ML-VAR Analyses

Sebastian Castro-Alvarez

Laura F. Bringmann

February 17, 2023

In this document, we analyze intensive longitudinal data of emotion dynamics. To analyze these data, we use the multilevel-vector autoregressive model (ML-VAR) and the vector autoregressive model (VAR). We also include all R code needed to run and replicate the analyses.

To start, we first load into our R environment all the required R packages:

packages <- rownames(installed.packages())  
if (!"psych" %in% packages) {install.packages("psych")}  
if (!"naniar" %in% packages) {install.packages("naniar")}  
if (!"lubridate" %in% packages) {install.packages("lubridate")}  
if (!"remotes" %in% packages) {install.packages("remotes")}  
if (!"esmpack" %in% packages) {  
 remotes::install\_github("secastroal/esmpack")  
 }  
library(esmpack)  
library(foreign)  
library(MplusAutomation)  
  
invisible(lapply(list.files("../R/", full.names = TRUE), source))  
rm(packages)

## Data Preprocessing

In this section, we read and clean the data to obtain the final sample that we use for the analyses.

mlvardata <- read.csv2("deid\_cleaned\_lifepak\_ttt\_phase\_1.csv",   
 header = TRUE, sep = ",")  
mlvardata <- mlvardata[order(mlvardata$lifepak\_id), ]

The data has 14 variables: ID, response number, beep time, response time, time difference (probably how long did it take the person to complete the questionnaire), beep completed (variable indicating that the person ‘completed’ the questionnaire on that beep), and scores on 8 emotions.

We start by excluding subject without any valid observation.

id\_exclude <- as.numeric(  
 names(  
 which(  
 calc.nomiss(response\_time, lifepak\_id, mlvardata) == 0  
 )  
 )  
 )  
  
mlvardata <- mlvardata[!(mlvardata$lifepak\_id %in% id\_exclude), ]  
  
rm(id\_exclude)

Regarding the variables indicating the exact notification time and response time, we turn these variables into POSIXct, which is used to manage dates and times in R.

mlvardata$notification\_time <- as.POSIXct(gsub("T", "",   
 mlvardata$notification\_time),   
 tz = "GMT")  
mlvardata$response\_time <- as.POSIXct(gsub("T", "",  
 mlvardata$response\_time),  
 tz = "GMT")

Turn response times into hours. Hours start counting for each individual on the first day that they have a valid observation. The hours correspond to the number of hours since the 00:00 hours of that first day. Therefore, if a person completed the first esm questionnaire at 9:30, the response time is 9.5. Also, if the completed another esm questionnaire the next day at 8:00, then the response time is 32.

tmp\_time <- tapply(mlvardata$response\_time,   
 mlvardata$lifepak\_id,   
 FUN = function(x) {  
 as.numeric(x) - as.numeric(min(x[which(!is.na(x))])) %/%   
 86400 \* 86400  
 },   
 simplify = TRUE)  
  
mlvardata$response\_time <- unlist(tmp\_time)/3600

We also create another time variable indicator which indicates the number of planned beeps but with one additional step when going from one day to another. For example, if the plan was to assess individuals 5 times a day for six days, the indicator in response\_no for the sixth beep of a person is 7 instead of 6. In other words, there is a jump of one unit for every five observations.

tmp\_days <- tapply(mlvardata$notification\_time,   
 mlvardata$lifepak\_id,  
 FUN = function (x) {  
 as.numeric(as.Date(x)) - min(as.numeric(as.Date(x)))  
 })   
mlvardata$response\_no\_night <- mlvardata$response\_no + unlist(tmp\_days)

We get a quick summary of the emotion scores as follows:

summary(mlvardata[, 7:14])

## bad control energy focus   
## Min. : 0.00 Min. : 0.00 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 14.00 1st Qu.: 7.00   
## Median : 21.00 Median : 25.00 Median : 53.00 Median : 40.00   
## Mean : 32.62 Mean : 36.69 Mean : 49.44 Mean : 43.14   
## 3rd Qu.: 61.00 3rd Qu.: 68.00 3rd Qu.: 81.00 3rd Qu.: 75.00   
## Max. :100.00 Max. :100.00 Max. :100.00 Max. :100.00   
## NA's :4216 NA's :4218 NA's :4217 NA's :4214   
## fun interest movement sad   
## Min. : 0.00 Min. : -2.00 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 26.00 1st Qu.: 4.00 1st Qu.: 2.00 1st Qu.: 0.00   
## Median : 61.00 Median : 29.50 Median : 28.00 Median : 23.00   
## Mean : 55.07 Mean : 37.24 Mean : 37.71 Mean : 34.21   
## 3rd Qu.: 82.00 3rd Qu.: 66.00 3rd Qu.: 69.00 3rd Qu.: 64.00   
## Max. :100.00 Max. :100.00 Max. :100.00 Max. :100.00   
## NA's :4218 NA's :4217 NA's :4217 NA's :4219

