A simulation study to demonstrate the biases in the diagnoses of mental illnesses: major depressive episodes, dysthymia, and manic episodes

Yi-Sheng Chao[1*], Kuan-Fu Lin,[2] Chao-Jung Wu[3], Hsing-Chien Wu[4], Hui-Ting Hsu[5], Lien-Cheng Tsao[5], Yen-Po Cheng[5], Yi-Chun Lai[6], Wei-Chih Chen[7,8]

[1]Independent researcher, Montréal, H2X 0A8 Canada, [2]National Taiwan University Hospital Yun-Lin Branch, Yunlin County, 640 Taiwan, [3]Département d'informatique Université du Québec à Montréal, Montréal H3B 1B4 Canada, [4]Taipei Hospital Ministry of Health and Welfare New Taipei city, 242 Taiwan, [5]Changhua Christian Hospital, Changhua County 526, Taiwan, [6]National Yang-Ming University Hospital, Yilan 260 Taiwan, [7]Department of Chest Medicine, Taipei Veterans General Hospital, Taipei 112, Taiwan, [8]Institute of Emergency and Critical Care Medicine, National Yang-Ming University, Taipei 112, Taiwan

[*]chaoyisheng@post.harvard.edu

Keywords: Frailty; bias; forward-stepwise regression; the Health and Retirement Study; index mining

Mental illnes	Illness variab	Classification	Criterion vari	Major or min	Domain varia	Symptoms	Symptom var	Equations	Approximatic	=	Assumed pre	)	Derived prev	definition	variable	R	formula	code	outcome	suboutcome
Major Depre	mde							mde = mde_r	mde = intercept + coef1 x mde_ma1 + coef2 x mde_ma2 + coef3 x					Major Depre	mde		mde = mde_r	sim$mde = si	mde	
		Major criteria, essential for diagnosis												Major criteri						
				Depressed mood or a loss of interest or pleasure in daily activities for more than two weeks.										Depressed mood or a loss of interest or pleasure in daily activities for more than two weeks.						
				Depressed m	mde_ma1					= rbinom(n=	0.4	)		Depressed m	mde_ma1		mde_ma1 = r	sim$mde_ma	mde	
				Loss of intere	mde_ma2					= rbinom(n=	0.4	)		Loss of intere	mde_ma2		mde_ma2 = r	sim$mde_ma	mde	
		Minor criteri	mde_mi											Minor criteria (at least 5 of the symptoms including the two in major criteria)						
				Significant ur	mde_mi3			mde_mi3 = mde_mi3_1 + mde_mi3_2 + mde_mi3_bias						Significant ur	mde_mi3		mde_mi3 = 1*((mde_mi3_		mde	mde_mi3
						Significant ur	mde_mi3_1			= rbinom(n=	0.4	)		Significant ur	mde_mi3_1		mde_mi3_1 =	sim$mde_mi	mde	mde_mi3
						Significant ur	mde_mi3_2			= rbinom(n=	0.4	)		Significant ur	mde_mi3_2		mde_mi3_2 =	sim$mde_mi	mde	mde_mi3
																		sim$mde_mi3 = 1*((sim$mde_mi3_1 + sim$mde_mi3_2) > 0)		
						Information c	mde_mi3_bias							Information c	mde_mi3_bias		mde_mi3_bia	sim$mde_mi	mde	mde_mi3
				Insomnia or s	mde_mi4			mde_mi4 = mde_mi4_1 + mde_mi4_2 + mde_mi4_bias						Insomnia or s	mde_mi4		mde_mi4 = 1*((mde_mi4_		mde	mde_mi4
						Insomnia	mde_mi4_1			= rbinom(n=	0.4	)		Insomnia	mde_mi4_1		mde_mi4_1 =	sim$mde_mi	mde	mde_mi4
						Sleeping too	mde_mi4_2			= rbinom(n=	0.4	)		Sleeping too	mde_mi4_2		mde_mi4_2 =	sim$mde_mi	mde	mde_mi4
																		sim$mde_mi4 = 1*((sim$mde_mi4_1 + sim$mde_mi4_2) > 0)		
						Information c	mde_mi4_bias							Information c	mde_mi4_bias		mde_mi4_bia	sim$mde_mi	mde	mde_mi4
				Agitation or p	mde_mi5			mde_mi5 = mde_mi5_1 + mde_mi5_2 + mde_mi5_bias						Agitation or p	mde_mi5		mde_mi5 = 1*((mde_mi5_		mde	mde_mi5
						Agitation	mde_mi5_1			= rbinom(n=	0.4	)		Agitation	mde_mi5_1		mde_mi5_1 =	sim$mde_mi	mde	mde_mi5
						Psychomotor	mde_mi5_2			= rbinom(n=	0.4	)		Psychomotor	mde_mi5_2		mde_mi5_2 =	sim$mde_mi	mde	mde_mi5
																		sim$mde_mi5 = 1*((sim$mde_mi5_1 + sim$mde_mi5_2) > 0)		
						Information c	mde_mi5_bias							Information c	mde_mi5_bias		mde_mi5_bia	sim$mde_mi	mde	mde_mi5
				Fatigue or los	mde_mi6			mde_mi6 = mde_mi6_1 + mde_mi6_2 + mde_mi6_bias						Fatigue or los	mde_mi6		mde_mi6 = 1*((mde_mi6_		mde	mde_mi6
						Fatigue	mde_mi6_1			= rbinom(n=	0.4	)		Fatigue	mde_mi6_1		mde_mi6_1 =	sim$mde_mi	mde	mde_mi6
						Loss of energ	mde_mi6_2			= rbinom(n=	0.4	)		Loss of energ	mde_mi6_2		mde_mi6_2 =	sim$mde_mi	mde	mde_mi6
																		sim$mde_mi6 = 1*((sim$mde_mi6_1 + sim$mde_mi6_2) > 0)		
						Information c	mde_mi6_bias							Information c	mde_mi6_bias		mde_mi6_bia	sim$mde_mi	mde	mde_mi6
				Feelings of w	mde_mi7			mde_mi7 = mde_mi7_1 + mde_mi7_2 + mde_mi7_bias						Feelings of w	mde_mi7		mde_mi7 = 1*((mde_mi7_		mde	mde_mi7
						Feelings of w	mde_mi7_1			= rbinom(n=	0.4	)		Feelings of w	mde_mi7_1		mde_mi7_1 =	sim$mde_mi	mde	mde_mi7
						Feelings of e	mde_mi7_2			= rbinom(n=	0.4	)		Feelings of e	mde_mi7_2		mde_mi7_2 =	sim$mde_mi	mde	mde_mi7
																		sim$mde_mi7 = 1*((sim$mde_mi7_1 + sim$mde_mi7_2) > 0)		
						Information c	mde_mi7_bias							Information c	mde_mi7_bias		mde_mi7_bia	sim$mde_mi	mde	mde_mi7
				Diminished a	mde_mi8			mde_mi8 = mde_mi8_1 + mde_mi8_2 + mde_mi8_bias						Diminished a	mde_mi8		mde_mi8 = 1*((mde_mi8_		mde	mde_mi8
						Diminished a	mde_mi8_1			= rbinom(n=	0.4	)		Diminished a	mde_mi8_1		mde_mi8_1 =	sim$mde_mi	mde	mde_mi8
						Indecisivenes	mde_mi8_2			= rbinom(n=	0.4	)		Indecisivenes	mde_mi8_2		mde_mi8_2 =	sim$mde_mi	mde	mde_mi8
																		sim$mde_mi8 = 1*((sim$mde_mi8_1 + sim$mde_mi8_2) > 0)		
						Information c	mde_mi8_bias							Information c	mde_mi8_bias		mde_mi8_bia	sim$mde_mi	mde	mde_mi8
				Recurrent th	mde_mi9					= rbinom(n=	0.4	)		Recurrent th	mde_mi9		mde_mi9 = rl	sim$mde_mi	mde	
				Information of the minor criteria not expained by the major or minir criteria										Information of the minor criteria not expained by the major or minir criteria						
																		sim$mde = sim$mde_ma1 * sim$mde_ma2 * (sim$mde_mi3 + sim$mde_mi4 + sim$mde_mi5 + sim$mde_mi6 + sim$mde_mi7 + sim$mde_mi8 + sim$mde_mi9 + sim$mde_bias1) + (1- sim$mde_ma1 * sim$mde_ma2) * (mde_ma1 * sim$mde_ma2) * (sim$mde_mi3 + sim$mde_mi4 + sim$mde_mi5 + sim$mde_mi6 + sim$mde_mi7 + sim$mde_mi8 + sim$mde_mi9 + sim$mde_bias2)		
		Information c	mde_bias1											Information c	mde_bias1		mde_bias1 =	sim$mde_bia	mde	
		Information c	mde_bias2											Information c	mde_bias2		mde_bias2 =	sim$mde_bia	mde	
		Information c	mde_bias											Information c	mde_bias		mde_bias = n	sim$mde_bia	mde	
Dysthymic Di	dys							dys = dys_ma	dys = intercept + coef1 x dys_ma + coef2 x dys_mi + coef3 x dys_b					Dysthymic Di	dys		dys = dys_ma	sim$dys = sin	dys	
		Major criteria, essential for diagnosis												Major criteri						
			dys_ma	Depressed mood most of the day for more days than not, for at least 2 years						= rbinom(n=	0.4	)		Depressed m	dys_ma		dys_ma = rbi	sim$dys_ma	dys	
		Minor criteri	dys_mi					dys_mi = dys_mi1 + dys_mi2 + dys_mi3 + dys_mi4 + dys_mi5 + dys_mi6 + dys_m						Minor criteri	dys_mi		dys_mi = 1*(	sim$dys_mi =	dys	
				Poor appetit	dys_mi1			dys_mi1 = dys_mi1_1 + dys_mi1_2 + dys_mi1_bias						Poor appetit	dys_mi1		dys_mi1 = 1*((dys_mi1_1 +		dys	dys_mi1
						Poor appetit	dys_mi1_1			= rbinom(n=	0.4	)		Poor appetit	dys_mi1_1		dys_mi1_1 =	sim$dys_mi1	dys	dys_mi1
						Overeating	dys_mi1_2			= rbinom(n=	0.4	)		Overeating	dys_mi1_2		dys_mi1_2 =	sim$dys_mi1	dys	dys_mi1
																		sim$dys_mi1 = 1*((sim$dys_mi1_1 + sim$dys_mi1_2) > 0)		
						Information c	dys_mi1_bias							Information c	dys_mi1_bias		dys_mi1_bia	sim$dys_mi1	dys	dys_mi1
				Insomnia or s	dys_mi2			dys_mi2 = mde_mi4 = mde_mi4_1 + mde_mi4_2 + mde_mi4_bias						Insomnia or s	mde_mi4		mde_mi4 = 1*((mde_mi4_		dys	
						Insomnia	mde_mi4_1			= rbinom(n=	0.4	)		Insomnia	mde_mi4_1		mde_mi4_1 =	sim$mde_mi	dys	
						Sleeping too	mde_mi4_2			= rbinom(n=	0.4	)		Sleeping too	mde_mi4_2		mde_mi4_2 =	sim$mde_mi	dys	
																		sim$mde_mi4 = 1*((sim$mde_mi4_1 + sim$mde_mi4_2) > 0)		
						Information c	mde_mi4_bias							Information c	mde_mi4_bias		mde_mi4_bia	sim$mde_mi	dys	
				Low energy c	dys_mi3			dys_mi3 = mde_mi6 = mde_mi6_1 + mde_mi6_2 + mde_mi6_bias						Low energy c	mde_mi6		mde_mi6 = 1*((mde_mi6_		dys	
						Fatigue	mde_mi6_1			= rbinom(n=	0.4	)		Fatigue	mde_mi6_1		mde_mi6_1 =	sim$mde_mi	dys	
						Loss of energ	mde_mi6_2			= rbinom(n=	0.4	)		Loss of energ	mde_mi6_2		mde_mi6_2 =	sim$mde_mi	dys	
																		sim$mde_mi6 = 1*((sim$mde_mi6_1 + sim$mde_mi6_2) > 0)		
						Information c	mde_mi6_bias							Information c	mde_mi6_bias		mde_mi6_bia	sim$mde_mi	dys	
				Low self-este	dys_mi4					= rbinom(n=	0.4	)		Low self-este	dys_mi4		dys_mi4 = rb	sim$dys_mi4	dys	
				Poor concent	dys_mi5			dys_mi5 = mde_mi8 = mde_mi8_1 + mde_mi8_2 + mde_mi8_bias						Poor concent	mde_mi8		mde_mi8 = 1*((mde_mi8_		dys	
						Diminished a	mde_mi8_1			= rbinom(n=	0.4	)		Diminished a	mde_mi8_1		mde_mi8_1 =	sim$mde_mi	dys	
						difficulty mal	mde_mi8_2			= rbinom(n=	0.4	)		difficulty mal	mde_mi8_2		mde_mi8_2 =	sim$mde_mi	dys	
																		sim$mde_mi8 = 1*((sim$mde_mi8_1 + sim$mde_mi8_2) > 0)		
						Information c	mde_mi8_bias							Information c	mde_mi8_bias		mde_mi8_bia	sim$mde_mi	dys	
				Feelings of h	dys_mi6					= rbinom(n=	0.4	)		Feelings of h	dys_mi6		dys_mi6 = rb	sim$dys_mi6	dys	
				Information c	dys_mi_bias									Information c	dys_mi_bias		dys_mi_bias	sim$dys_mi_	dys	
																		sim$dys = sim$dys_ma * sim$dys_mi		
		Information c	dys_bias											Information c	dys_bias		dys_bias = dy	sim$dys_bias	dys	
Manic episoc	manic							manic = (1- r	manic = intercept + coef1 x man_ma1 + coef2 x man_ma2 + coef3 :					Manic episoc	manic		manic = (1- r	sim$manic =	dys	
		Major criteria, essential for the diagnosis of a manic episode (more than one bipolar episode required to diagnose bipolar disorder)												Major criteri						
				A distinct period of abnormally and persistently elevated, expansive, or irritable mood, lasting at least 1 week (or any duration if hosp										A distinct per						
						Elevated moo	man_ma1			= rbinom(n=	0.1	)		Elevated moo	man_ma1		man_ma1 = r	sim$man_ma	manic	
						Expansive mo	man_ma2			= rbinom(n=	0.1	)		Expansive mo	man_ma2		man_ma2 = r	sim$man_ma	manic	
						Irritable moo	man_ma3			= rbinom(n=	0.1	)		Irritable moo	man_ma3		man_ma3 = r	sim$man_ma	manic	
		Minor criteria (3 or more of the following symptoms have persisted; 4 if the mood is only irritable)												Minor criteri						
				Increased sel	man_mi1			man_mi1 = man_mi1_1 + man_mi1_2 + man_mi1_bias						Increased sel	man_mi1		man_mi1 = 1*((man_mi1_		manic	man_mi1
						Increased sel	man_mi1_1			= rbinom(n=	0.1	)		Increased sel	man_mi1_1		man_mi1_1 =	sim$man_mi	manic	man_mi1
						Grandiosity	man_mi1_2			= rbinom(n=	0.1	)		Grandiosity	man_mi1_2		man_mi1_2 =	sim$man_mi	manic	man_mi1
																		sim$man_mi1 = 1*((sim$man_mi1_1 + sim$man_mi1_2) >0)		
						Information c	man_mi1_bias							Information c	man_mi1_bias		man_mi1_bia	sim$man_mi	manic	man_mi1
				Decreased ne	man_mi2					= rbinom(n=	0.1	)		Decreased ne	man_mi2		man_mi2 = rl	sim$man_mi	manic	
				More talkati	man_mi3			man_mi3 = man_mi3_1 + man_mi3_2 + man_mi3_bias						More talkati	man_mi3		man_mi3 = 1*((man_mi3_		manic	man_mi3
						More talkati	man_mi3_1			= rbinom(n=	0.1	)		More talkati	man_mi3_1		man_mi3_1 =	sim$man_mi	manic	man_mi3
						Pressure to k	man_mi3_2			= rbinom(n=	0.1	)		Pressure to k	man_mi3_2		man_mi3_2 =	sim$man_mi	manic	man_mi3
																		sim$man_mi3 = 1*((sim$man_mi3_1 + sim$man_mi3_2) > 0)		
						Information c	man_mi3_bias							Information c	man_mi3_bias		man_mi3_bia	sim$man_mi	manic	man_mi3
				Flight of idea	man_mi4			man_mi4 = man_mi4_1 + man_mi4_2 + man_mi4_bias						Flight of idea	man_mi4		man_mi4 = 1*((man_mi4_		manic	man_mi4
						Flight of idea	man_mi4_1			= rbinom(n=	0.1	)		Flight of idea	man_mi4_1		man_mi4_1 =	sim$man_mi	manic	man_mi4
						Subjective ex	man_mi4_2			= rbinom(n=	0.1	)		Subjective ex	man_mi4_2		man_mi4_2 =	sim$man_mi	manic	man_mi4
																		sim$man_mi4 = 1*((sim$man_mi4_1 + sim$man_mi4_2) > 0)		
						Information c	man_mi4_bias							Information c	man_mi4_bias		man_mi4_bia	sim$man_mi	manic	man_mi4
				Distractibility	man_mi5					= rbinom(n=	0.1	)		Distractibility	man_mi5		man_mi5 = rl	sim$man_mi	manic	
				Increase in g	man_mi6			man_mi6 = man_mi6_1 + man_mi6_2 + man_mi6_bias						Increase in g	man_mi6		man_mi6 = 1*((man_mi6_		manic	man_mi6
						Increase in g	man_mi6_1			= rbinom(n=	0.1	)		Increase in g	man_mi6_1		man_mi6_1 =	sim$man_mi	manic	man_mi6
						Psychomotor	man_mi6_2			= rbinom(n=	0.1	)		Psychomotor	man_mi6_2		man_mi6_2 =	sim$man_mi	manic	man_mi6
																		sim$man_mi6 = 1*((sim$man_mi6_1 + sim$man_mi6_2) > 0)		
						Information c	man_mi6_bias							Information c	man_mi6_bias		man_mi6_bia	sim$man_mi	manic	man_mi6
				Excessive inv	man_mi7					= rbinom(n=	0.1	)		Excessive inv	man_mi7		man_mi7 = rl	sim$man_mi	manic	
																		sim$sim$manic = (1- sim$man_ma1 * sim$man_ma2) * (sim$man_ma1 + sim$man_ma2) * sim$man_ma3 * (sim$man_mi1 + sim$man_mi2 + sim$man_mi3 + sim$man_mi4 + sim$man_mi5 + sim$man_mi6 + sim$man_mi7 + sim$man_bias1) + [1 - (1 - sim$man_ma1 * sim$man_ma2)(sim$man_ma1 + sim$man_ma2)] * sim$man_ma3 * (sim$man_mi1 + sim$man_mi2 + sim$man_mi3 + sim$man_mi4 + sim$man_mi5 + sim$man_mi6 + sim$man_mi7 + sim$man_bias2)		
		Information c	man_bias1											Information c	man_bias1		man_bias1 =	sim$man_bia	manic	
		Information c	man_bias2											Information c	man_bias2		man_bias2 =	sim$man_bia	manic	
		Information c	man_bias											Information c	man_bias		man_bias = n	sim$man_bia	manic	
			female												female		female = rbin	sim$female =	mde	
			age												age		age = sample	sim$age = sa	mde	
			edu												edu		edu = rnorm(	sim$edu = rn	mde	
																	edu[which(si	sim$edu[which(sim$edu <= 0)] = 0		

~~~
---
title: "2019_09_06 simulated mental illnesses"
author: "Yi-Sheng Chao"
date: "November 22, 2018"
output: pdf_document
editor_options:
  chunk_output_type: inline
---


##Adding correlations to the random variables

```{r}
library(bindata)

library(openxlsx)
resu = read.xlsx("A simulation study to demonstrate the biases in three
diagnoses of mental illnesses.xlsx", sheet = "Prob 1")
names(resu)
unique(resu$variable)
memory.limit(size = 10^13)
ssize = 10^5
times = 10^2

prevalence = c(0.05, 0.1, 0.3, 0.5, 0.7)
rho = c(0, 0.1, 0.4, 0.7, 0.9)#correlation coefficients of the input
symptoms

collect = c("mean", "max",
"min","derivedprevalence","coef","coefse","p","intercept",
"interceptp","r2", "subcoef","subcoefse","subp","subintercept",
"subinterceptp","subr2", "appbyownr2", "appbybiasr2", "appbyallr2",
"appbyownvar", "appbybiasvar", "appbyallvar", "appbyownn", "appbybiasn",
"appbyalln")


set.seed(1)


##Create a simulated data set to extract variables
for(preval in 1:length(prevalence)){
  for(rh in 1:length(rho)){


  library(openxlsx)
resu = read.xlsx("A simulation study to demonstrate the biases in three
diagnoses of mental illnesses.xlsx", sheet = "Prob 1")

    # foreach(c = 1:times) %dopar% {
    for(c in 1:times){

library(bindata)
bindata = as.data.frame(rmvbin(ssize, rep(prevalence[preval], 40),
bincorr=(1 - rho[rh])*diag(40) + rho[rh]))
bindata2 = as.data.frame(rmvbin(ssize, rep(prevalence[preval], 20),
bincorr=(1 - rho[rh])*diag(20) + rho[rh]))

