439 -9.3094478  
## [69,] 4.9722916 4.9238253  
## [70,] 3.9504311 3.8953592  
## [71,] 5.7031256 5.4890893  
## [72,] 6.5864234 6.2441135  
## [73,] -6.5881148 -6.4095039  
## [74,] -0.8472156 -0.8280865  
## [75,] 6.9363705 6.6761824  
## [76,] 6.9309875 7.1692123  
## [77,] 9.7432804 9.5585265  
## [78,] 0.2326774 0.4716570  
## [79,] -9.6662992 -9.6372148  
## [80,] -0.6839564 -0.7469148  
## [81,] 8.5321060 8.1294332  
## [82,] -3.9536987 -3.6458746  
## [83,] -8.8594895 -8.3984274  
## [84,] -5.4253839 -5.8834703  
## [85,] 4.8257035 5.1217755  
## [86,] 3.0473129 3.0397043  
## [87,] -7.6038208 -7.3058516  
## [88,] -1.3107346 -1.2474422  
## [89,] -6.3882198 -6.1530189  
## [90,] -1.4398996 -1.4697712  
## [91,] -5.6734247 -5.7858508  
## [92,] -2.4925482 -2.5723408  
## [93,] 0.1665331 -0.1852511  
## [94,] 0.7943937 1.0360707  
## [95,] -2.4159201 -2.9805594  
## [96,] 4.8351729 4.6447452  
## [97,] 8.7900964 8.7107261  
## [98,] 0.5203719 0.4167397  
## [99,] -7.1194002 -7.3275187  
## [100,] -2.9608155 -2.6246257

## Bayesian MCMC Algorithm(d=10):

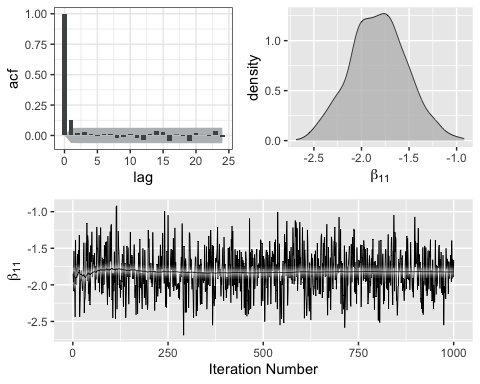
Here we obtain the posterior samples of the regression coefficients using the MCMC algorithm.

# Change Sample Size to get the full MCMC. MCMC step takes time depending on the sample size. This step can take 20 to 30 minutes. Prints output after every 100 samples are generated.  
lst=MCMC\_Dir\_regression\_sampler\_V1(Y=data\_lst$Y, X=data\_lst$X, prior=NULL, beta\_init = NULL, MCSamplerSize =Set\_MCSamplerSize)

## [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] 65  
## [1] 66  
## [1] 67  
## [1] 68  
## [1] 69  
## [1] 70  
## [1] 71  
## [1] 72  
## [1] 73  
## [1] 74  
## [1] 75  
## [1] 76  
## [1] 77  
## [1] 78  
## [1] 79  
## [1] 80  
## [1] 81  
## [1] 82  
## [1] 83  
## [1] 84  
## [1] 85  
## [1] 86  
## [1] 87  
## [1] 88  
## [1] 89  
## [1] 90  
## [1] 91  
## [1] 92  
## [1] 93  
## [1] 94  
## [1] 95  
## [1] 96  
## [1] 97  
## [1] 98  
## [1] 99  
## [1] 100  
## [1] 101  
## [1] 102  
## [1] 103  
## [1] 104  
## [1] 105  
## [1] 106  
## [1] 107  
## [1] 108  
## [1] 109  
## [1] 110  
## [1] 111  
## [1] 112  
## [1] 113  
## [1] 114  
## [1] 115  
## [1] 116  
## [1] 117  
## [1] 118  
## [1] 119  
## [1] 120  
## [1] 121  
## [1] 122  
## [1] 123  
## [1] 124  
## [1] 125  
## [1] 126  
## [1] 127  
## [1] 128  
## [1] 129  
## [1] 130  
## [1] 131  
## [1] 132  
## [1] 133  
## [1] 134  
## [1] 135  
## [1] 136  
## [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"

The triplet plot (autocorrelation, traceplot and density plot) for some of the regression coefficient is plotted (d=10)

# Summary from MCMC output  
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\_SD=apply( lst$MC$Mc\_Beta, MARGIN = c(2,3), FUN = sd)  
 print(cbind(Estimated\_Value=c(t(Posterior\_Mean)),TrueValue=c(t(data\_lst$beta))))