This shows that the emotions were measured based on a visual analogue scale (VAS) from 0 to 100. For these analyses, we focus on the emotions *interest* and *sad*. Notice, that there are 60 observations with negative scores on the variable *interest*. As this should not be the case, we further explore these observations.

First, we create a subset of the data that only includes these observations and the variables *ID*, *interest*, and *sad*:

mlvardata\_minus0 <- mlvardata[mlvardata[, "interest"] < 0 &   
 !is.na(mlvardata[, "interest"]),   
 c("lifepak\_id", "response\_time",   
 "interest", "sad")]

The negative scores either -1 or -2 and come from 2 subjects. The complete data of these subjects is stored in the following data.frame:

mlvardata\_neg <- mlvardata[mlvardata$lifepak\_id %in%   
 unique(mlvardata\_minus0$lifepak\_id),   
 c("lifepak\_id", "response\_time",   
 "interest", "sad")]

Moreover, the observed time series of these 2 subjects are presented in the following Figure:

par(mfrow = c(2, 1), mar = c(4, 4, 1, 2) + 0.1)  
  
plot.ts(mlvardata\_neg$interest[mlvardata\_neg$lifepak\_id ==   
 unique(mlvardata\_minus0$lifepak\_id)[1]],  
 ylab = "Interest", ylim = c(-5, 100),  
 xlab = "", xlim = c(0, 110), las = 1)  
mtext(paste0("ID = ", unique(mlvardata\_minus0$lifepak\_id)[1]), 3, at = 100)  
plot.ts(mlvardata\_neg$interest[mlvardata\_neg$lifepak\_id ==  
 unique(mlvardata\_minus0$lifepak\_id)[2]],  
 ylab = "Interest", ylim = c(-5, 100),  
 xlab = "Beep", xlim = c(0, 110), las = 1)  
mtext(paste0("ID = ", unique(mlvardata\_minus0$lifepak\_id)[2]), 3, at = 100)

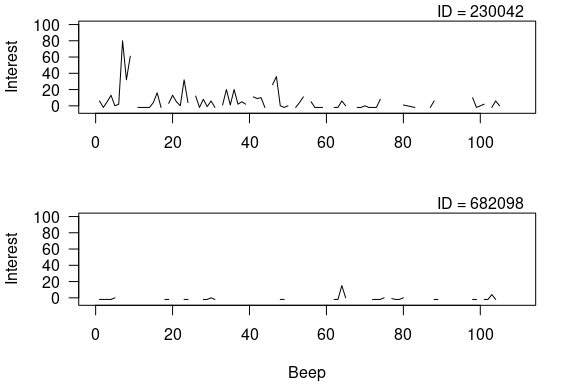


Figure 1: Time series of the variable ‘interest’ of the 2 subjects with scores lower than 0.

As the data from these subjects is not reliable, we decided to exclude them from the analyses. Thus, we create another data.frame without their data.

mlvardata02 <- mlvardata[!(mlvardata$lifepak\_id %in%   
 unique(mlvardata\_minus0$lifepak\_id)), ]

We also want to exclude participants that do not have enough variation on their time series. For this, we first compute the mean and the standard deviation of each variable per person. The means are not necessary but it is still a good a descriptive to comprehend the data at hand. To compute these descriptive statistics, we use the following code:

mean\_interest <- tapply(mlvardata02$interest, mlvardata02$lifepak\_id, mean, na.rm = TRUE)   
mean\_sad <- tapply(mlvardata02$sad, mlvardata02$lifepak\_id, mean, na.rm = TRUE)   
  
sd\_interest <- tapply(mlvardata02$interest, mlvardata02$lifepak\_id, sd, na.rm = TRUE)   
sd\_sad <- tapply(mlvardata02$sad, mlvardata02$lifepak\_id, sd, na.rm = TRUE)

We also plot the mean and the standard deviation of the two variables to visualize their relation.

plot(mean\_interest, mean\_sad)