##demographic characteristics
~~~

```
sim = data.frame(1:ssize)
names(sim) = "id"
sim$female = rbinom(n = ssize, size = 1, prob = 0.51)
sim$age = sample(30:60, ssize, replace = TRUE)
sim$edu = rnorm(ssize, mean = 12, sd = 5)
sim$edu[which(sim$edu <= 0)] = 0
sim$id = NULL


sim$mde_ma1 = bindata[,1]
sim$mde_ma2 = bindata[,2]

sim$mde_mi3_1 = bindata[,3]
sim$mde_mi3_2 = bindata[,4]
sim$mde_mi3 = 1*((sim$mde_mi3_1 + sim$mde_mi3_2) > 0)
sim$mde_mi3_bias = sim$mde_mi3 - sim$mde_mi3_1 - sim$mde_mi3_2

sim$mde_mi4_1 = bindata[,5]
sim$mde_mi4_2 = bindata[,6]
sim$mde_mi4 = 1*((sim$mde_mi4_1 + sim$mde_mi4_2) > 0)
sim$mde_mi4_bias = sim$mde_mi4 - sim$mde_mi4_1 - sim$mde_mi4_2

sim$mde_mi5_1 = bindata[,7]
sim$mde_mi5_2 = bindata[,8]
sim$mde_mi5 = 1*((sim$mde_mi5_1 + sim$mde_mi5_2) > 0)
sim$mde_mi5_bias = sim$mde_mi5 - sim$mde_mi5_1 - sim$mde_mi5_2

sim$mde_mi6_1 = bindata[,9]
sim$mde_mi6_2 = bindata[,10]
sim$mde_mi6 = 1*((sim$mde_mi6_1 + sim$mde_mi6_2) > 0)
sim$mde_mi6_bias = sim$mde_mi6 - sim$mde_mi6_1 - sim$mde_mi6_2

sim$mde_mi7_1 = bindata[,11]
sim$mde_mi7_2 = bindata[,12]
sim$mde_mi7 = 1*((sim$mde_mi7_1 + sim$mde_mi7_2) > 0)
sim$mde_mi7_bias = sim$mde_mi7 - sim$mde_mi7_1 - sim$mde_mi7_2

sim$mde_mi8_1 = bindata[,13]
sim$mde_mi8_2 = bindata[,14]
sim$mde_mi8 = 1*((sim$mde_mi8_1 + sim$mde_mi8_2) > 0)
sim$mde_mi8_bias = sim$mde_mi8 - sim$mde_mi8_1 - sim$mde_mi8_2

sim$mde_mi9 = bindata[,15]

sim$mde_bias1 = 1 * ((sim$mde_mi3 + sim$mde_mi4 + sim$mde_mi5 +
sim$mde_mi6 + sim$mde_mi7 + sim$mde_mi8 + sim$mde_mi9)>2) - (sim$mde_mi3
+ sim$mde_mi4 + sim$mde_mi5 + sim$mde_mi6 + sim$mde_mi7 + sim$mde_mi8 +
sim$mde_mi9)
sim$mde_bias2 = 1 * ((sim$mde_mi3 + sim$mde_mi4 + sim$mde_mi5 +
sim$mde_mi6 + sim$mde_mi7 + sim$mde_mi8 + sim$mde_mi9)>3) - (sim$mde_mi3
+ sim$mde_mi4 + sim$mde_mi5 + sim$mde_mi6 + sim$mde_mi7 + sim$mde_mi8 +
sim$mde_mi9)

sim$mde = sim$mde_ma1 * sim$mde_ma2 * (sim$mde_mi3 + sim$mde_mi4 +
sim$mde_mi5 + sim$mde_mi6 + sim$mde_mi7 + sim$mde_mi8 + sim$mde_mi9 +
sim$mde_bias1) + (1- sim$mde_ma1 * sim$mde_ma2) * (sim$mde_ma1 *
sim$mde_ma2) * (sim$mde_mi3 + sim$mde_mi4 + sim$mde_mi5 + sim$mde_mi6 +
sim$mde_mi7 + sim$mde_mi8 + sim$mde_mi9 + sim$mde_bias2)
```

```
sim$mde_bias = sim$mde - (sim$mde_ma1 + sim$mde_ma2) - (sim$mde_mi3 +
sim$mde_mi4 + sim$mde_mi5 + sim$mde_mi6 + sim$mde_mi7 + sim$mde_mi8 +
sim$mde_mi9 + sim$mde_bias1) - (sim$mde_bias2)

##Definition Below: even the bias and own input variables could not fully
explain the diagnosis
# sim$mde_bias = sim$mde - (sim$mde_mi3 + sim$mde_mi4 + sim$mde_mi5 +
sim$mde_mi6 + sim$mde_mi7 + sim$mde_mi8 + sim$mde_mi9 + sim$mde_bias1) -
(sim$mde_mi3 + sim$mde_mi4 + sim$mde_mi5 + sim$mde_mi6 + sim$mde_mi7 +
sim$mde_mi8 + sim$mde_mi9 + sim$mde_bias2)

# sim$mde_bias = sim$mde - (sim$mde_ma1 + sim$mde_ma2 + sim$mde_mi3 +
sim$mde_mi4 + sim$mde_mi5 + sim$mde_mi6 + sim$mde_mi7 + sim$mde_mi8 +
sim$mde_mi9)


# sim$mde_bias = resid(lm(sim$mde ~ sim$mde_ma1 + sim$mde_ma2 +
sim$mde_mi3 + sim$mde_mi4 + sim$mde_mi5 + sim$mde_mi6 + sim$mde_mi7 +
sim$mde_mi8 + sim$mde_mi9, data=sim))

##DYS

sim$dys_ma = bindata[,16]

sim$dys_mi1_1 = bindata[,17]
sim$dys_mi1_2 = bindata[,18]
sim$dys_mi1 = 1*((sim$dys_mi1_1 + sim$dys_mi1_2) > 0)
sim$dys_mi1_bias = sim$dys_mi1 - sim$dys_mi1_1 - sim$dys_mi1_2

sim$dys_mi4 = bindata[,19]

sim$dys_mi6 = bindata[,20]

sim$dys_mi = 1*((sim$dys_mi1 + sim$mde_mi4 + sim$mde_mi6 + sim$dys_mi4 +
sim$mde_mi8 + sim$dys_mi6)>1)

sim$dys_mi_bias = sim$dys_mi - (sim$dys_mi1 + sim$mde_mi4 + sim$mde_mi6 +
sim$dys_mi4 + sim$mde_mi8 + sim$dys_mi6)

sim$dys = sim$dys_ma * sim$dys_mi


sim$dys_bias = sim$dys - (sim$dys_ma + sim$dys_mi)

# sim$dys_bias = resid(lm(sim$dys ~ sim$dys_ma + sim$dys_mi, data=sim))


##Manic
sim$man_ma1 = bindata2[,1]
sim$man_ma2 = bindata2[,2]
sim$man_ma3 = bindata2[,3]

sim$man_mi1_1 = bindata2[,4]
sim$man_mi1_2 = bindata2[,5]
sim$man_mi1 = 1*((sim$man_mi1_1 + sim$man_mi1_2) >0)
sim$man_mi1_bias = sim$man_mi1 - (sim$man_mi1_1 + sim$man_mi1_2)
sim$man_mi2 = bindata2[,6]
```

```
sim$man_mi3_1 = bindata2[,7]
sim$man_mi3_2 = bindata2[,8]
sim$man_mi3 = 1*((sim$man_mi3_1 + sim$man_mi3_2) > 0)
sim$man_mi3_bias = sim$man_mi3 - (sim$man_mi3_1 + sim$man_mi3_2)
sim$man_mi4_1 = bindata2[,9]
sim$man_mi4_2 = bindata2[,10]
sim$man_mi4 = 1*((sim$man_mi4_1 + sim$man_mi4_2) > 0)
sim$man_mi4_bias = sim$man_mi4 - (sim$man_mi4_1 + sim$man_mi4_2)
sim$man_mi5 = bindata2[,11]
sim$man_mi6_1 = bindata2[,12]
sim$man_mi6_2 = bindata2[,13]
sim$man_mi6 = 1*((sim$man_mi6_1 + sim$man_mi6_2) > 0)
sim$man_mi6_bias = sim$man_mi6 - (sim$man_mi6_1 + sim$man_mi6_2)
sim$man_mi7 = bindata2[,14]
sim$man_bias1 = 1*((sim$man_mi1 + sim$man_mi2 + sim$man_mi3 + sim$man_mi4
+ sim$man_mi5 + sim$man_mi6 + sim$man_mi7) > 2) - (sim$man_mi1 +
sim$man_mi2 + sim$man_mi3 + sim$man_mi4 + sim$man_mi5 + sim$man_mi6 +
sim$man_mi7)
sim$man_bias2 = 1*((sim$man_mi1 + sim$man_mi2 + sim$man_mi3 + sim$man_mi4
+ sim$man_mi5 + sim$man_mi6 + sim$man_mi7) > 3) - (sim$man_mi1 +
sim$man_mi2 + sim$man_mi3 + sim$man_mi4 + sim$man_mi5 + sim$man_mi6 +
sim$man_mi7)


sim$manic = (1- sim$man_ma1 * sim$man_ma2) * (sim$man_ma1 + sim$man_ma2)
* sim$man_ma3 * (sim$man_mi1 + sim$man_mi2 + sim$man_mi3 + sim$man_mi4 +
sim$man_mi5 + sim$man_mi6 + sim$man_mi7 + sim$man_bias1) + (1 - (1 -
sim$man_ma1 * sim$man_ma2) * (sim$man_ma1 + sim$man_ma2)) * sim$man_ma3 *
(sim$man_mi1 + sim$man_mi2 + sim$man_mi3 + sim$man_mi4 + sim$man_mi5 +
sim$man_mi6 + sim$man_mi7 + sim$man_bias2)


sim$man_bias = sim$manic - (sim$man_ma1 + sim$man_ma2 + sim$man_ma3) -
(sim$man_mi1 + sim$man_mi2 + sim$man_mi3 + sim$man_mi4 + sim$man_mi5 +
sim$man_mi6 + sim$man_mi7 + sim$man_bias1) - (sim$man_bias2)

  ##end of generate data


  resu[, paste(collect, "_", c, sep = "")] = NA
    for(r in 1:nrow(resu)){
      #variable characteristics
      if(is.na(resu$variable[r]) == FALSE){
        resu[r, paste0("derivedprevalence_", c, collapse = "")] =
nrow(sim[which(sim[, resu$variable[r]] == 1),])/ssize
        resu[r, paste0("mean_", c, collapse = "")] =
mean(sim[,resu$variable[r]])
        resu[r, paste0("max_", c, collapse = "")] =
max(sim[,resu$variable[r]])
        resu[r, paste0("min_", c, collapse = "")] =
min(sim[,resu$variable[r]])
      }
      ##regression for the diagnosis
      if(is.na(resu$variable[r]) == FALSE & resu$variable[r] !=
resu$outcome[r]){
        eval(parse(text = paste0("templm = summary(lm(", resu$outcome[r],
" ~ ", resu$variable[r], ", data = sim))", collpase = "")))
```

```
        resu[r, paste0("coef_", c, collapse = "")] =
templm$coefficients[resu$variable[r], "Estimate"]
        resu[r, paste0("coefse_", c, collapse = "")] =
templm$coefficients[resu$variable[r], "Std. Error"]
        resu[r, paste0("p_", c, collapse = "")] =
templm$coefficients[resu$variable[r], "Pr(>|t|)"]
        resu[r, paste0("intercept_", c, collapse = "")] =
templm$coefficients["(Intercept)", "Estimate"]
        resu[r, paste0("interceptp_", c, collapse = "")] =
templm$coefficients["(Intercept)", "Pr(>|t|)"]
        resu[r, paste0("r2_", c, collapse = "")] = templm$r.squared
      }
            ##regression for the suboutcome/domain variables
      if(is.na(resu$variable[r]) == FALSE & is.na(resu$suboutcome[r]) ==
FALSE & resu$variable[r] != resu$outcome[r] & resu$variable[r] !=
resu$suboutcome[r]){
        eval(parse(text = paste0("templm = summary(lm(",
resu$suboutcome[r], " ~ ", resu$variable[r], ", data = sim))", collpase =
"")))
        resu[r, paste0("subcoef_", c, collapse = "")] =
templm$coefficients[resu$variable[r], "Estimate"]
        resu[r, paste0("subcoefse_", c, collapse = "")] =
templm$coefficients[resu$variable[r], "Std. Error"]
        resu[r, paste0("subp_", c, collapse = "")] =
templm$coefficients[resu$variable[r], "Pr(>|t|)"]
        resu[r, paste0("subintercept_", c, collapse = "")] =
templm$coefficients["(Intercept)", "Estimate"]
        resu[r, paste0("subinterceptp_", c, collapse = "")] =
templm$coefficients["(Intercept)", "Pr(>|t|)"]
        resu[r, paste0("subr2_", c, collapse = "")] = templm$r.squared
      }


            if(r %in% as.character(1:100*50)){print(c("r:", r))}
    }#r = rows of the variable list


        ##Approximation by own, bias or all variables


          #ploting area_start: only the last simulation data set used for
plotting
          library(leaps)
          #MDE
          #own variables only
          mdeown = NA
          library(car)
          sim.new = sim[,c("mde",
names(summary((lm(as.formula(paste0("mde ~ ", paste0(names(sim)
[grepl("mde_", names(sim)) == TRUE & grepl("bias", names(sim)) == FALSE],
collapse = " + "), collapse = "")), data = sim)))$aliased)
[summary((lm(as.formula(paste0("mde ~ ", paste0(names(sim)[grepl("mde_",
names(sim)) == TRUE & grepl("bias", names(sim)) == FALSE], collapse = " +
"), collapse = "")), data = sim)))$aliased == FALSE &
names(summary((lm(as.formula(paste0("mde ~ ", paste0(names(sim)
[grepl("mde_", names(sim)) == TRUE & grepl("bias", names(sim)) == FALSE],
collapse = " + "), collapse = "")), data = sim)))$aliased) !=
"(Intercept)"])]
```

```
          for(repe in 1:40){
            tempvif = vif(lm(mde~., data = sim.new))
            if(any(tempvif > 10)){
            sim.new = sim.new[,which(names(sim.new) != names(tempvif)
[which(tempvif == tempvif[order(-tempvif)][1])])]
            }
          }


          try(
            (mdeown = regsubsets(mde~., data = sim.new, really.big=T, 
method = "forward", nvmax = ncol(sim.new))), silent = F
          )

          mdeownsummary = NA
          if(any(is.na(mdeown)) == FALSE){
            mdeownsummary = summary(mdeown)
          }

          mdeownsummary$adjr2



          ##own and bias variables
          mdebias = NA

          mdebias = regsubsets(as.formula(paste0("mde ~ ", 
paste0(names(sim)[grepl("mde_", names(sim)) == TRUE & grepl("bias", 
names(sim)) == TRUE], collapse = " + "), collapse = "")), data = sim, 
nvmax = 100, really.big=T, method = "forward")
          mdebiassummary = summary(mdebias)
          mdebiassummary$adjr2


          ##all variables
          ###in case of collinearity
          mdeall = NA


          ##Deal with collinearity
          library(car)
          sim.new = sim[,c("mde", names(summary((lm(mde~., data = sim)))
$aliased)[summary((lm(mde~., data = sim)))$aliased == FALSE & 
names(summary((lm(mde~., data = sim)))$aliased) != "(Intercept)"])]


          for(repe in 1:40){
            tempvif = vif(lm(mde~., data = sim.new))
            if(any(tempvif > 10)){
            sim.new = sim.new[,which(names(sim.new) != names(tempvif)
[which(tempvif == tempvif[order(-tempvif)][1])])]
            }
          }

          ##Somehow there are problems in executing regsubsets even after 
removing collinear variables
```

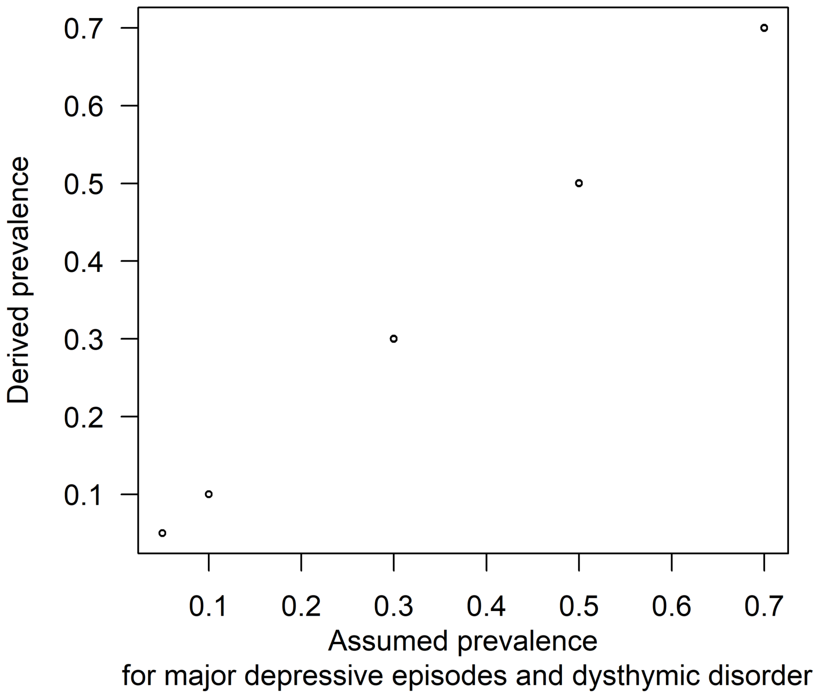
```
          try(
            (mdeall = regsubsets(mde~., data = sim.new, really.big=T, 
method = "forward", nvmax = ncol(sim.new))), silent = F
          )
          
          mdeallsummary = NA
          if(any(is.na(mdeall)) == FALSE){
            mdeallsummary = summary(mdeall)
          }
          
          # mdeallsummary$adjr2
          
          
          #DYS
          #dys
          #own variables only
          dysown = NA
                library(car)
          sim.new = sim[,c("dys", 
names(summary((lm(as.formula(paste0("dys ~ ", paste0(c(names(sim)
[(grepl("dys_", names(sim)) == TRUE | grepl("mde_mi4", names(sim)) == 
TRUE | grepl("mde_mi6", names(sim)) == TRUE | grepl("mde_mi8", 
names(sim)) == TRUE) & grepl("bias", names(sim)) == FALSE]), collapse = " 
+ "), collapse = "")), data = sim)))$aliased)
[summary((lm(as.formula(paste0("dys ~ ", paste0(c(names(sim)
[(grepl("dys_", names(sim)) == TRUE | grepl("mde_mi4", names(sim)) == 
TRUE | grepl("mde_mi6", names(sim)) == TRUE | grepl("mde_mi8", 
names(sim)) == TRUE) & grepl("bias", names(sim)) == FALSE]), collapse = " 
+ "), collapse = "")), data = sim)))$aliased == FALSE & 
names(summary((lm(as.formula(paste0("dys ~ ", paste0(c(names(sim)
[(grepl("dys_", names(sim)) == TRUE | grepl("mde_mi4", names(sim)) == 
TRUE | grepl("mde_mi6", names(sim)) == TRUE | grepl("mde_mi8", 
names(sim)) == TRUE) & grepl("bias", names(sim)) == FALSE]), collapse = " 
+ "), collapse = "")), data = sim)))$aliased) != "(Intercept)"])]
          
          for(repe in 1:40){
                      tempvif = vif(lm(dys~., data = sim.new))
              if(any(tempvif > 10)){
              sim.new = sim.new[,which(names(sim.new) != names(tempvif)
[which(tempvif == tempvif[order(-tempvif)][1])])]
              }
          }
          ##Somehow there are problems in executing regsubsets even after 
removing collinear variables
          try(
            (dysown = regsubsets(dys~., data = sim.new, really.big=T, 
method = "forward", nvmax = 100)), silent = T
          )
          
          if(any(is.na(dysown)) == FALSE){
            dysownsummary = summary(dysown)
          }
          
          ##own and bias variables
          dysbias = NA
          dysbias = regsubsets(as.formula(paste0("dys ~ ",
paste0(names(sim)[(grepl("dys_", names(sim)) == TRUE | grepl("mde_mi4",
names(sim)) == TRUE | grepl("mde_mi6", names(sim)) == TRUE |
grepl("mde_mi8", names(sim)) == TRUE) & grepl("bias", names(sim)) ==
TRUE], collapse = " + "), collapse = "")), data = sim, nvmax = 100,
really.big=T, method = "forward")
          dysbiassummary = summary(dysbias)
          dysbiassummary$adjr2


          ##all variables
          ###in case of collinearity
          dysall = NA
          library(car)
          sim.new = sim[,c("dys", names(summary((lm(dys~., data = sim)))
$aliased)[summary((lm(dys~., data = sim)))$aliased == FALSE &
names(summary((lm(dys~., data = sim)))$aliased) != "(Intercept)"])]

          for(repe in 1:40){
                      tempvif = vif(lm(dys~., data = sim.new))
              if(any(tempvif > 10)){
              sim.new = sim.new[,which(names(sim.new) != names(tempvif)
[which(tempvif == tempvif[order(-tempvif)][1])])]
              }
          }
          ##Somehow there are problems in executing regsubsets even after
removing collinear variables
          try(
            (dysall = regsubsets(dys~., data = sim.new, really.big=T,
method = "forward", nvmax = 100)), silent = T
          )

          if(any(is.na(dysall)) == FALSE){
            dysallsummary = summary(dysall)
          }
          # dysallsummary$adjr2




          #manic
          #own variables only
          manown = NA
                    library(car)
          sim.new = sim[,c("manic",
names(summary((lm(as.formula(paste0("manic ~ ", paste0(names(sim)
[grepl("man_", names(sim)) == TRUE & grepl("bias", names(sim)) == FALSE],
collapse = " + "), collapse = "")), data = sim)))$aliased)
[summary((lm(as.formula(paste0("manic ~ ", paste0(names(sim)
[grepl("man_", names(sim)) == TRUE & grepl("bias", names(sim)) == FALSE],
collapse = " + "), collapse = "")), data = sim)))$aliased == FALSE &
names(summary((lm(as.formula(paste0("manic ~ ", paste0(names(sim)
[grepl("man_", names(sim)) == TRUE & grepl("bias", names(sim)) == FALSE],
collapse = " + "), collapse = "")), data = sim)))$aliased) !=
"(Intercept)"])]
          for(repe in 1:40){
                      tempvif = vif(lm(manic~., data = sim.new))
              if(any(tempvif > 10)){
```

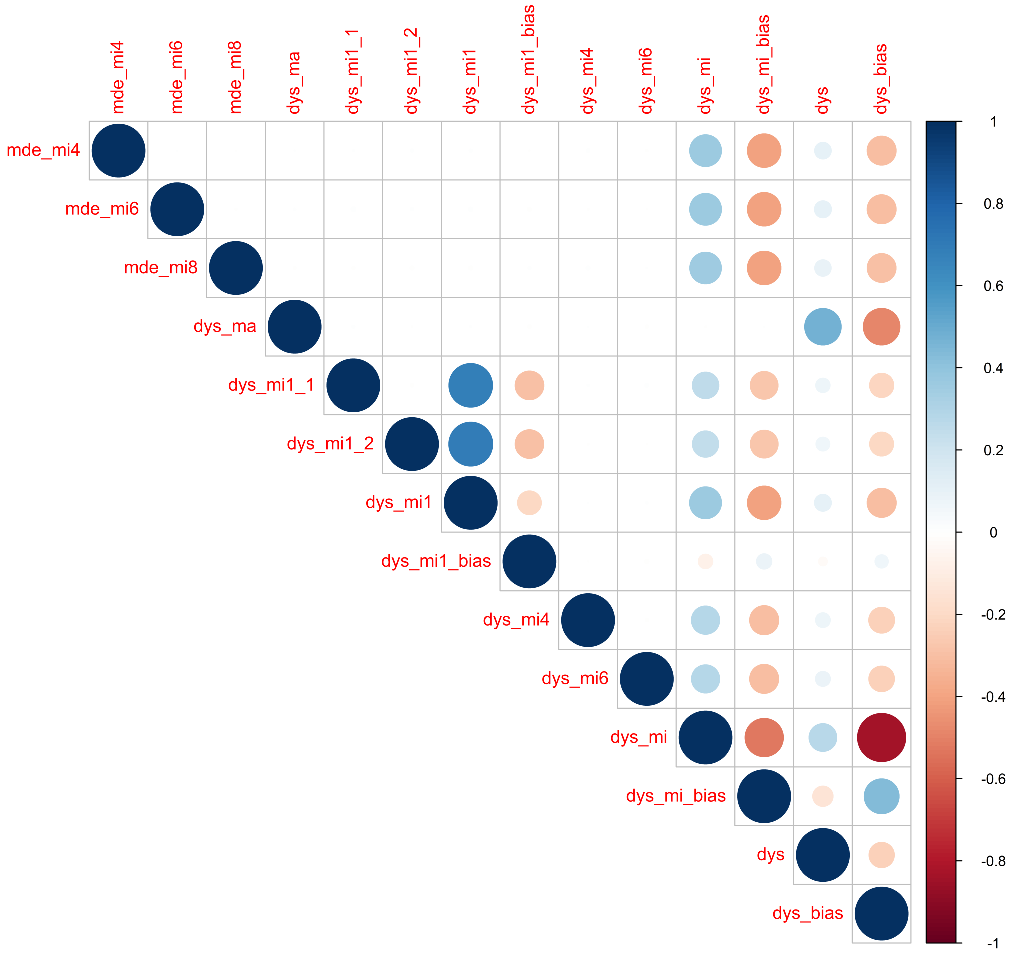
```
              sim.new = sim.new[,which(names(sim.new) != names(tempvif)
[which(tempvif == tempvif[order(-tempvif)][1])])]
              }
          }
          try(
            (manown = regsubsets(manic~., data = sim.new, really.big=T,
method = "forward", nvmax = 100)), silent = T
          )
          manownsummary = NA
          if(any(is.na(manown)) == FALSE){
            manownsummary = summary(manown)
          }