## Estimated\_Value TrueValue  
## [1,] -1.8228648 -1.6698773  
## [2,] 3.0410726 3.5166189  
## [3,] 3.9765018 4.2548866  
## [4,] 6.3140142 5.9123033  
## [5,] -4.1009728 -4.1660718  
## [6,] -9.0194666 -9.3179983  
## [7,] -9.0014269 -9.4273595  
## [8,] 7.2835808 6.8788346  
## [9,] -9.7705382 -9.7381236  
## [10,] -4.4262244 -3.8936974  
## [11,] 0.6826268 0.7380284  
## [12,] 5.7889192 6.0411335  
## [13,] 7.3058593 7.1083025  
## [14,] -5.9896508 -5.8658903  
## [15,] -2.3625829 -2.2136283  
## [16,] 8.7479008 8.4367928  
## [17,] -0.1647828 -0.3551959  
## [18,] -5.0145872 -4.8945231  
## [19,] 6.1620240 5.9712232  
## [20,] -1.7278005 -1.8597362  
## [21,] 0.1972972 0.1675250  
## [22,] 7.9277550 7.8608981  
## [23,] -3.3619512 -3.1848677  
## [24,] -9.7620462 -9.6033902  
## [25,] -8.1715391 -7.6909486  
## [26,] -0.1364442 0.1658053  
## [27,] -5.6381369 -5.8948960  
## [28,] 8.5472831 8.4279143  
## [29,] 7.1809232 6.3521159  
## [30,] -2.6037871 -3.3002317  
## [31,] -1.8765600 -1.9484992  
## [32,] 1.2149882 1.0493206  
## [33,] -3.3631023 -3.7147376  
## [34,] 9.3768653 8.7291110  
## [35,] 9.7254624 9.6212789  
## [36,] -6.2916360 -6.3388244  
## [37,] -9.2704062 -9.0559942  
## [38,] -4.9041075 -4.4323506  
## [39,] 5.8351493 5.6481174  
## [40,] 4.6137549 4.6877488  
## [41,] -1.6095488 -1.3840989  
## [42,] 2.3877166 2.1877198  
## [43,] 1.1191216 0.9553325  
## [44,] 7.9648975 7.8349960  
## [45,] 8.2966043 8.0653042  
## [46,] 1.3360031 1.5011557  
## [47,] 10.2758546 9.9877867  
## [48,] 3.2833935 3.1901197  
## [49,] 0.9480173 0.9961095  
## [50,] 0.4849897 0.4026503  
## [51,] 7.8018460 7.2653239  
## [52,] 5.8435111 5.3569360  
## [53,] -6.8095239 -6.3511406  
## [54,] 5.6897278 5.7366700  
## [55,] 8.5732201 8.2418984  
## [56,] 3.9743939 3.7340688  
## [57,] 6.0201140 5.8368395  
## [58,] -1.5385674 -1.1193659  
## [59,] -5.7346319 -5.7101099  
## [60,] 8.7519837 8.2333657  
## [61,] 3.3734474 3.6809960  
## [62,] -1.9485349 -1.6520009  
## [63,] 4.2722160 4.2710378  
## [64,] -8.6977322 -8.2018146  
## [65,] 2.0298822 2.2482077  
## [66,] -7.9702088 -8.0248900  
## [67,] -5.9926066 -6.2213572  
## [68,] -9.4551567 -9.3094478  
## [69,] 5.0066851 4.9238253  
## [70,] 3.9933514 3.8953592  
## [71,] 5.7410207 5.4890893  
## [72,] 6.6424978 6.2441135  
## [73,] -6.6473234 -6.4095039  
## [74,] -0.8558160 -0.8280865  
## [75,] 6.9863047 6.6761824  
## [76,] 6.9989964 7.1692123  
## [77,] 9.8476146 9.5585265  
## [78,] 0.2277516 0.4716570  
## [79,] -9.7557425 -9.6372148  
## [80,] -0.6955992 -0.7469148  
## [81,] 8.6257926 8.1294332  
## [82,] -3.9883014 -3.6458746  
## [83,] -8.9317016 -8.3984274  
## [84,] -5.4761047 -5.8834703  
## [85,] 4.8841286 5.1217755  
## [86,] 3.0771167 3.0397043  
## [87,] -7.6362626 -7.3058516  
## [88,] -1.3149674 -1.2474422  
## [89,] -6.4423160 -6.1530189  
## [90,] -1.4580915 -1.4697712  
## [91,] -5.7076238 -5.7858508  
## [92,] -2.5080279 -2.5723408  
## [93,] 0.1712663 -0.1852511  
## [94,] 0.7980005 1.0360707  
## [95,] -2.4283469 -2.9805594  
## [96,] 4.8886167 4.6447452  
## [97,] 8.8929419 8.7107261  
## [98,] 0.5387310 0.4167397  
## [99,] -7.1826204 -7.3275187  
## [100,] -2.9875550 -2.6246257

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

## [1] "Total Run Time=24.4914363503456"