          ##own and bias variables
          manbias = NA

          manbias = regsubsets(as.formula(paste0("manic ~ ",
paste0(names(sim)[grepl("man_", names(sim)) == TRUE & grepl("bias",
names(sim)) == TRUE], collapse = " + "), collapse = "")), data = sim,
nvmax = 100, really.big=T, method = "forward")
          manbiassummary = summary(manbias)
          manbiassummary$adjr2


          ##all variables
          ###in case of collinearity
          manall = NA
                    library(car)
          sim.new = sim[,c("manic", names(summary((lm(manic~., data =
sim)))$aliased)[summary((lm(manic~., data = sim)))$aliased == FALSE &
names(summary((lm(manic~., data = sim)))$aliased) != "(Intercept)"])]
          for(repe in 1:40){
                      tempvif = vif(lm(manic~., data = sim.new))
              if(any(tempvif > 10)){
              sim.new = sim.new[,which(names(sim.new) != names(tempvif)
[which(tempvif == tempvif[order(-tempvif)][1])])]
              }
          }
          ##Somehow there are problems in executing regsubsets even after
removing collinear variables
          try(
            (manall = regsubsets(manic~., data = sim.new, really.big=T,
method = "forward", nvmax = 100)), silent = T
          )
          manallsummary = NA
          if(any(is.na(manall)) == FALSE){
            manallsummary = summary(manall)
          }

  ##extract information from the outmat
       #MDE

          resu[which(resu$variable == "mde"), paste0("appbyownr2_", c,
collapse = "")] = mdeownsummary$adjr2[which.max(mdeownsummary$adjr2)]
          resu[which(resu$variable == "mde"), paste0("appbyownn_", c,
collapse = "")] = which.max(mdeownsummary$adjr2)
```

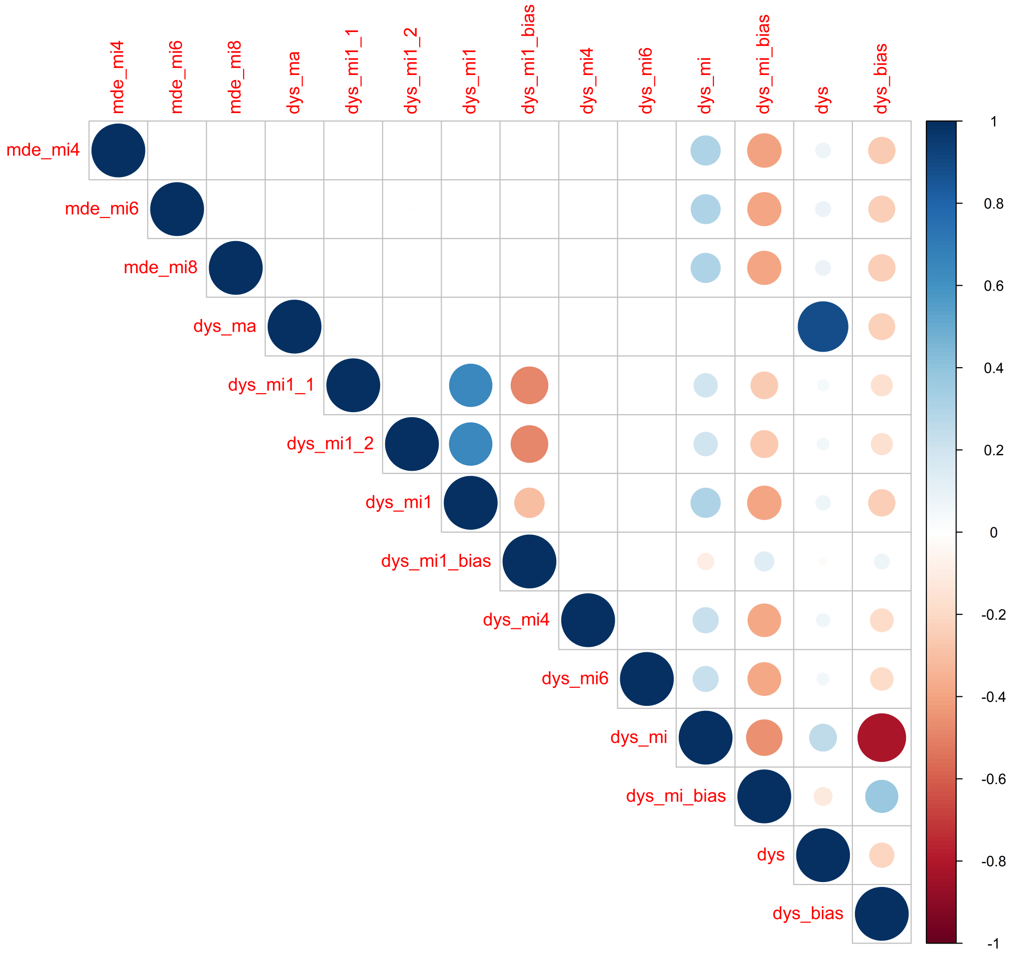
```
          resu[which(resu$variable == "mde"), paste0("appbyownvar_", c,
collapse = "")] = paste0(dimnames(mdeownsummary$outmat)[[2]]
[which(mdeownsummary$outmat[which.max(mdeownsummary$adjr2),] == "*")],
collapse = ",")

          resu[which(resu$variable == "mde"), paste0("appbybiasr2_", c,
collapse = "")] = mdebiassummary$adjr2[which.max(mdebiassummary$adjr2)]
          resu[which(resu$variable == "mde"), paste0("appbybiasn_", c,
collapse = "")] = which.max(mdebiassummary$adjr2)
          resu[which(resu$variable == "mde"), paste0("appbybiasvar_", c,
collapse = "")] = paste0(dimnames(mdebiassummary$outmat)[[2]]
[which(mdebiassummary$outmat[which.max(mdebiassummary$adjr2),] == "*")],
collapse = ",")

          if(any(is.na(mdeall)) == FALSE){
              resu[which(resu$variable == "mde"), paste0("appbyallr2_",
c, collapse = "")] = mdeallsummary$adjr2[which.max(mdeallsummary$adjr2)]
              resu[which(resu$variable == "mde"), paste0("appbyalln_", c,
collapse = "")] = which.max(mdeallsummary$adjr2)
              resu[which(resu$variable == "mde"), paste0("appbyallvar_",
c, collapse = "")] = paste0(dimnames(mdeallsummary$outmat)[[2]]
[which(mdeallsummary$outmat[which.max(mdeallsummary$adjr2),] == "*")],
collapse = ",")

          }


       #DYS
          resu[which(resu$variable == "dys"), paste0("appbyownr2_", c,
collapse = "")] = dysownsummary$adjr2[which.max(dysownsummary$adjr2)]
          resu[which(resu$variable == "dys"), paste0("appbyownn_", c,
collapse = "")] = which.max(dysownsummary$adjr2)
          resu[which(resu$variable == "dys"), paste0("appbyownvar_", c,
collapse = "")] = paste0(dimnames(dysownsummary$outmat)[[2]]
[which(dysownsummary$outmat[which.max(dysownsummary$adjr2),] == "*")],
collapse = ",")

          resu[which(resu$variable == "dys"), paste0("appbybiasr2_", c,
collapse = "")] = dysbiassummary$adjr2[which.max(dysbiassummary$adjr2)]
          resu[which(resu$variable == "dys"), paste0("appbybiasn_", c,
collapse = "")] = which.max(dysbiassummary$adjr2)
          resu[which(resu$variable == "dys"), paste0("appbybiasvar_", c,
collapse = "")] = paste0(dimnames(dysbiassummary$outmat)[[2]]
[which(dysbiassummary$outmat[which.max(dysbiassummary$adjr2),] == "*")],
collapse = ",")

          if(any(is.na(dysall)) == FALSE){
            resu[which(resu$variable == "dys"), paste0("appbyallr2_", c,
collapse = "")] = dysallsummary$adjr2[which.max(dysallsummary$adjr2)]
          resu[which(resu$variable == "dys"), paste0("appbyalln_", c,
collapse = "")] = which.max(dysallsummary$adjr2)
          resu[which(resu$variable == "dys"), paste0("appbyallvar_", c,
collapse = "")] = paste0(dimnames(dysallsummary$outmat)[[2]]
[which(dysallsummary$outmat[which.max(dysallsummary$adjr2),] == "*")],
collapse = ",")
          }
       #MANIC
          resu[which(resu$variable == "manic"), paste0("appbyownr2_", c, 
collapse = "")] = manownsummary$adjr2[which.max(manownsummary$adjr2)]
          resu[which(resu$variable == "manic"), paste0("appbyownn_", c, 
collapse = "")] = which.max(manownsummary$adjr2)
          resu[which(resu$variable == "manic"), paste0("appbyownvar_", c, 
collapse = "")] = paste0(dimnames(manownsummary$outmat)[[2]]
[which(manownsummary$outmat[which.max(manownsummary$adjr2),] == "*")], 
collapse = ",")

          resu[which(resu$variable == "manic"), paste0("appbybiasr2_", c, 
collapse = "")] = manbiassummary$adjr2[which.max(manbiassummary$adjr2)]
          resu[which(resu$variable == "manic"), paste0("appbybiasn_", c, 
collapse = "")] = which.max(manbiassummary$adjr2)
          resu[which(resu$variable == "manic"), paste0("appbybiasvar_", 
c, collapse = "")] = paste0(dimnames(manbiassummary$outmat)[[2]]
[which(manbiassummary$outmat[which.max(manbiassummary$adjr2),] == "*")], 
collapse = ",")


          if(any(is.na(manall)) == FALSE){
            resu[which(resu$variable == "manic"), paste0("appbyallr2_", 
c, collapse = "")] = manallsummary$adjr2[which.max(manallsummary$adjr2)]
            resu[which(resu$variable == "manic"), paste0("appbyalln_", c, 
collapse = "")] = which.max(manallsummary$adjr2)
            resu[which(resu$variable == "manic"), paste0("appbyallvar_", 
c, collapse = "")] = paste0(dimnames(manallsummary$outmat)[[2]]
[which(manallsummary$outmat[which.max(manallsummary$adjr2),] == "*")], 
collapse = ",")
          }




            print(c("c:", c))
            print(c("cor:", rho[rh]))
            print(c("Prevalence: ", prevalence[preval]))

    }#c



    ##adding summary statistics to the result data frame
    resu[, paste(collect, "_mean", sep = "")] = NA
resu[, paste(collect, "_sd", sep = "")] = NA
resu[, paste(collect, "_se", sep = "")] = NA
resu[, paste(collect, "_95up", sep = "")] = NA
resu[, paste(collect, "_95lo", sep = "")] = NA
resu[, paste(collect, "_rangeup", sep = "")] = NA
resu[, paste(collect, "_rangelo", sep = "")] = NA

for(co in 1:length(collect)){
  for(r in 1:nrow(resu)){
    if((collect[co] %in% c("appbyownvar", "appbybiasvar", "appbyallvar")) 
== FALSE){
          resu[r,paste0(collect[co], "_mean", collapse = "")] = 
mean(unlist(resu[r, paste(collect[co], "_", 1:times, sep = "")])[which(!
is.na(unlist(resu[r, paste(collect[co], "_", 1:times, sep = "")])))])
```

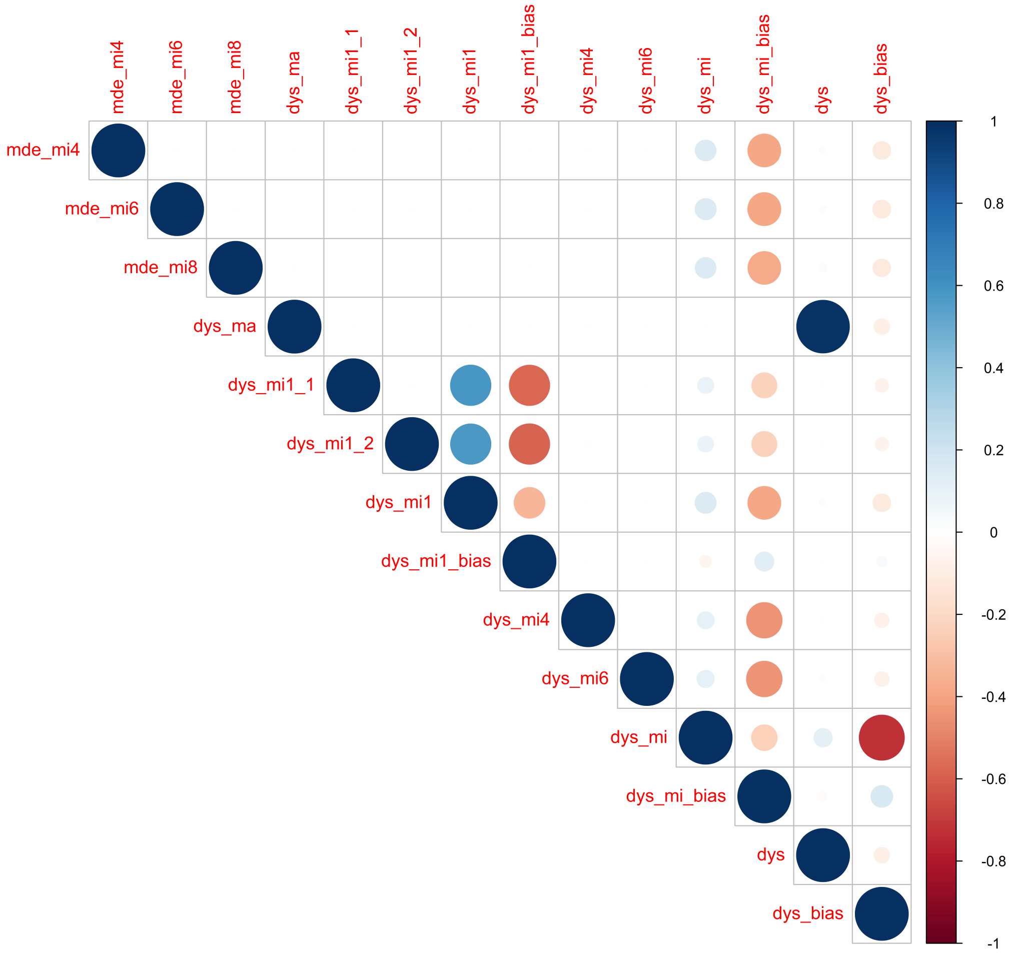
```
    resu[r,paste0(collect[co], "_sd", collapse = "")] = sd(unlist(resu[r,
paste(collect[co], "_", 1:times, sep = "")])[which(!is.na(unlist(resu[r,
paste(collect[co], "_", 1:times, sep = "")])))])
    resu[r,paste0(collect[co], "_se", collapse = "")] = sd(unlist(resu[r,
paste(collect[co], "_", 1:times, sep = "")])[which(!is.na(unlist(resu[r,
paste(collect[co], "_", 1:times, sep = "")])))])/(times^0.5)

    #95% CIs
    resu[r,paste0(collect[co], "_95up", collapse = "")] =
resu[r,paste0(collect[co], "_mean", collapse = "")] +
1.96*resu[r,paste0(collect[co], "_se", collapse = "")]
    resu[r,paste0(collect[co], "_95lo", collapse = "")] =
resu[r,paste0(collect[co], "_mean", collapse = "")] -
1.96*resu[r,paste0(collect[co], "_se", collapse = "")]

    #range
        resu[r,paste0(collect[co], "_rangelo", collapse = "")] =
min(resu[r,paste(collect[co], "_", 1:times, sep = "")])
        resu[r,paste0(collect[co], "_rangeup", collapse = "")] =
max(resu[r,paste(collect[co], "_", 1:times, sep = "")])
  }#r

    ##Add information about the aliased variables



    ##save in another data set
    eval(parse(text = paste0("resu_cor", rho[rh], "_preval",
prevalence[preval], " = resu", collapse = "")))


    }

#export results
write.csv(cbind(resu[,c("definition", "variable", "mean_mean",
"mean_95up","mean_95lo", "max_rangeup",
"min_rangelo","derivedprevalence_mean", "derivedprevalence_95up",
"derivedprevalence_95lo", "coef_mean", "coef_95up", "coef_95lo",
"p_mean","p_95up","p_95lo", "r2_mean", "r2_95up",
"r2_95lo","subcoef_mean", "subcoef_95up", "subcoef_95lo",
"subp_mean","subp_95up","subp_95lo", "subr2_mean", "subr2_95up",
"subr2_95lo", "appbyownr2_mean", "appbyownr2_95up", "appbyownr2_95lo",
"appbyownn_mean", "appbyownn_95up", "appbyownn_95lo", "appbybiasr2_mean",
"appbybiasr2_95up", "appbybiasr2_95lo", "appbybiasn_mean",
"appbybiasn_95up", "appbybiasn_95lo", "appbyallr2_mean",
"appbyallr2_95up", "appbyallr2_95lo", "appbyalln_mean", "appbyalln_95up",
"appbyalln_95lo"

)], resu), file = paste0("simulation results_cor", rho[rh], "_preval",
prevalence[preval], ".csv"))



  }#co
```

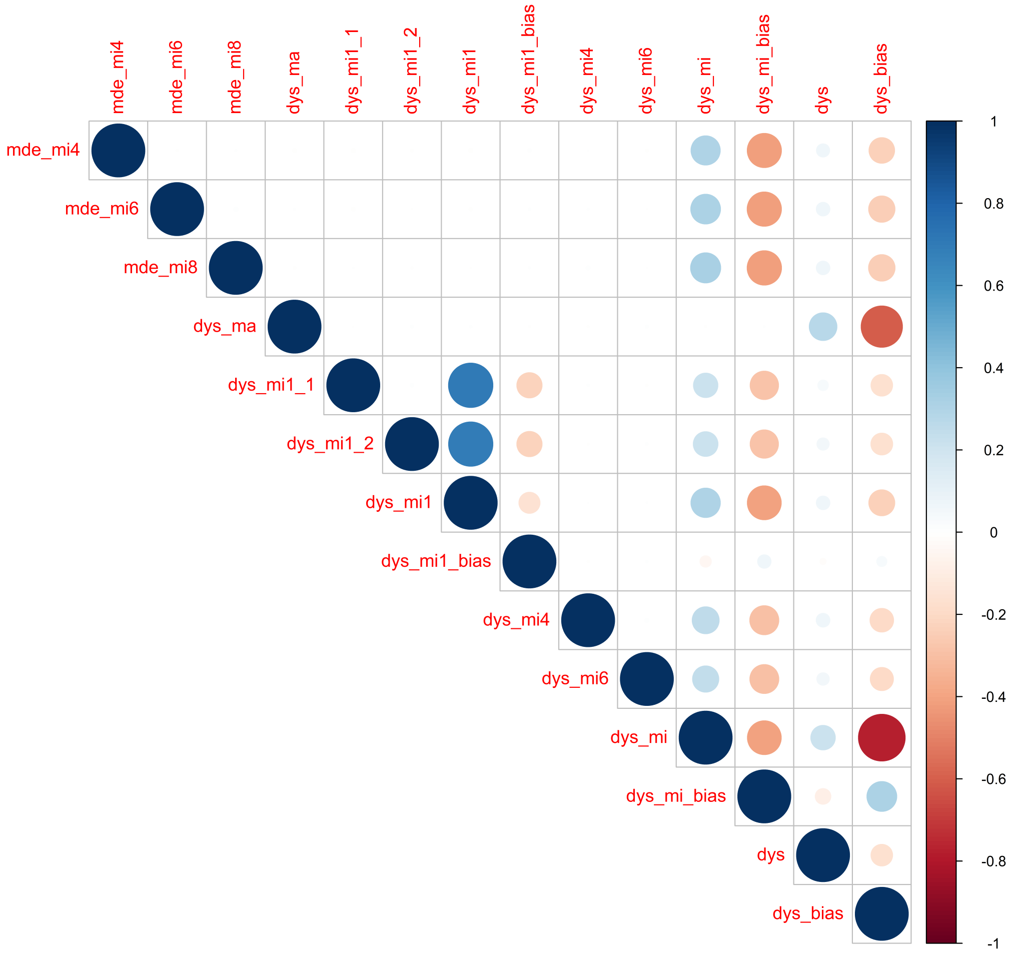
~~~
print(c("cor:", rho[rh]))
print(c("Prevalence: ", prevalence[preval]))

  }#rho
#store data

print(c("Prevalence: ", prevalence[preval]))
  }#prevalence


```
~~~

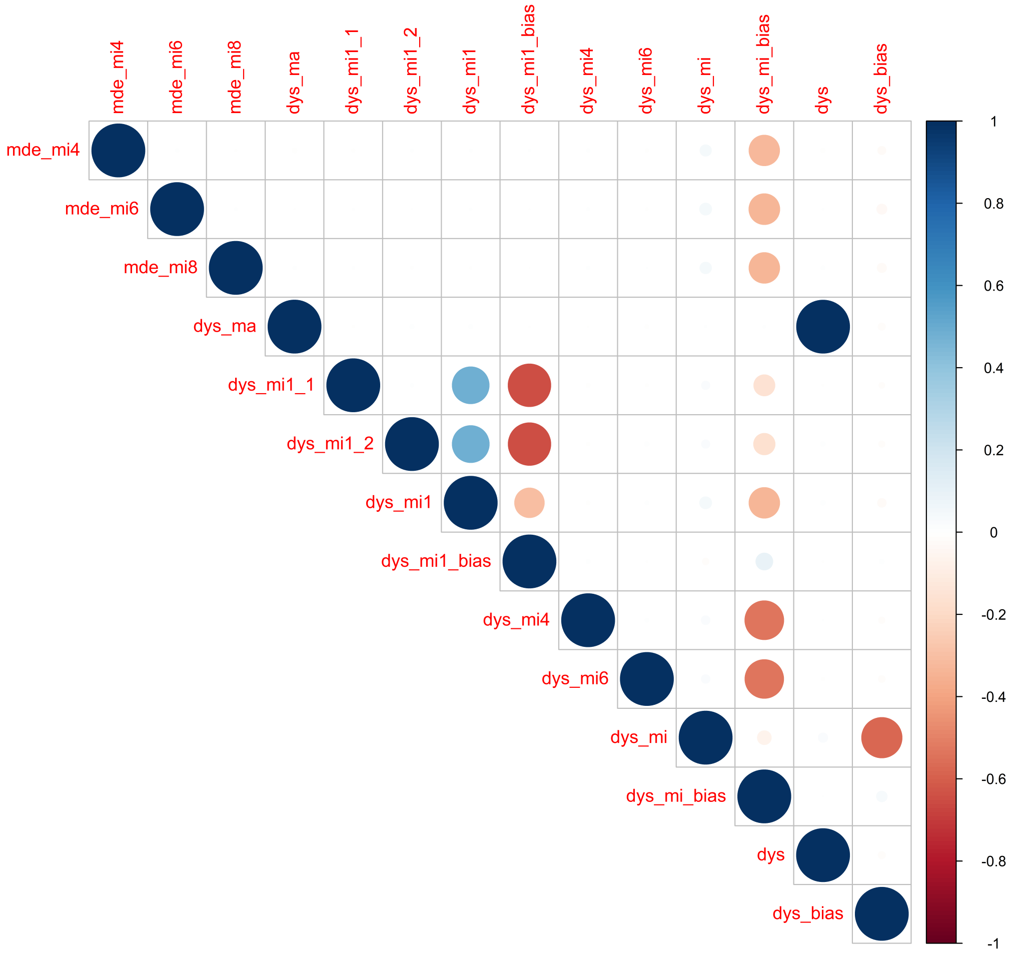

0.7
0.6
0.5
0.4
0.3
0.2
0.1
Derived prevalence
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Assumed prevalence
for major depressive episodes and dysthymic disorder

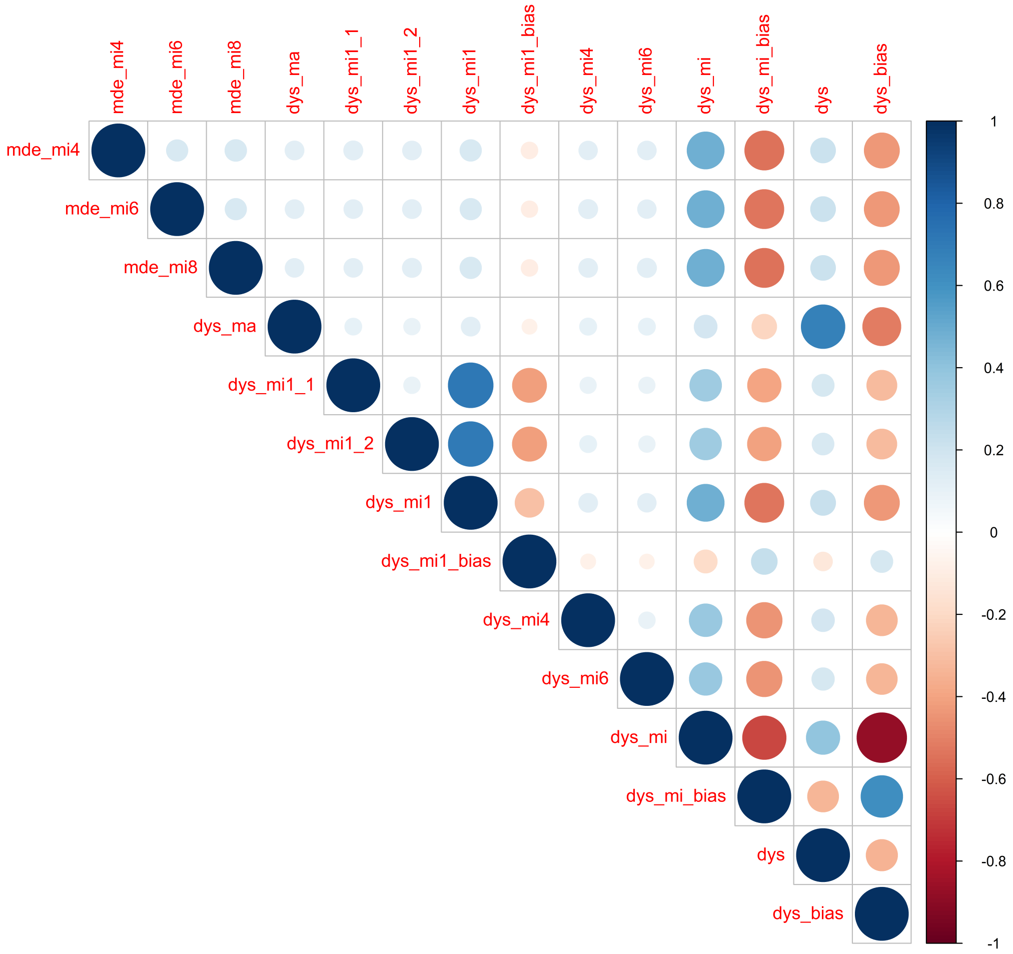

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

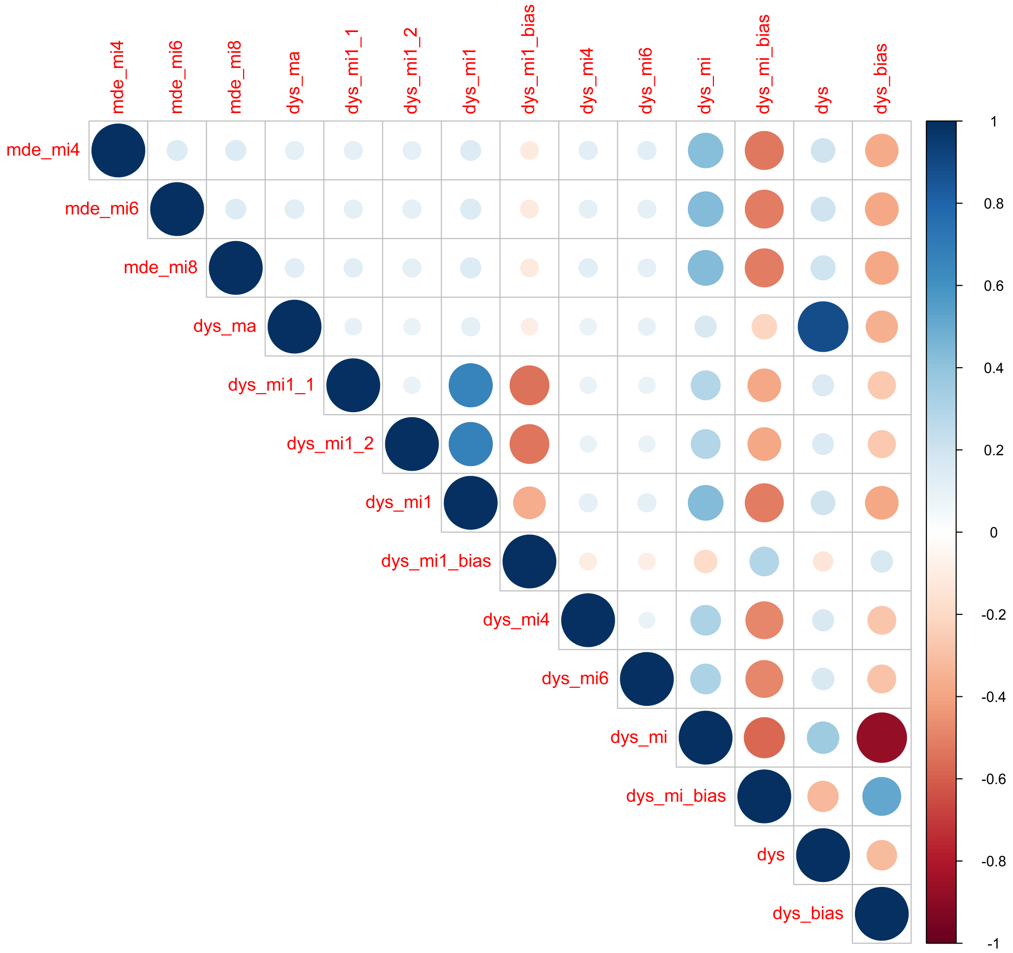

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

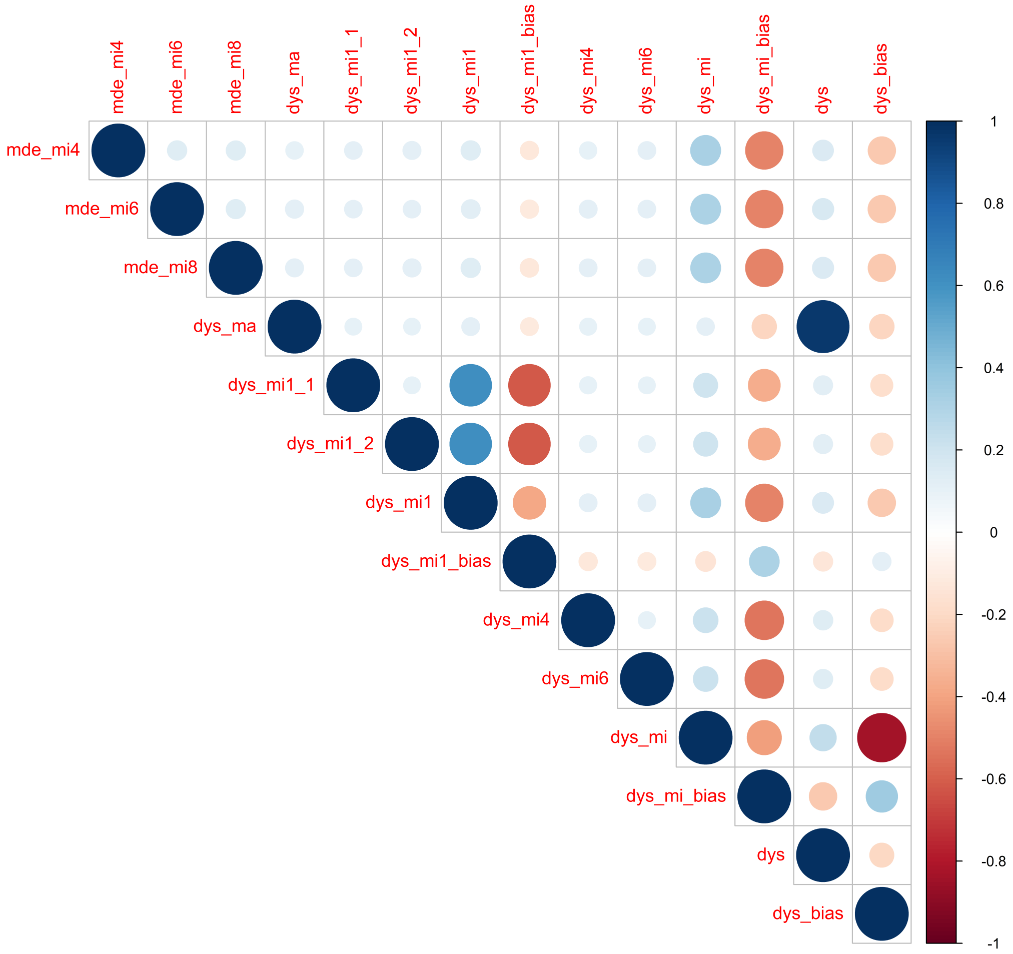

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

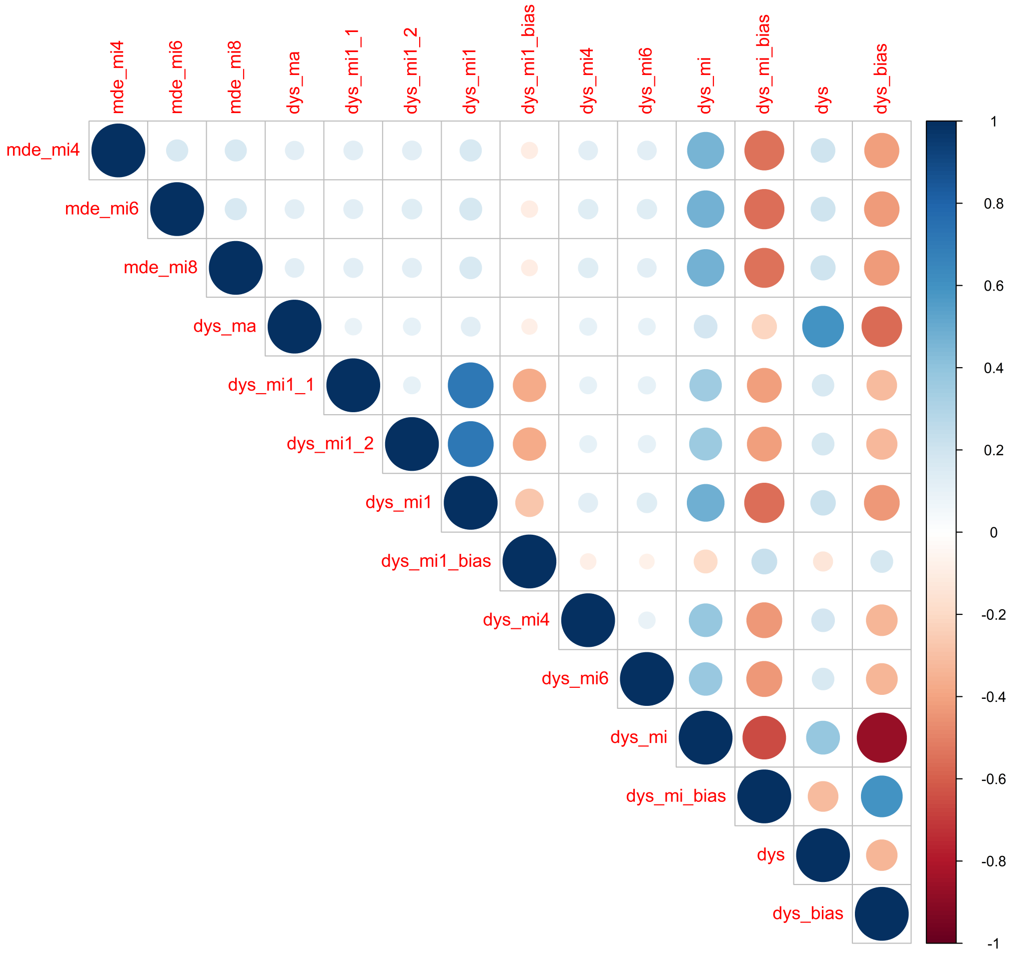

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

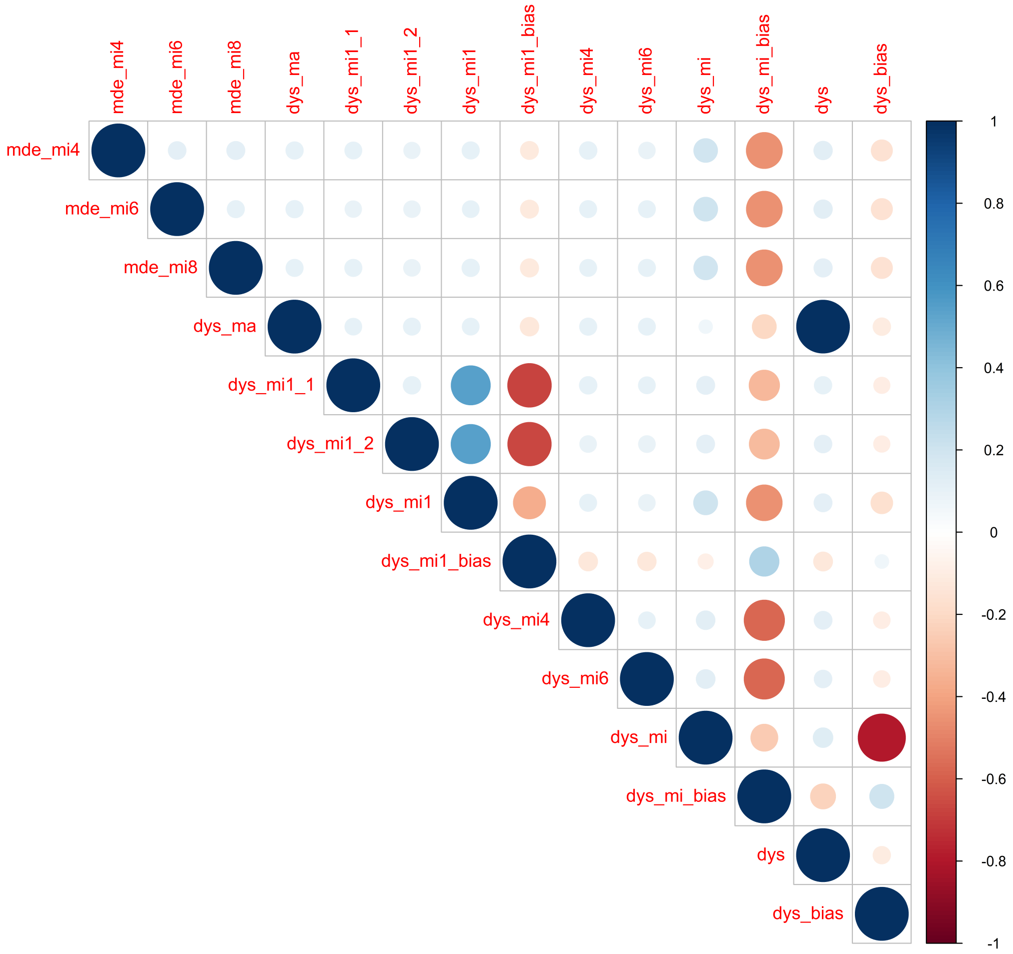

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

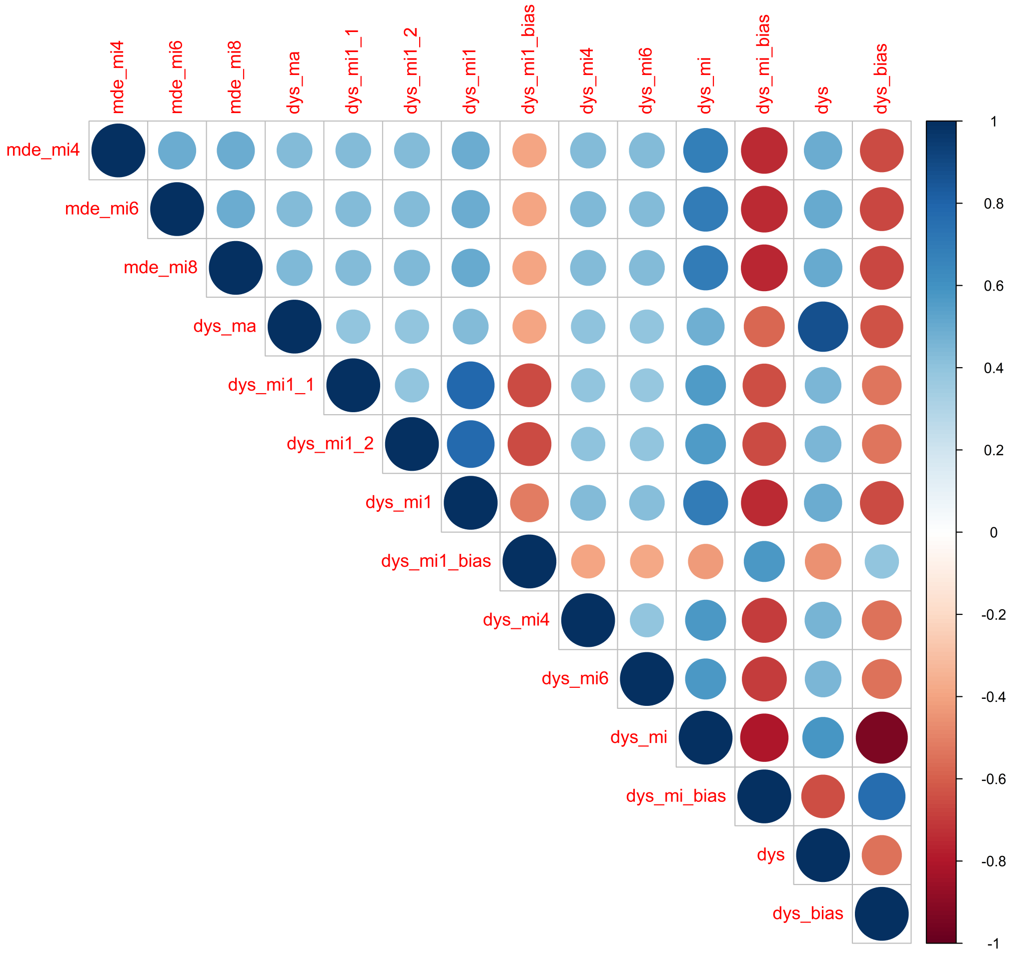

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

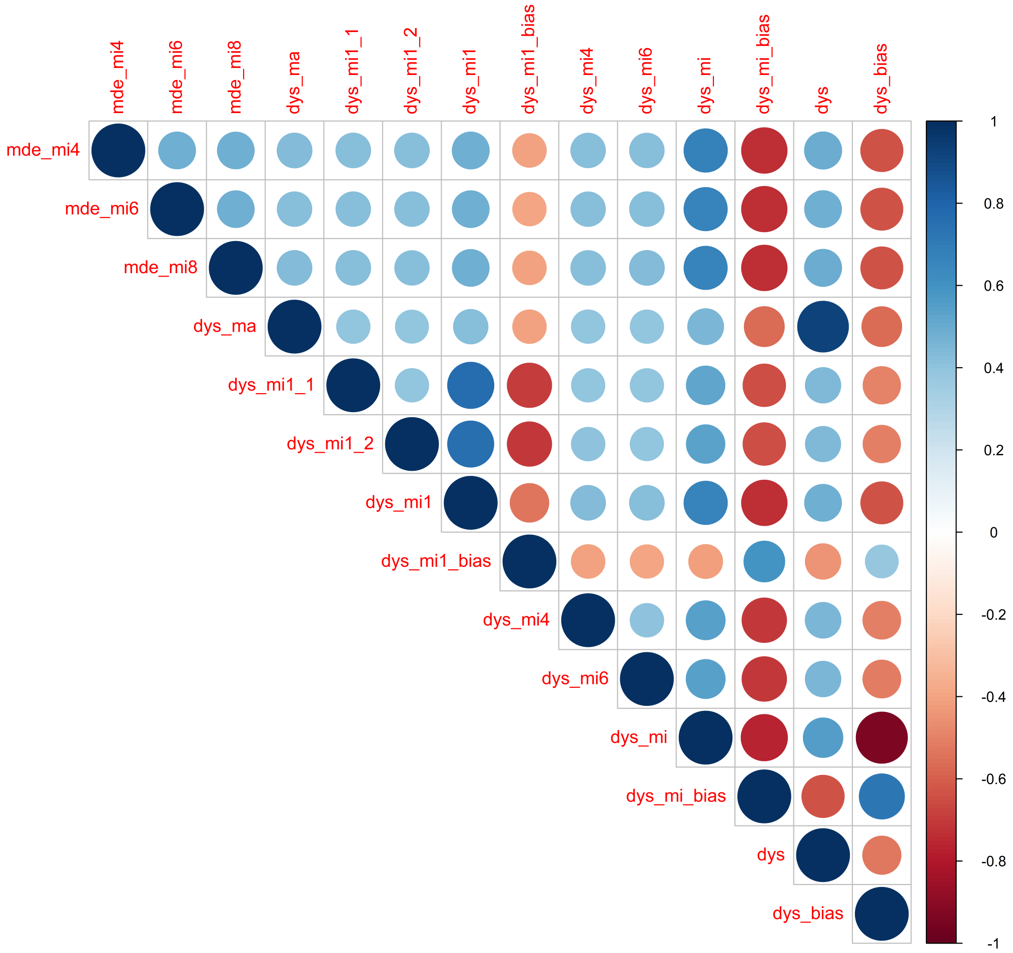

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

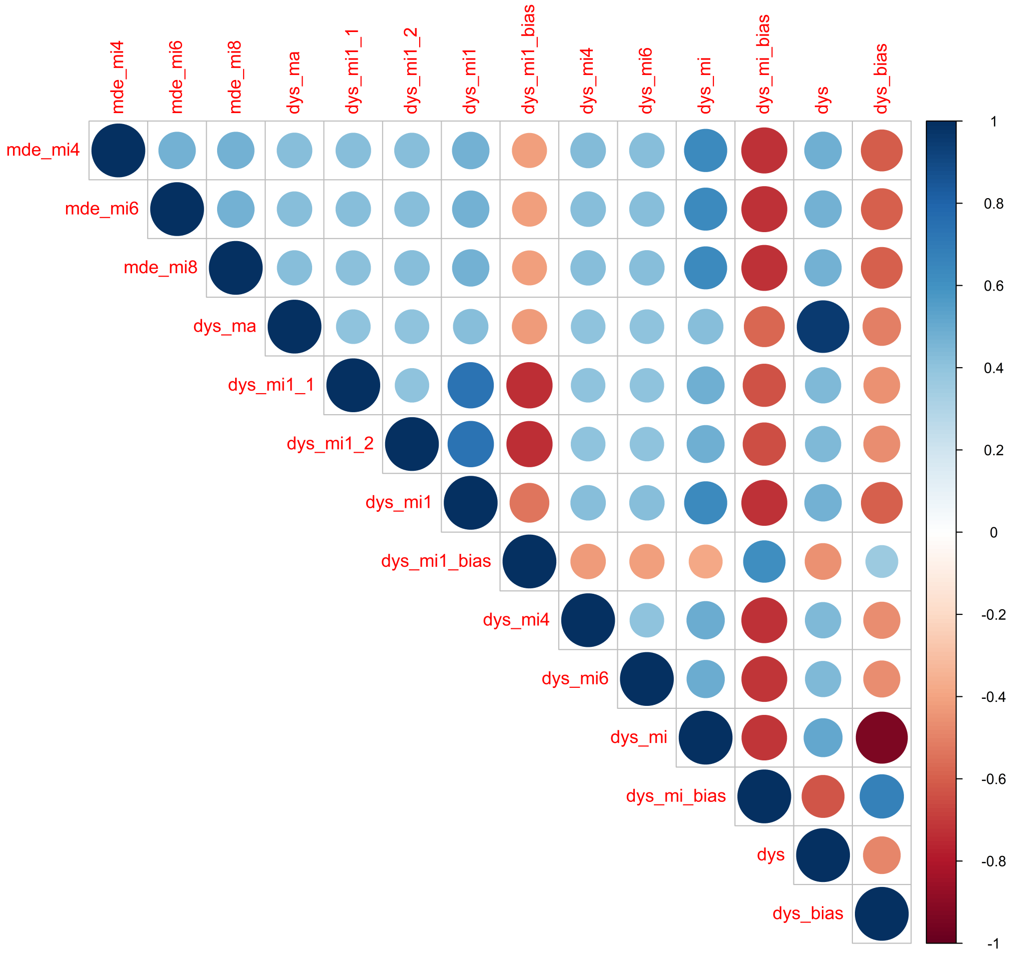

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

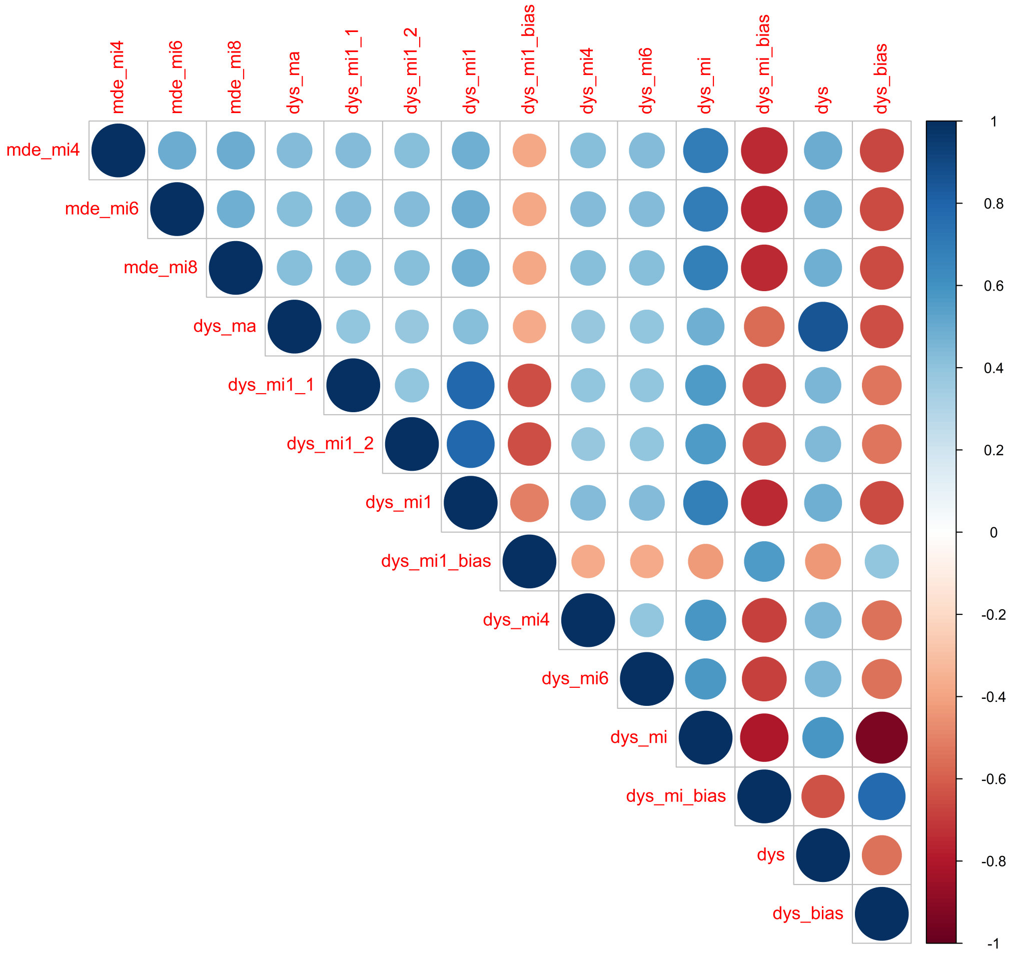

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

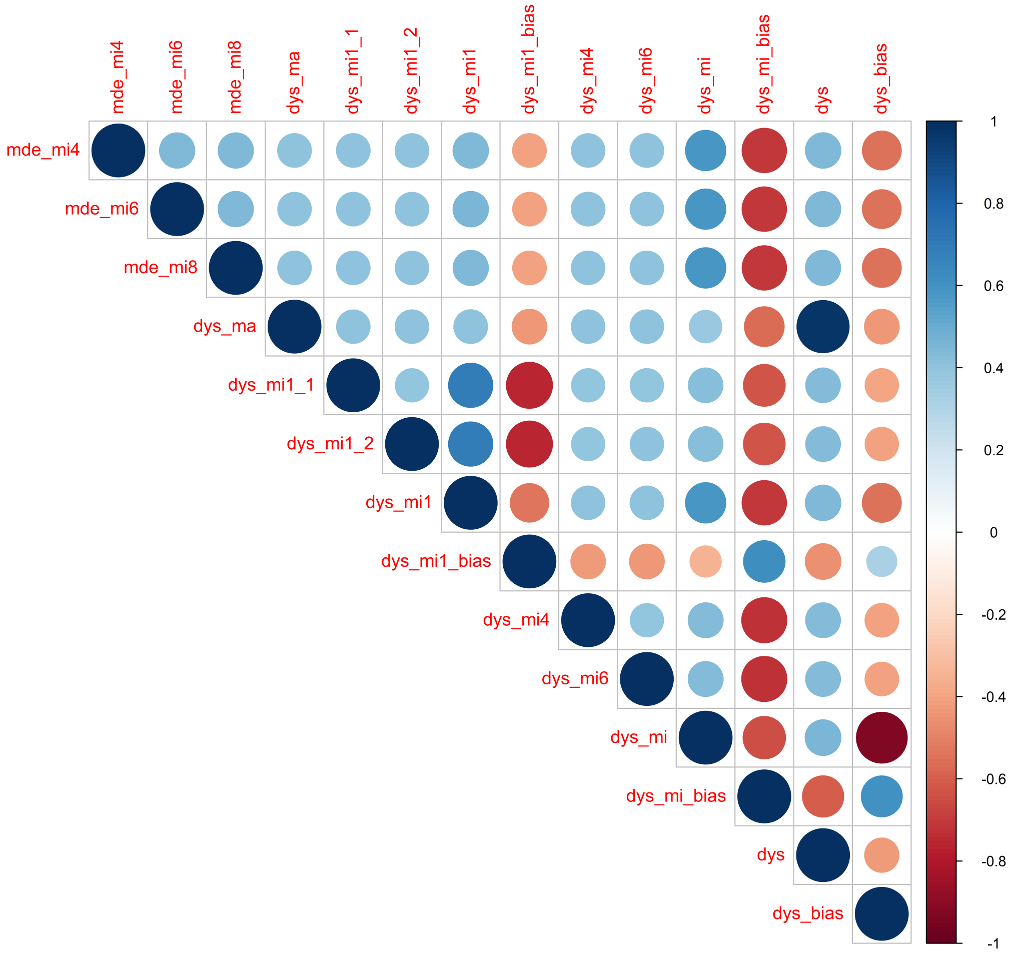

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

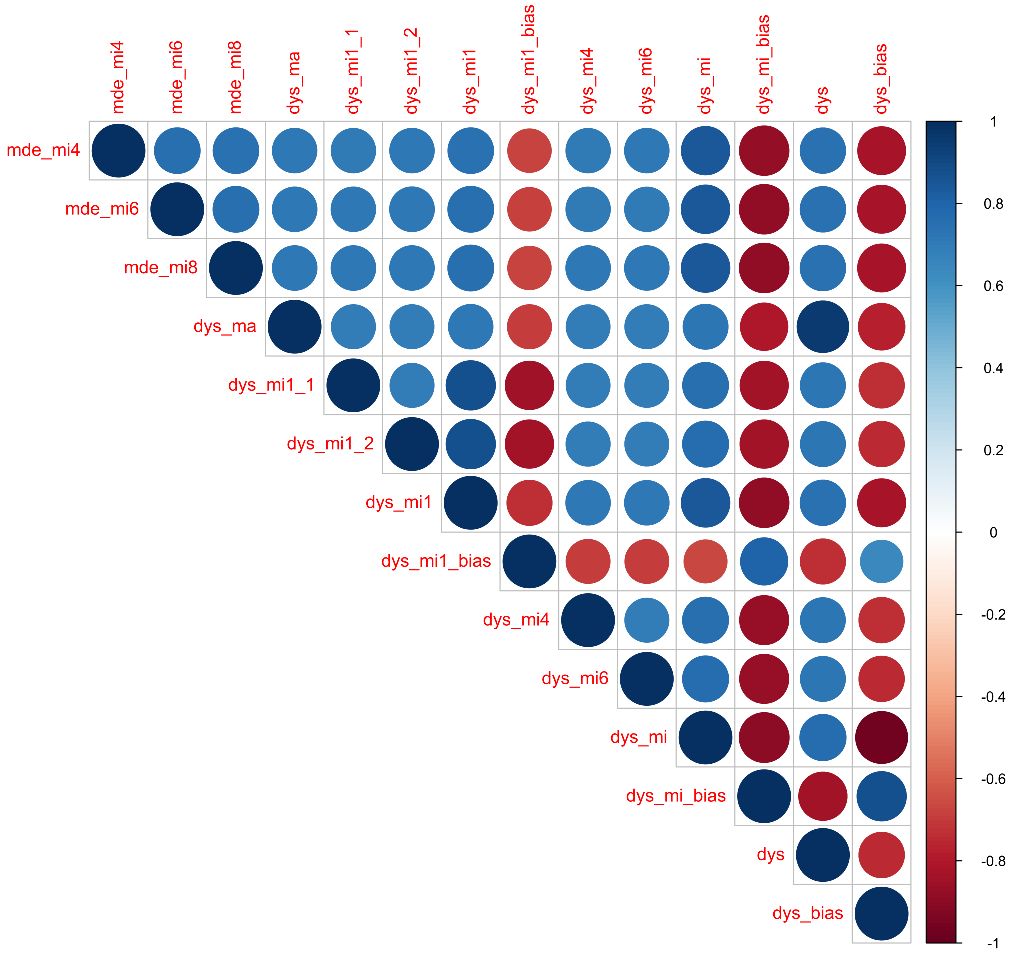

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

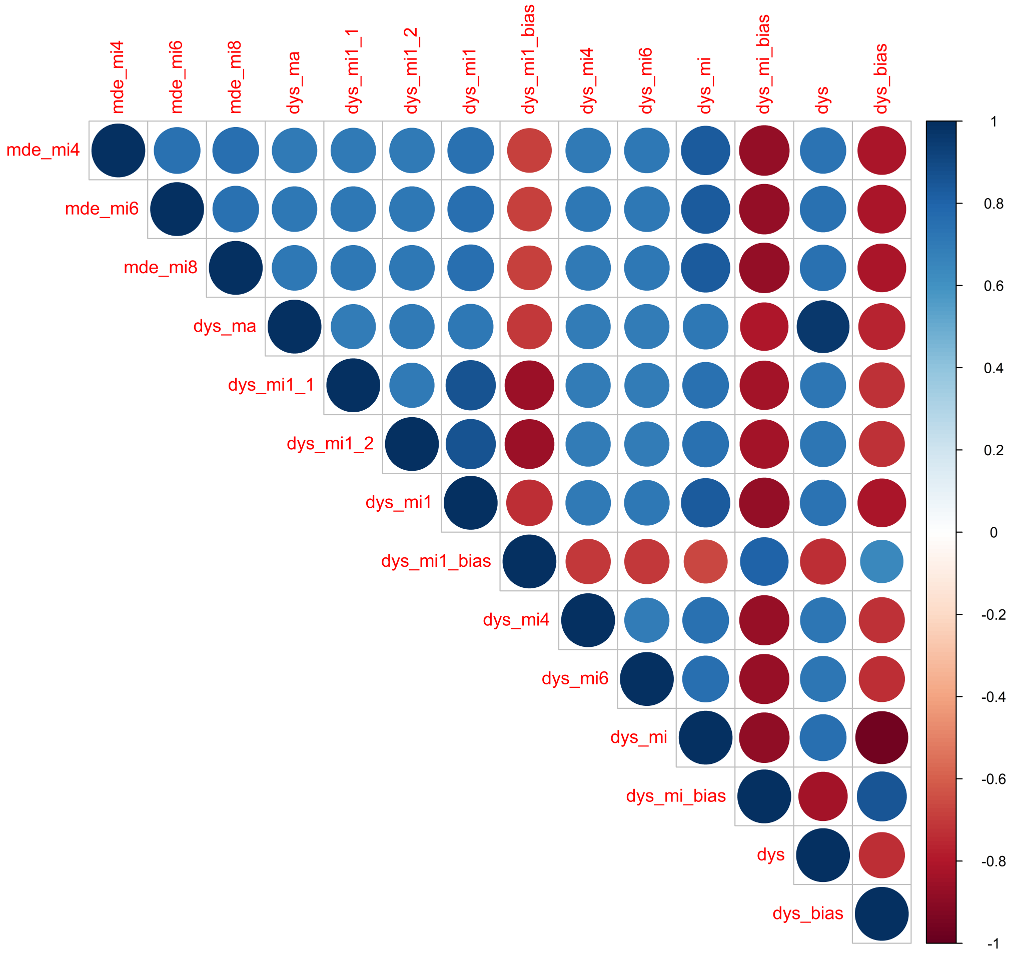

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

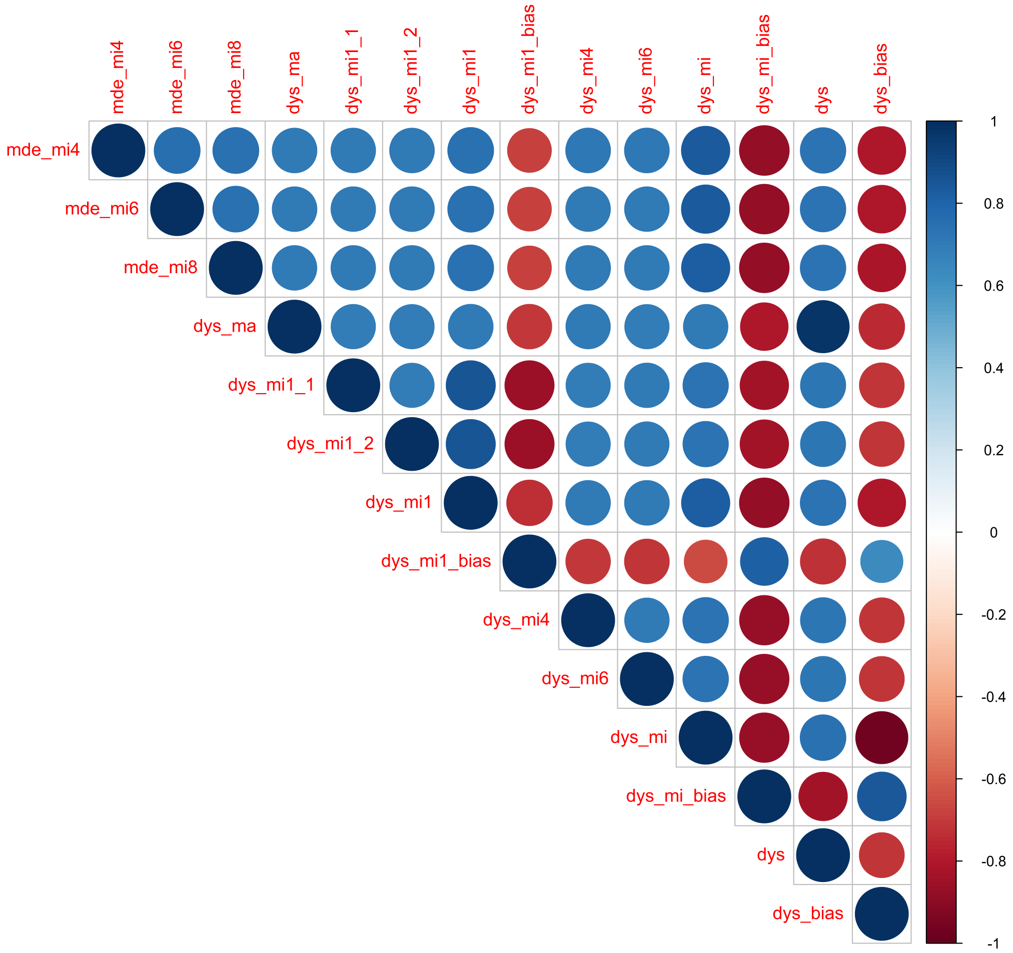

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

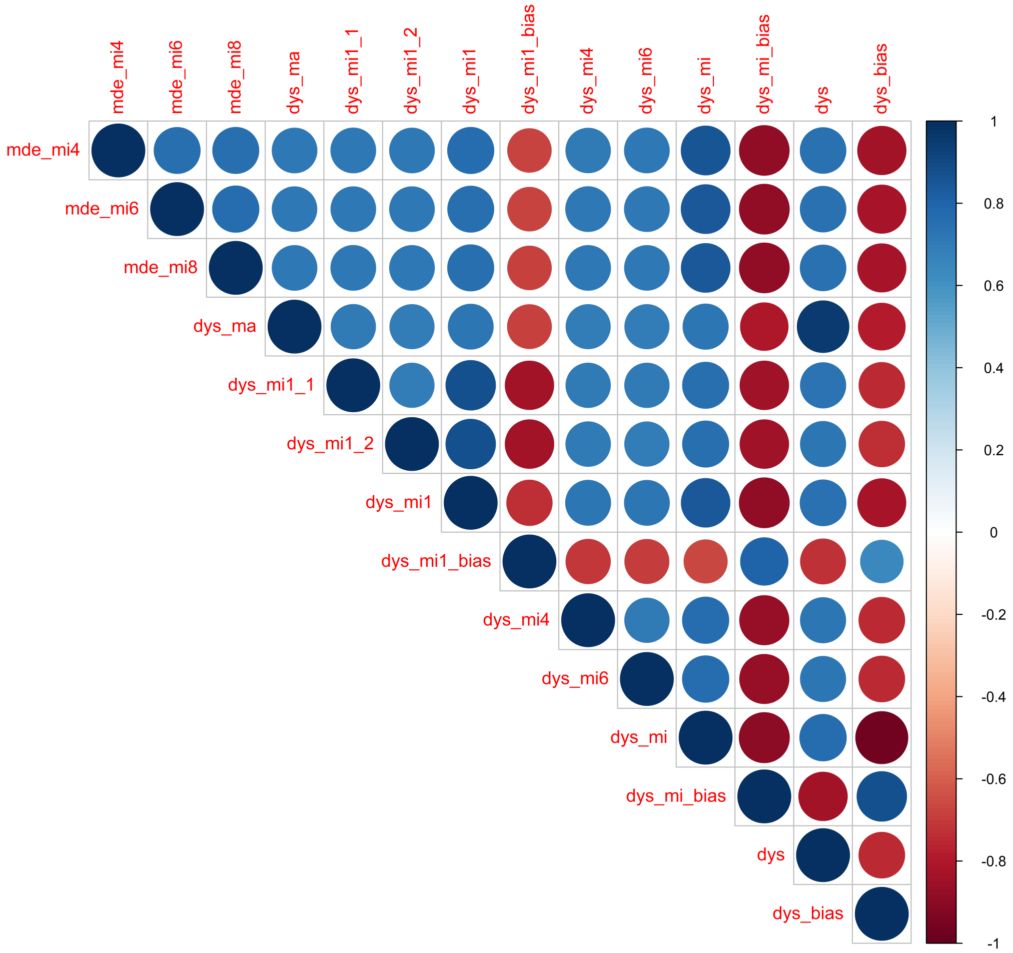

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

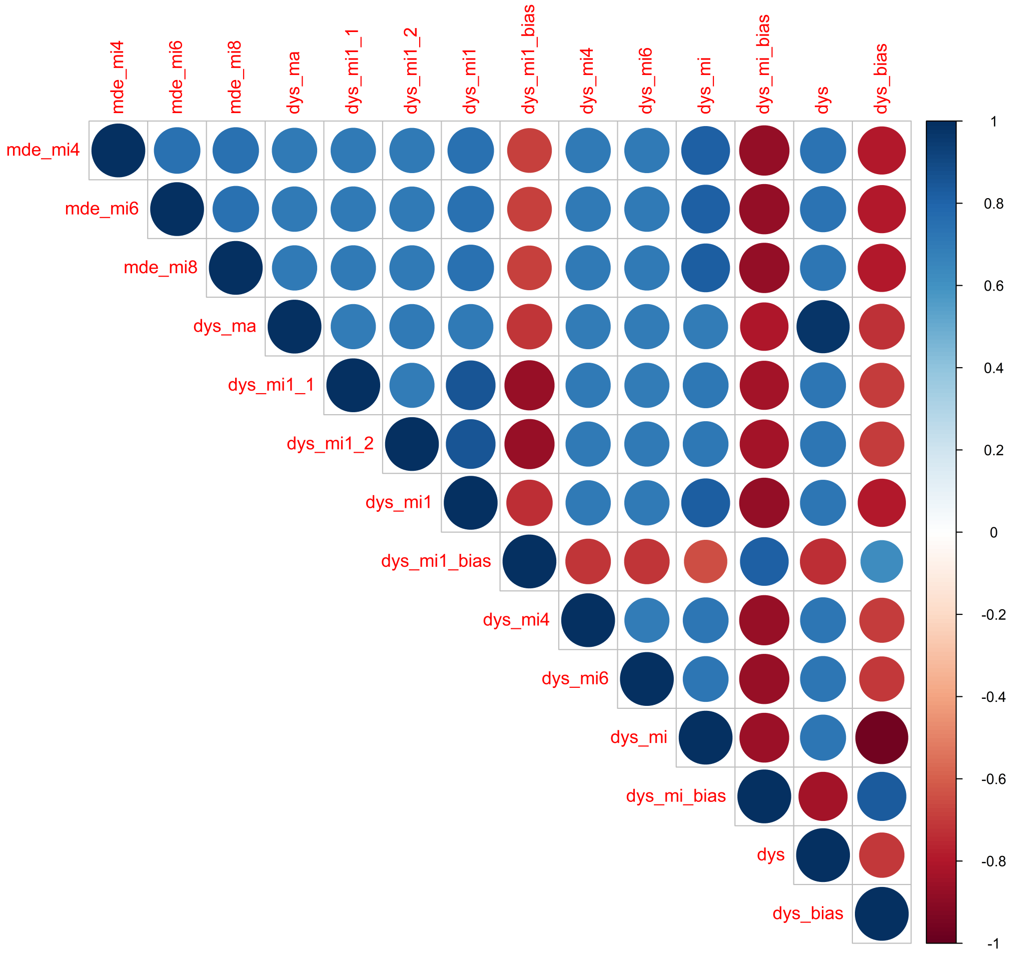

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

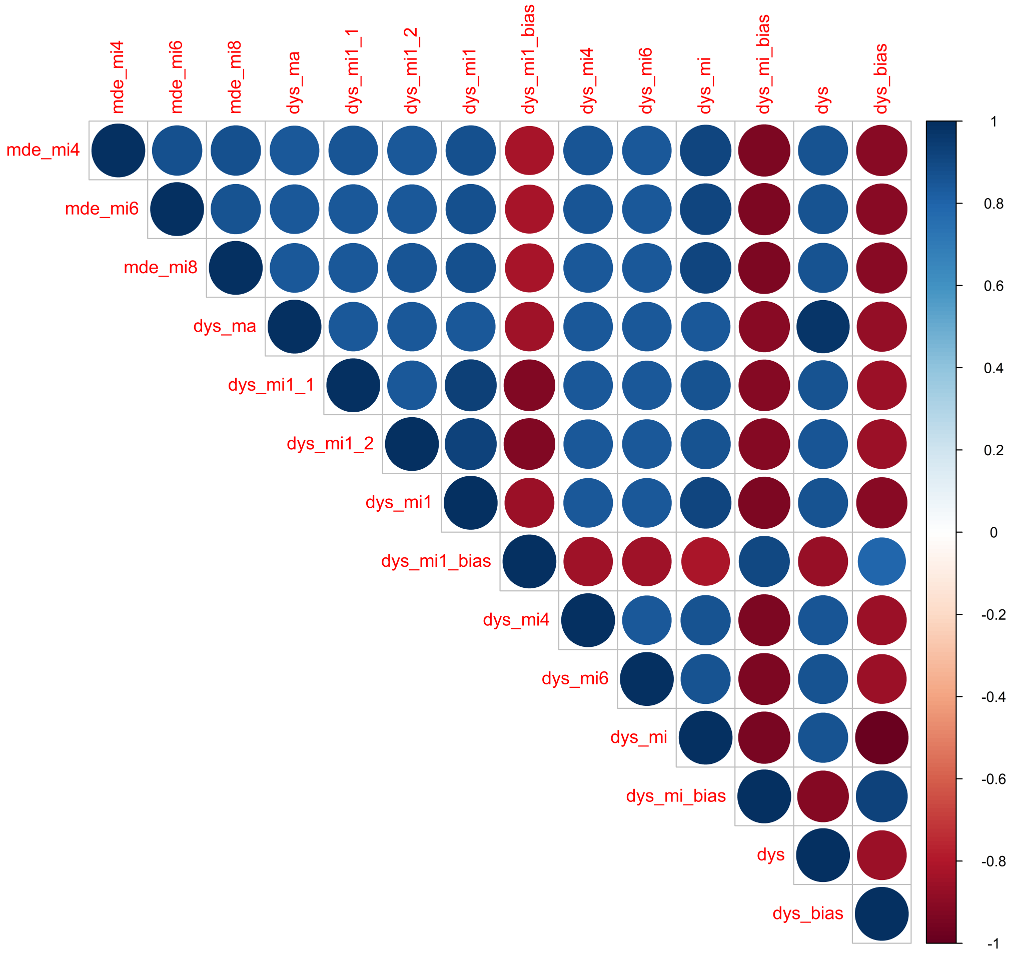

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

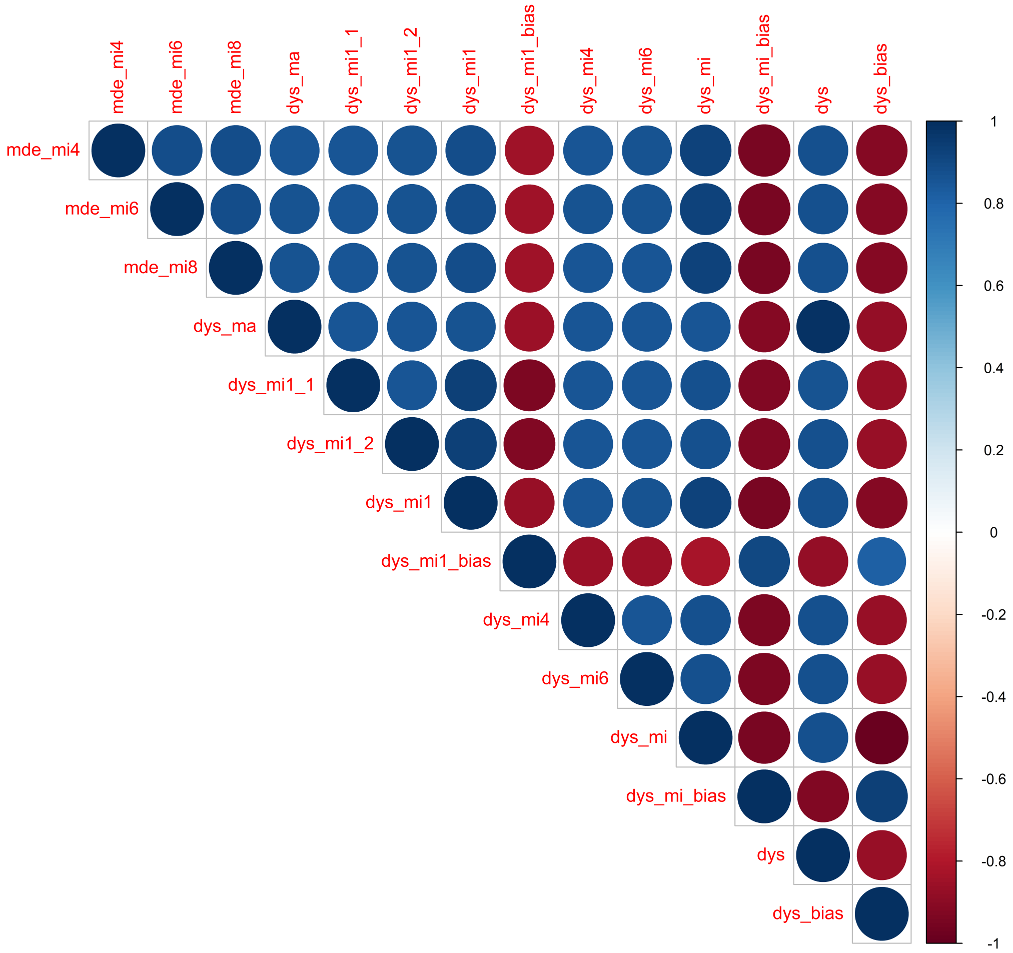

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

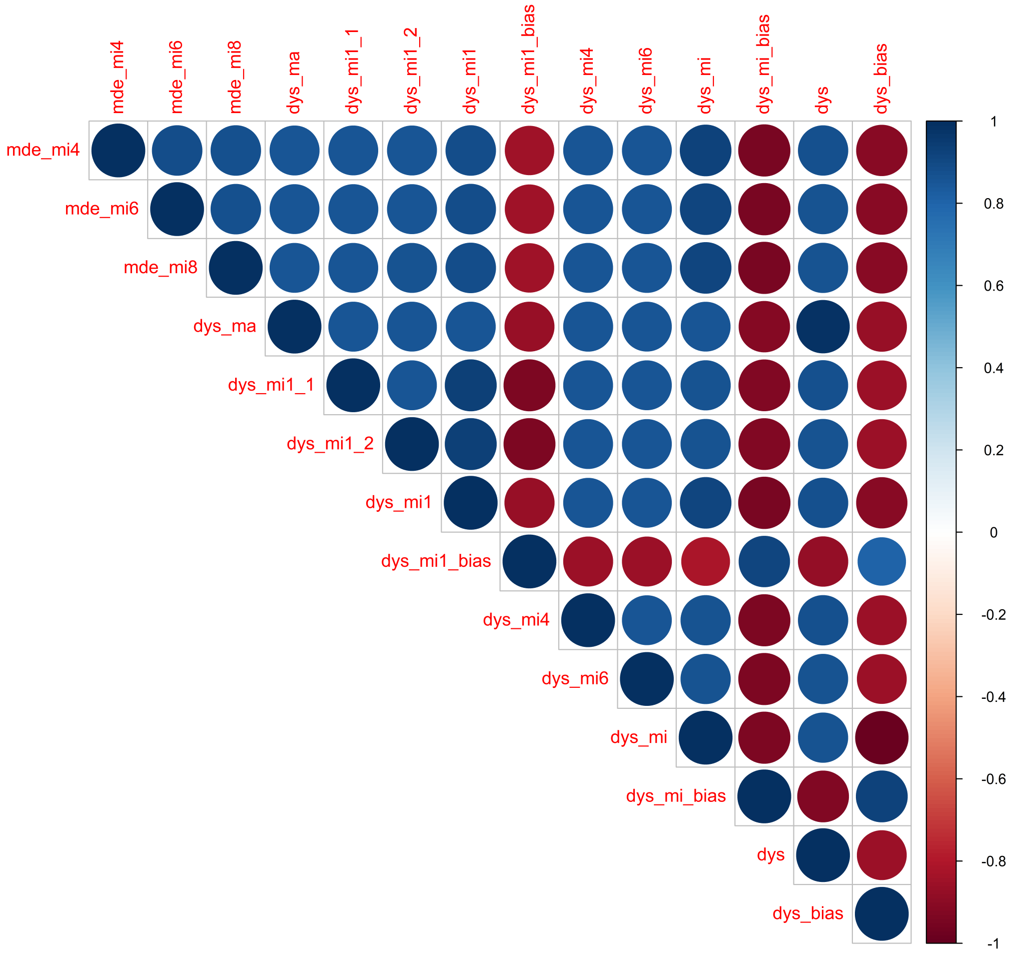

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

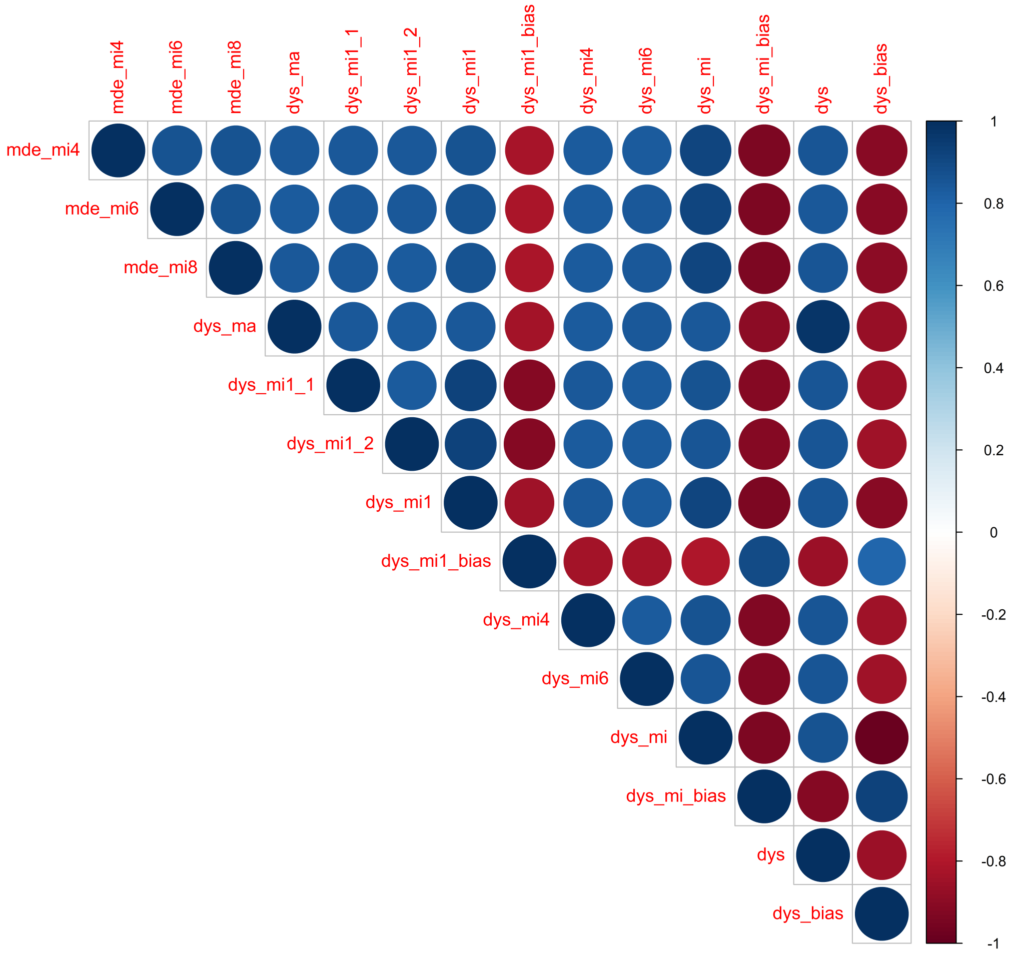

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

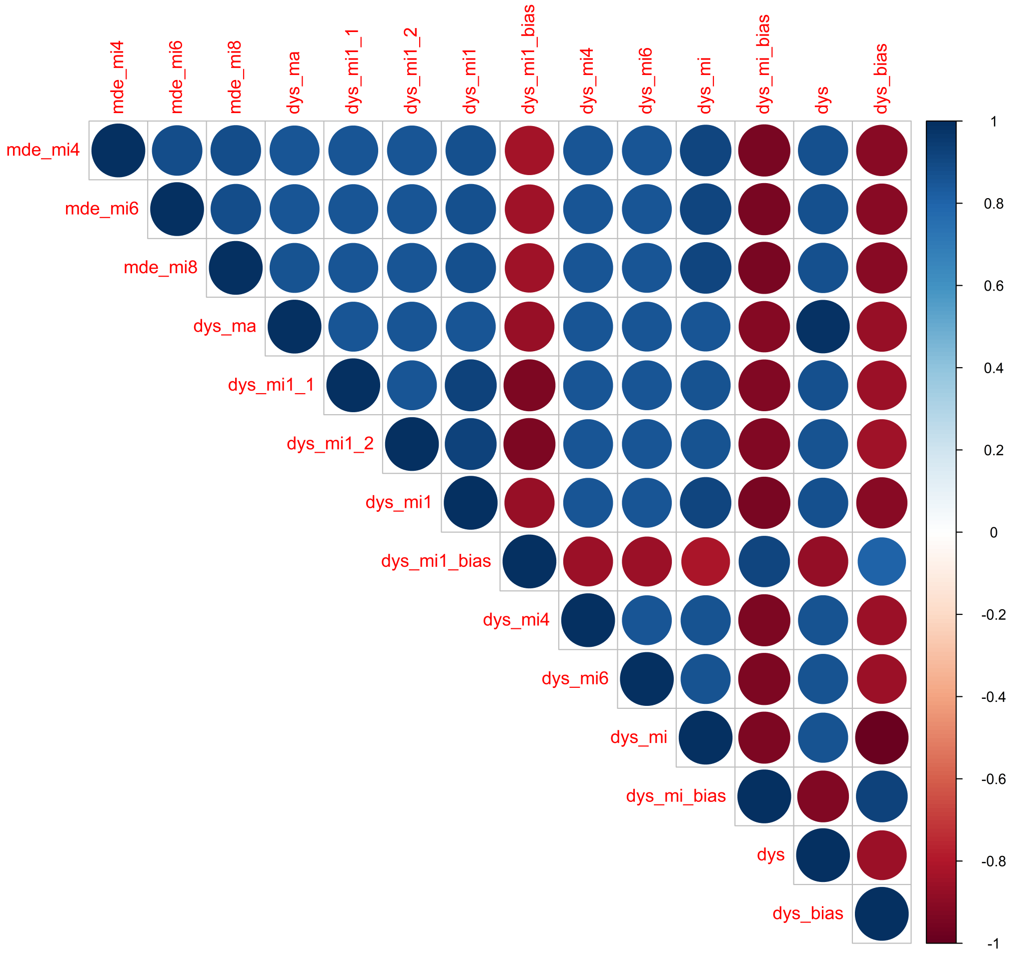

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

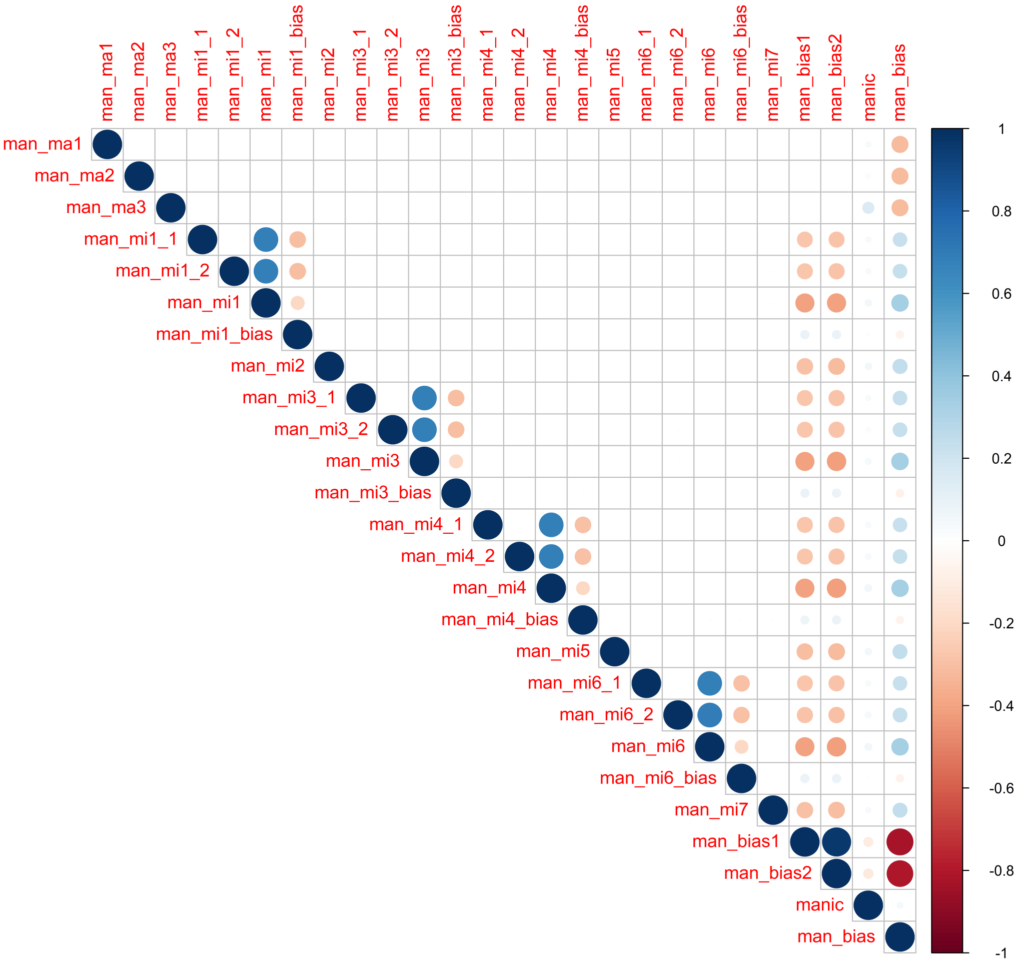

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

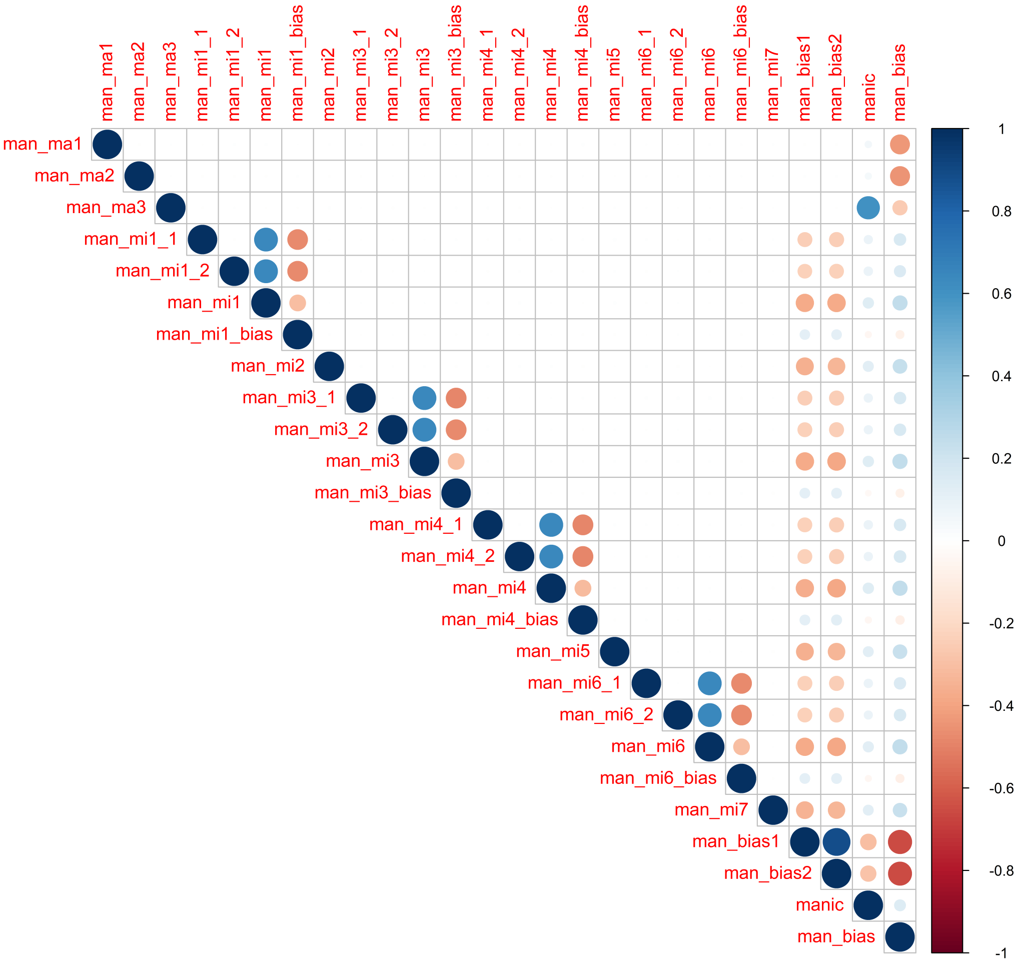

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

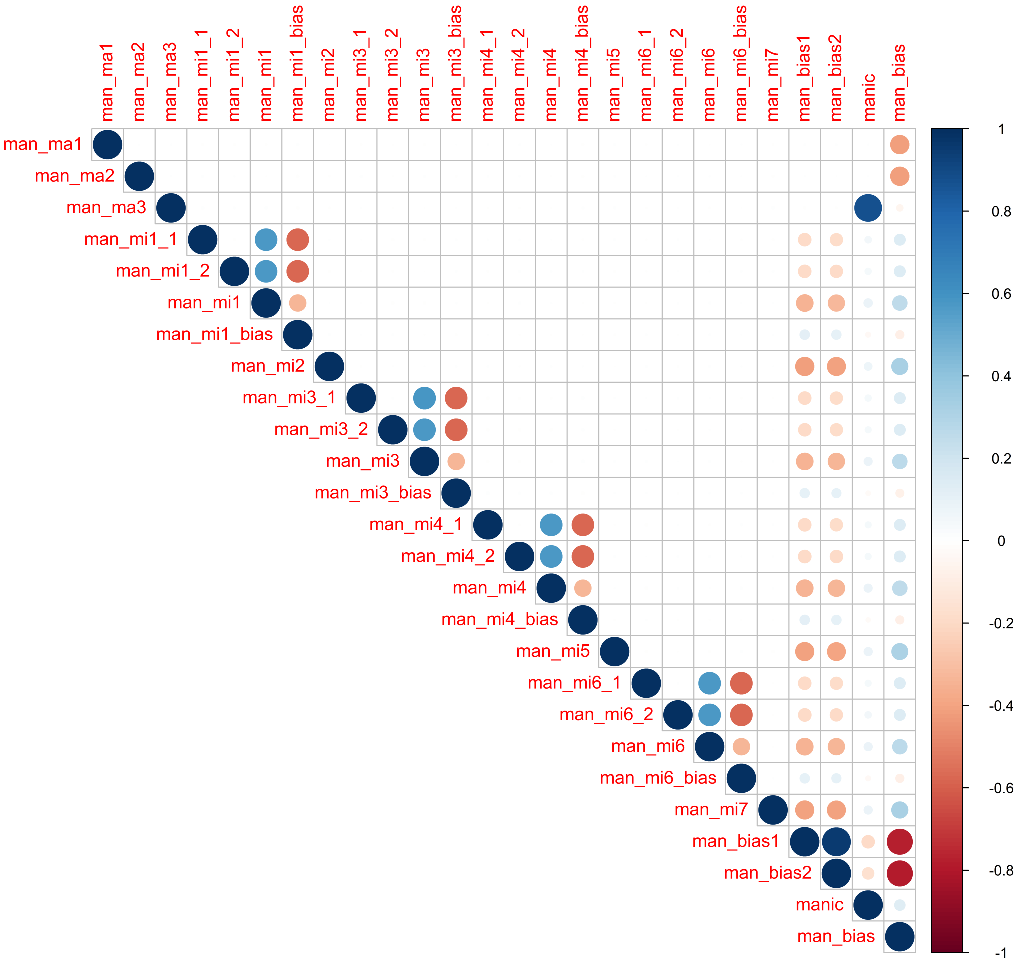

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

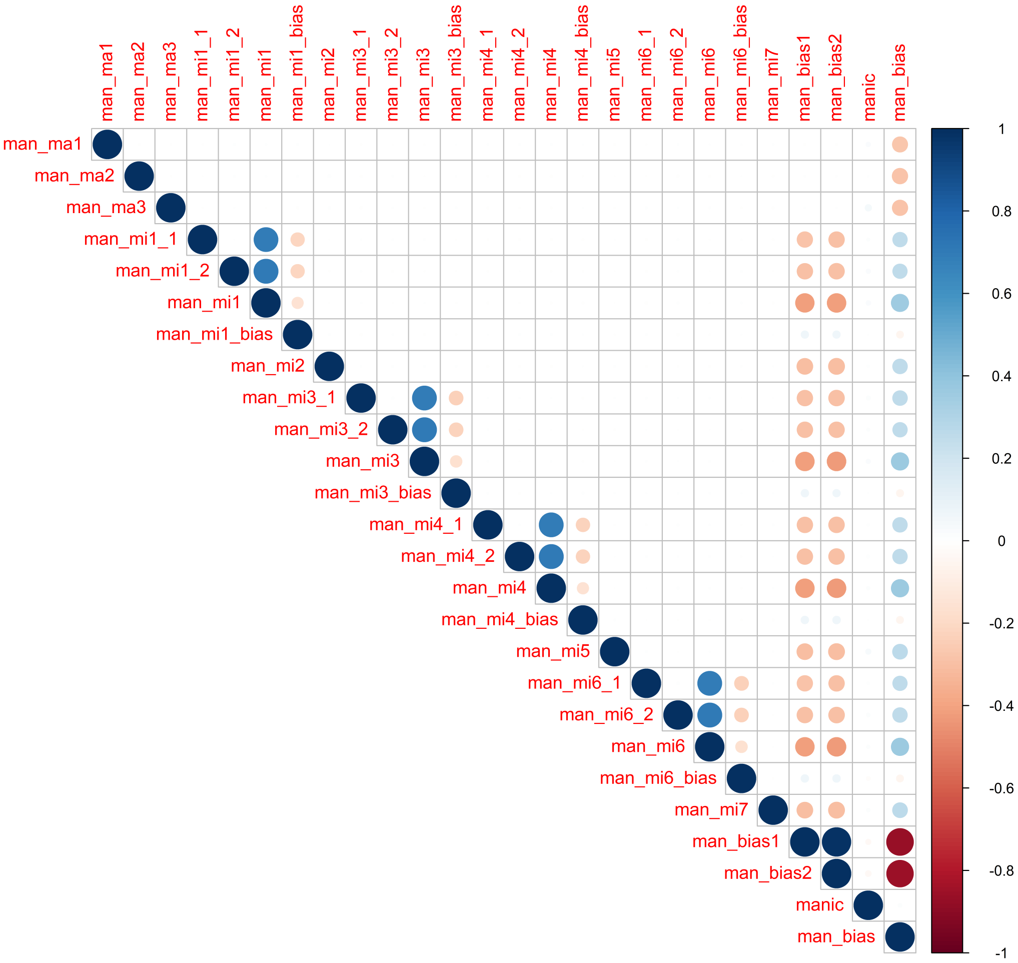

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

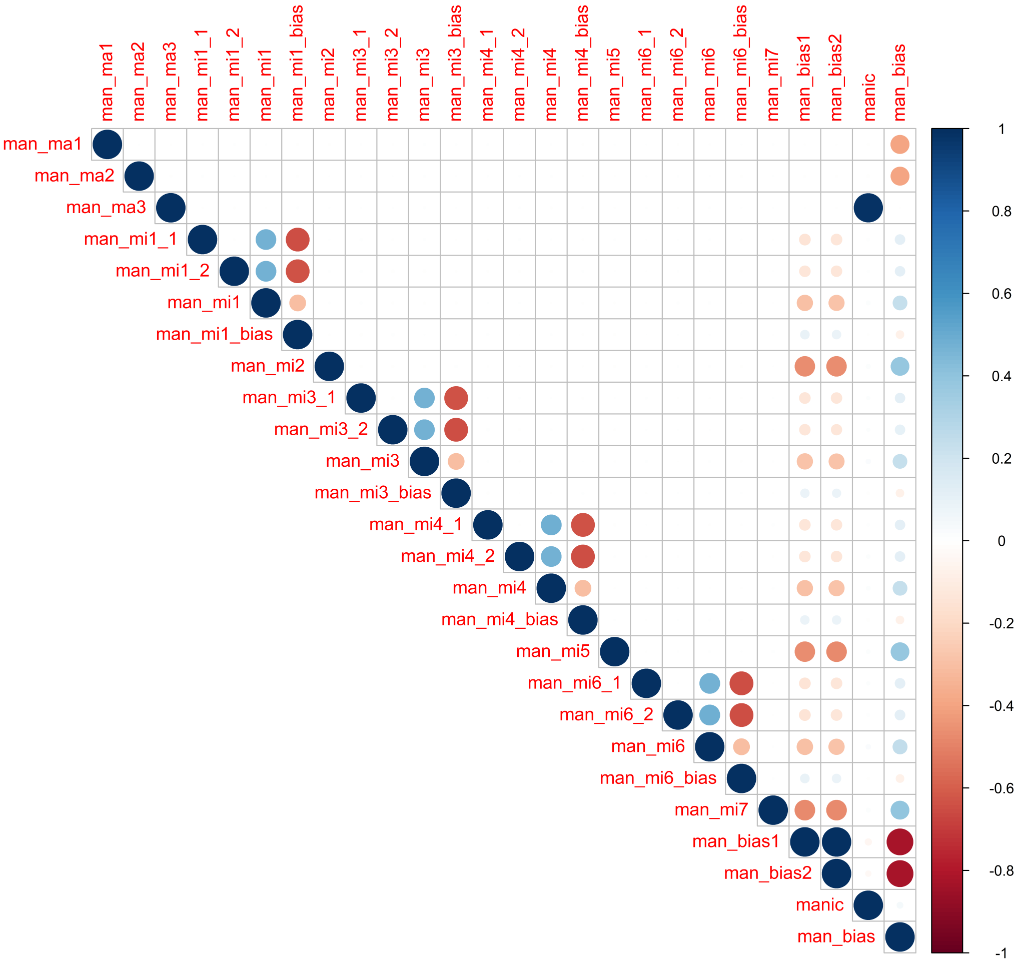

mde_mi4
mde_mi6
mde_mi8
dys_ma
dys_mi1_1
dys_mi1_2
dys_mi1
dys_mi1_bias
dys_mi4
dys_mi6
dys_mi
dys_mi_bias
dys
dys_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

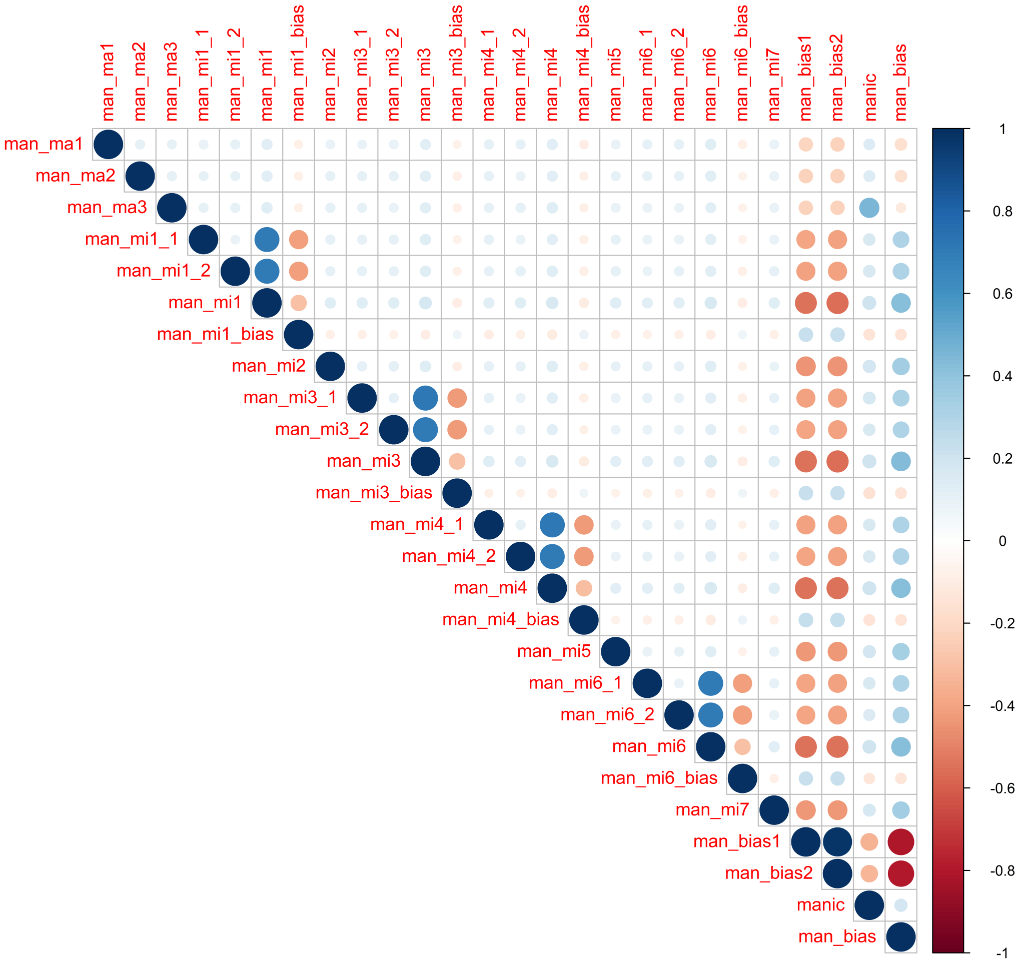

man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

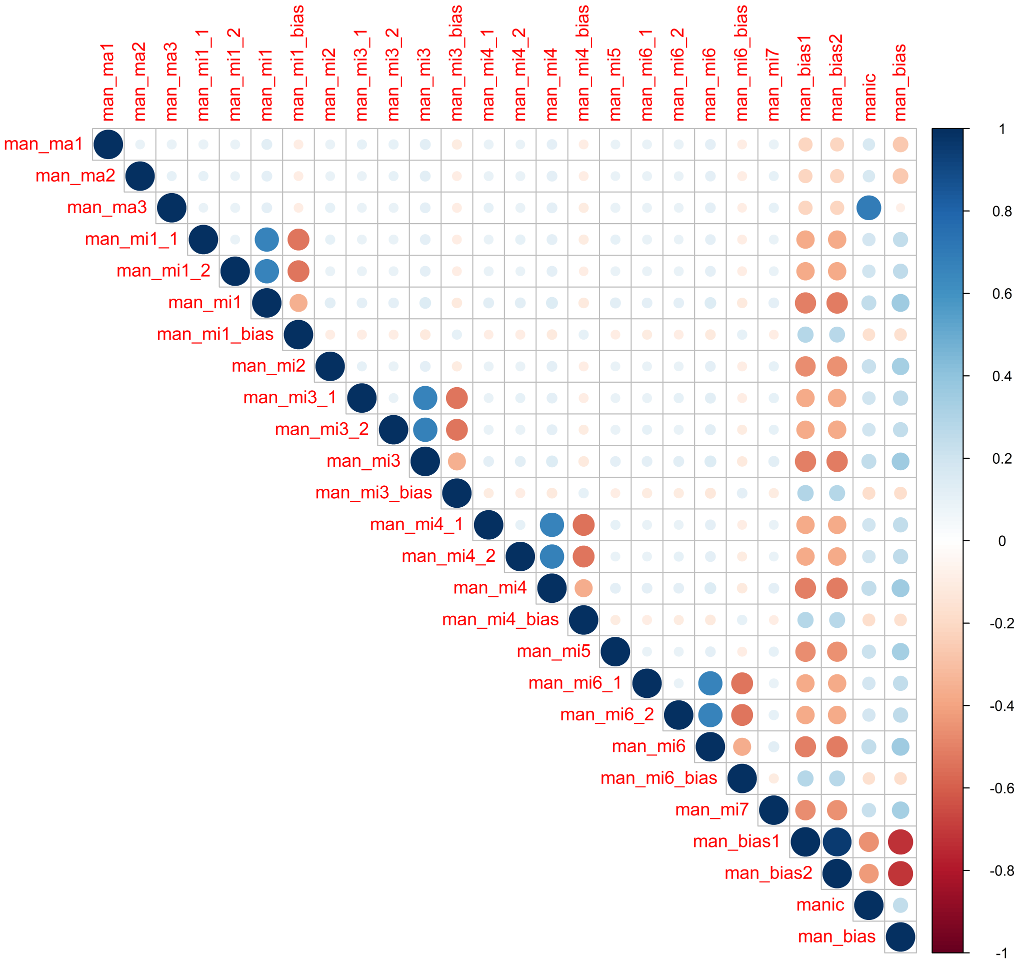

man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

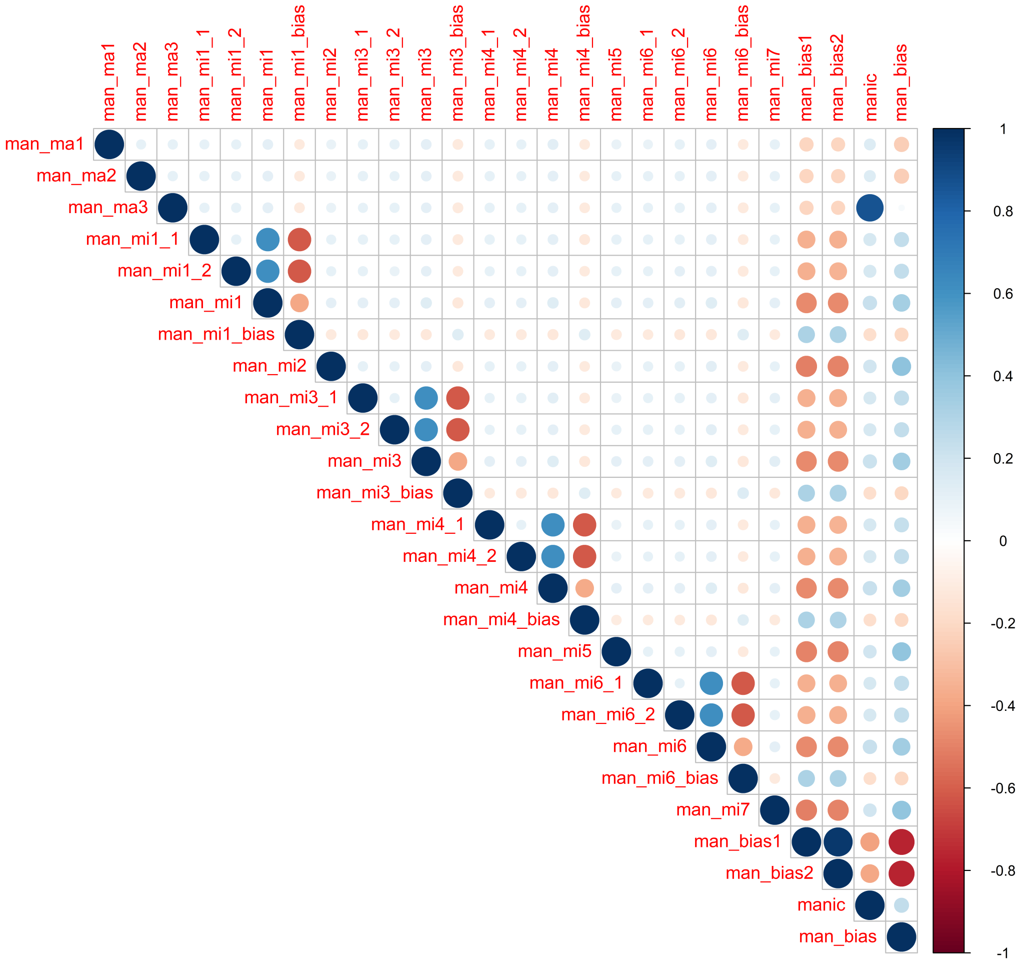

man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

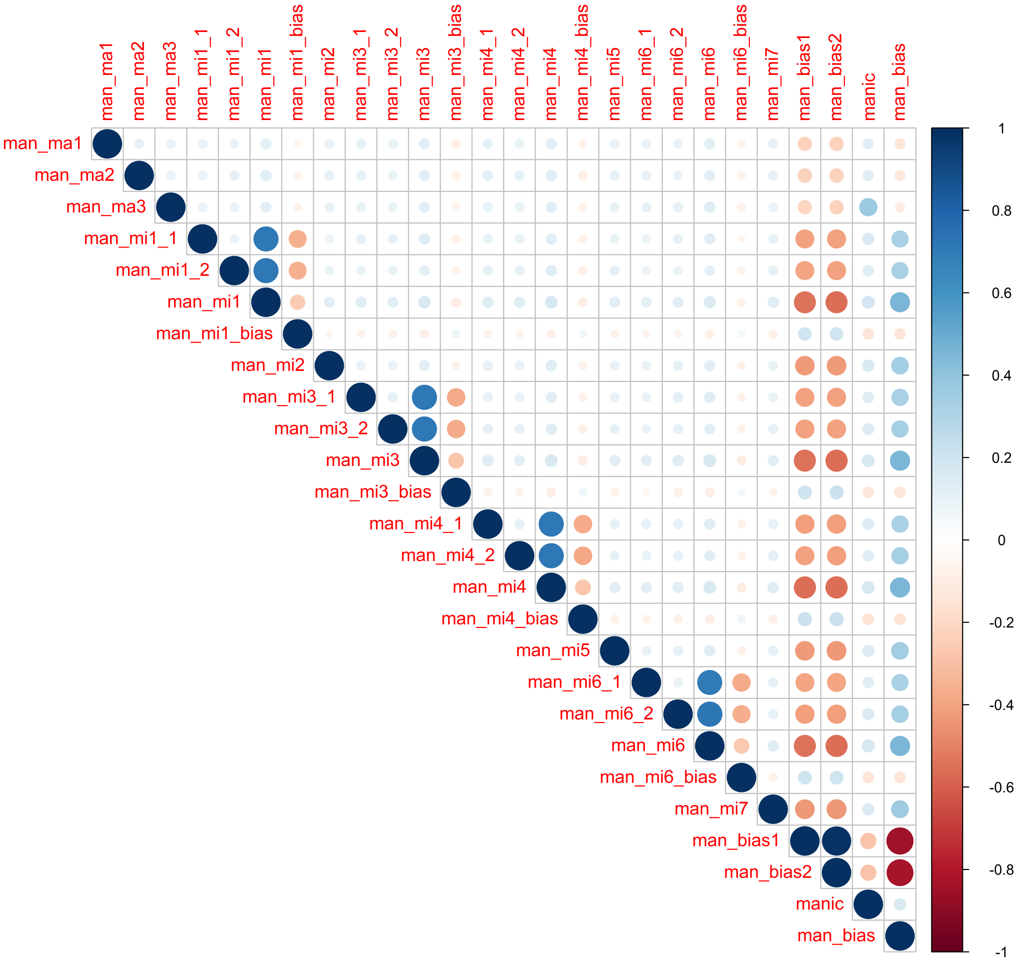

man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

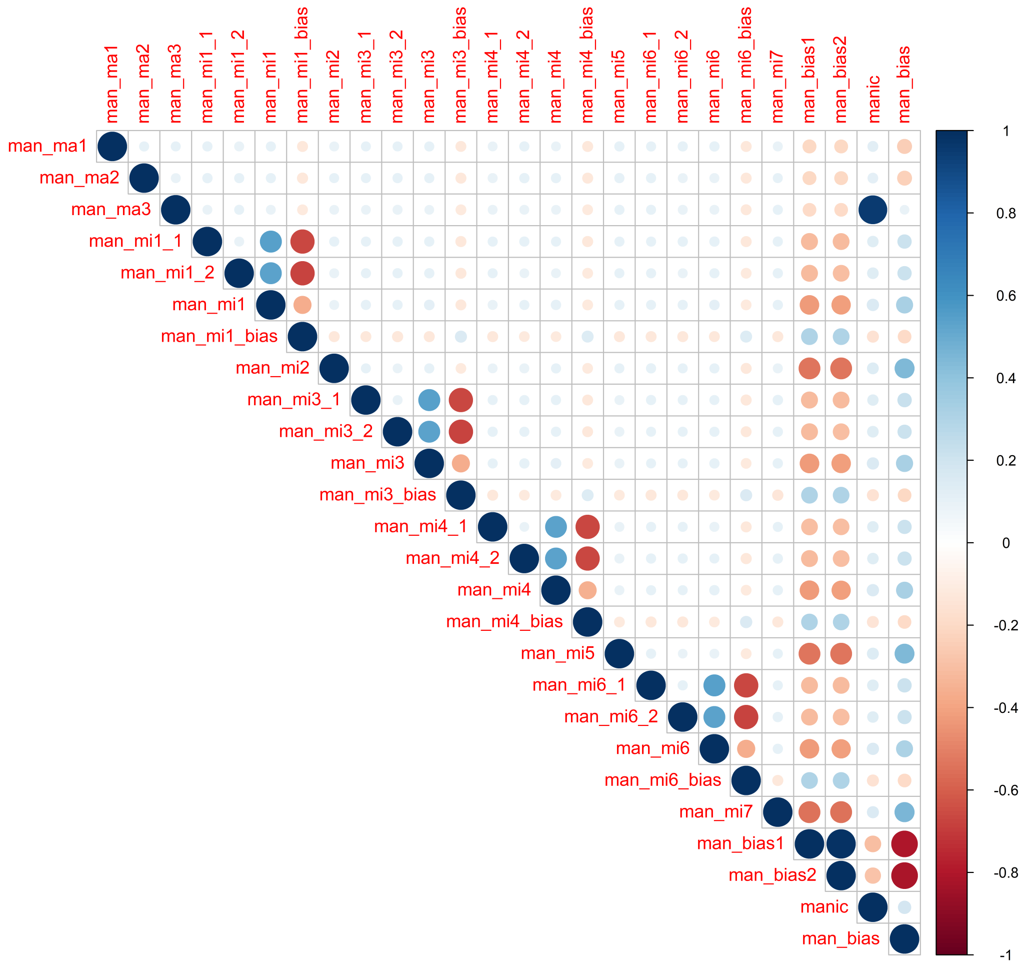

man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

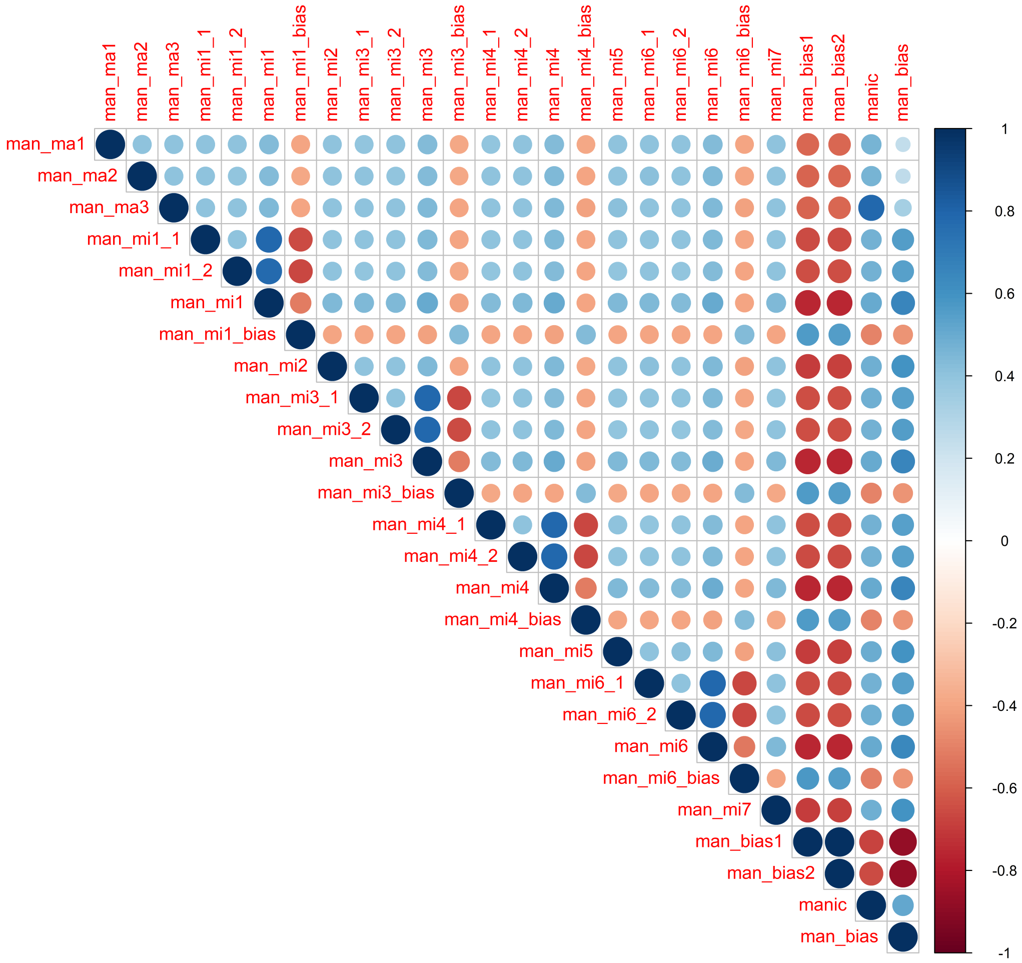

man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

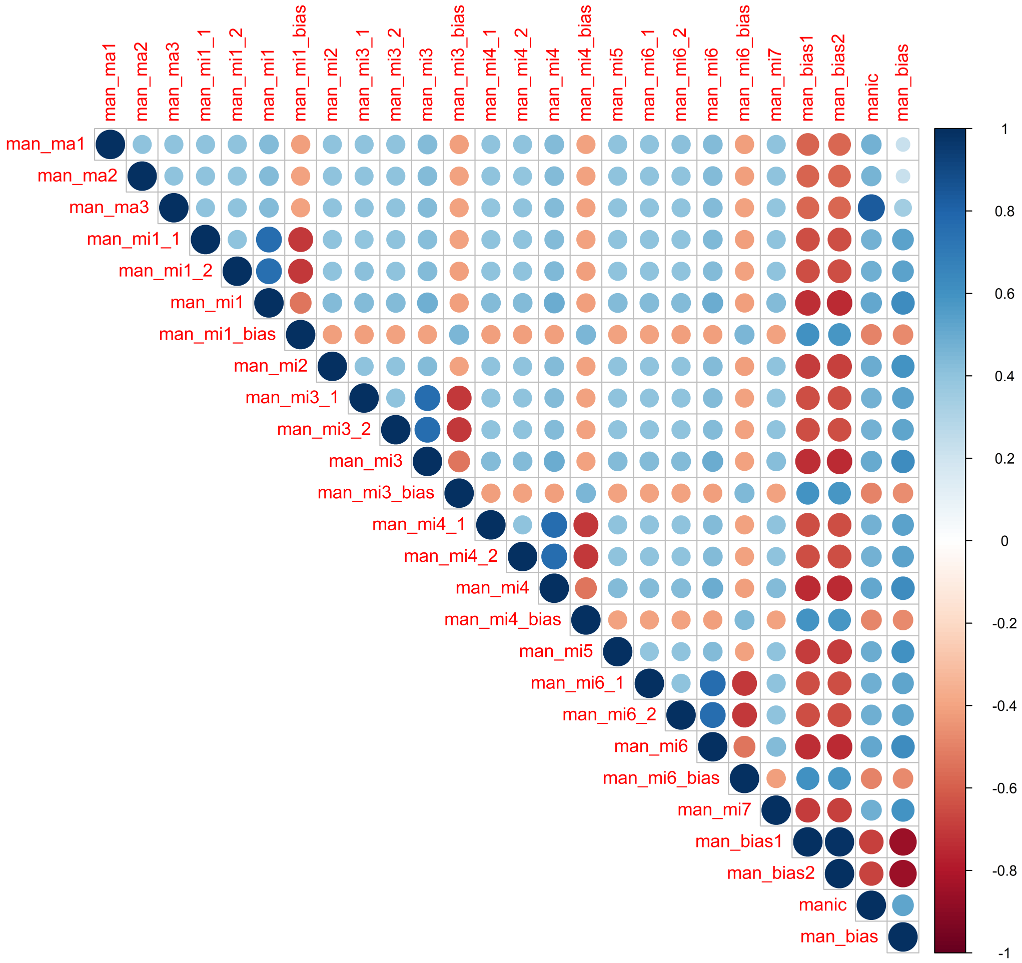

man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

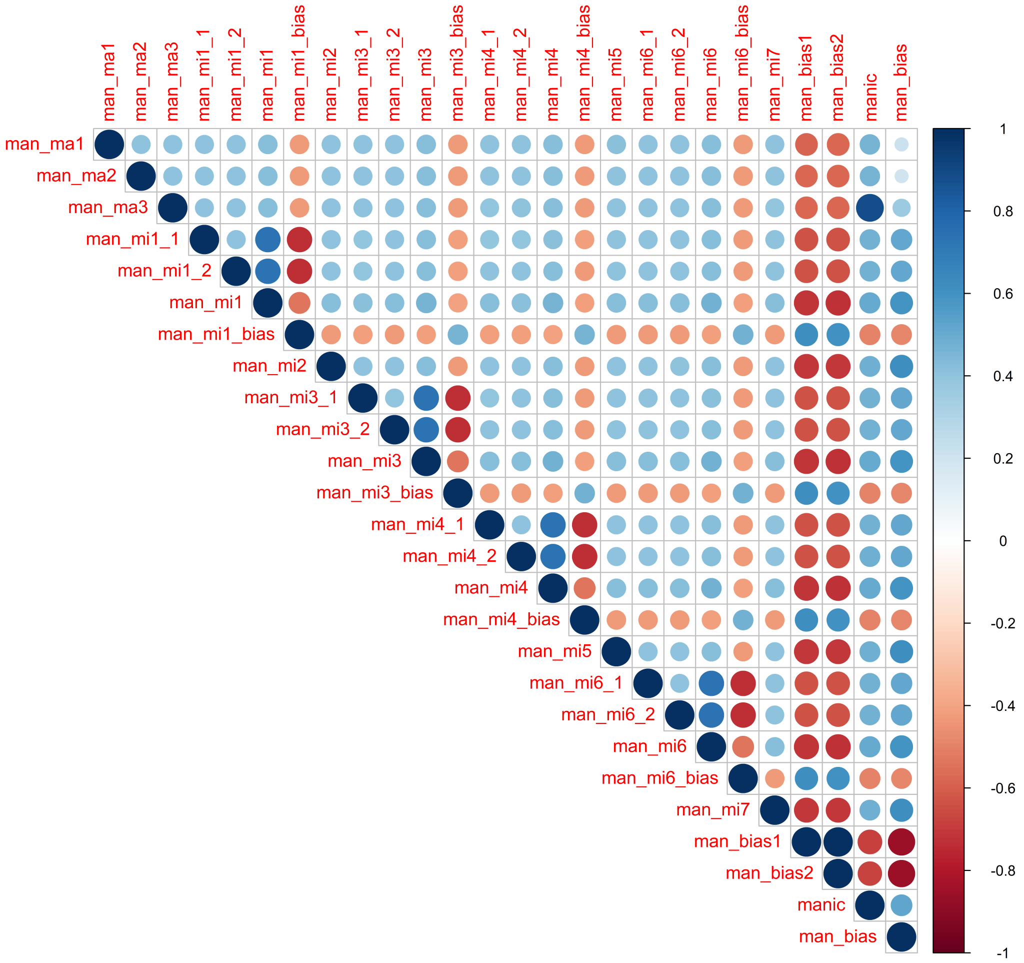

man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

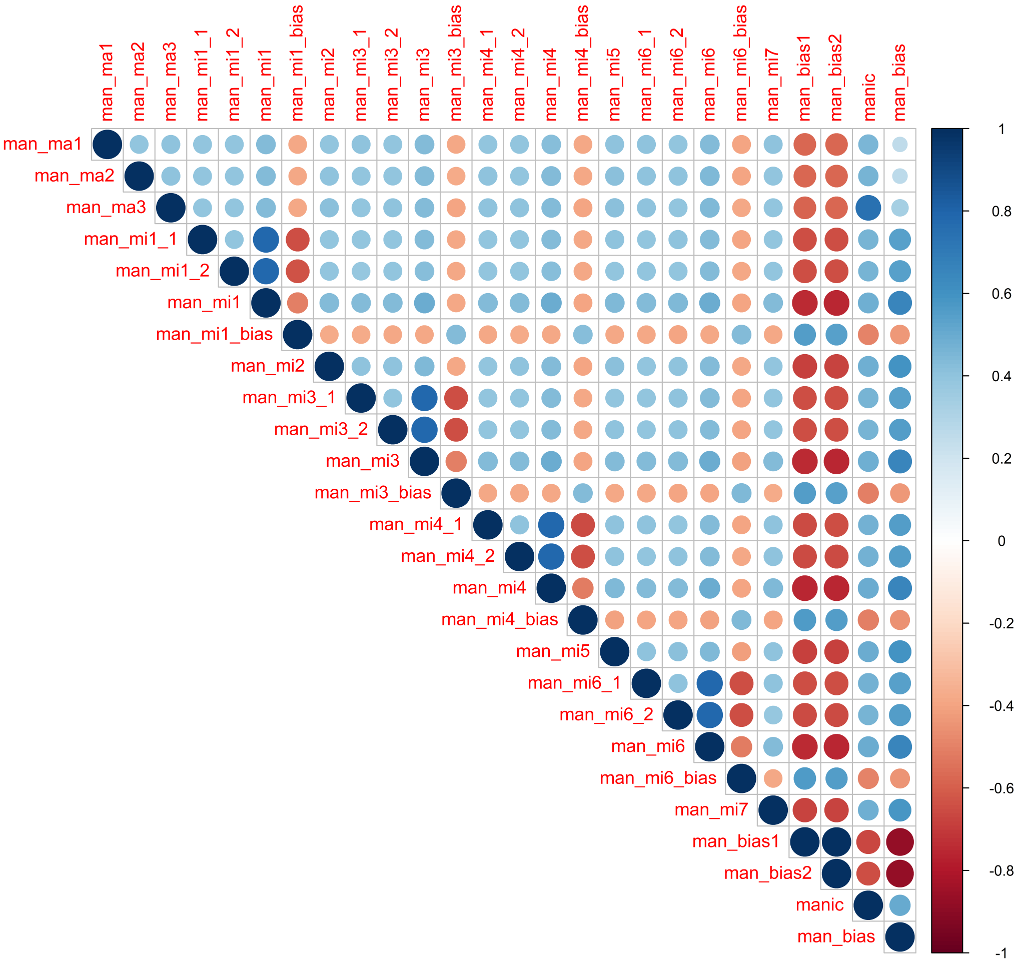

man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

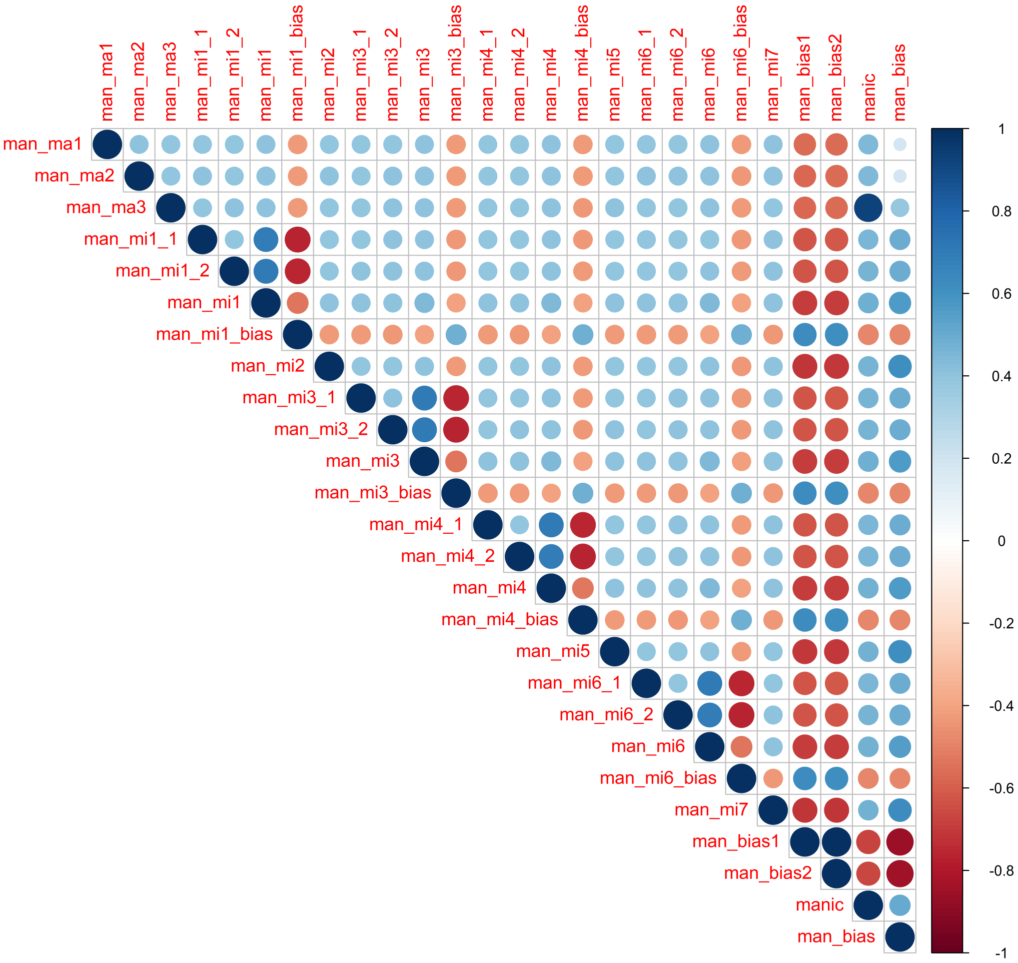

man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


man_ma1
man_ma2
man_ma3
man_mi1_1
man_mi1_2
man_mi1
man_mi1_bias
man_mi2
man_mi3_1
man_mi3_2
man_mi3
man_mi3_bias
man_mi4_1
man_mi4_2
man_mi4
man_mi4_bias
man_mi5
man_mi6_1
man_mi6_2
man_mi6
man_mi6_bias
man_mi7
man_bias1
man_bias2
manic
man_bias
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1