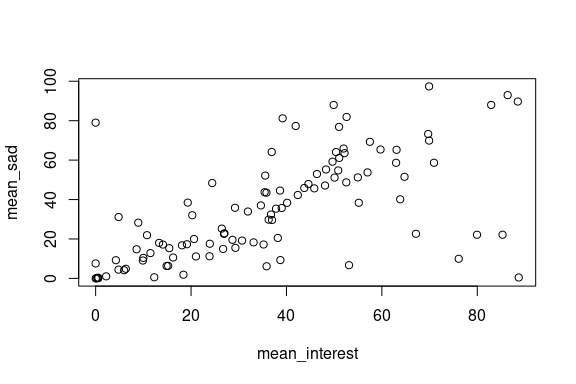


Figure 2: Scatterplot of intraindividual means of ‘interest’ and ‘sad’.

plot(sd\_interest, sd\_sad)  
abline(h = 9.5, col = gray(0.5))  
abline(v = 9.5, col = gray(0.5))

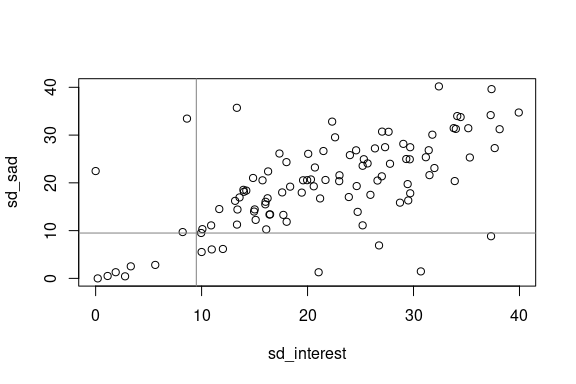


Figure 3: Scatterplot of intraindividual sd of ‘interest’ and ‘sad’.

As mentioned before, we exclude participants with not enough variation. The time series of an individual is considered to not have enough variation if the standard deviation of any of the variables is lower than 9.5. Therefore, to exclude these participants, we do:

sd\_lower <- ifelse(sd\_interest < 9.5 | sd\_sad < 9.5, TRUE, FALSE)  
sd\_exclude <- as.numeric(names(which(sd\_lower)))  
  
mlvardata02 <- mlvardata02[!(mlvardata02$lifepak\_id %in% sd\_exclude), ]  
ids <- unique(mlvardata02$lifepak\_id)

As a result, we have excluded 16 more participants, for a final sample of 88 participants to use in the following analyses.

Now, we also create two subsets of the data. The first subset includes the whole remaining sample and we use the ML-VAR to analyze this data. In this case, we also exclude subjects with less than 5 observations. The second subset includes subjects that complied with at least 60 observations on both variables (*interest* and *sad*). We use the VAR model to analyze the data of each individual on this dataset.

We create the first subset as follows:

# Get compliance per subject for 'interest'  
comp\_interest <- calc.nomiss(interest, lifepak\_id, mlvardata02)  
  
# Get IDs of subjects with less than 5 responses  
id\_exclude <- as.numeric(names(which(comp\_interest < 5)))  
  
# Exclude subjects with less than 5 responses  
data\_mlvar <- mlvardata02[!(mlvardata02$lifepak\_id %in% id\_exclude), ]

For the second subset, we first create a variable that indicates if the person has valid scores on both variables *interest* and *sad*, then, based on this new variable, we select the individuals with more than 60 observations.

# Create indicator variable of valid scores on 'interest' and 'sad'  
mlvardata02$ind\_nomiss <- ifelse(is.na(mlvardata02$interest) &  
 is.na(mlvardata02$sad), NA, 1)  
  
# Get compliance on both variables per person  
comp\_both <- calc.nomiss(ind\_nomiss, lifepak\_id, mlvardata02)  
  
# Get IDs of subjects with 60 or more responses on both variables  
id\_include <- as.numeric(names(which(comp\_both >= 60)))  
  
# Exclude subjects with less than 60 responses on both variables  
data\_var <- mlvardata02[mlvardata02$lifepak\_id %in% id\_include, ]  
  
# Remove indicator variable in final subset  
data\_var <- data\_var[, 1:15]

## Sample’s Descriptives

After cleaning the data, we get that ML-VAR data includes responses from 87 participants. These participants replied between 5 and 102 beeps. In particular, the distribution of the compliance is presented in Figure 4:

hist(calc.nomiss(interest, lifepak\_id, data\_mlvar),  
 ylim = c(0, 8),  
 xlim = c(0, 120),  
 xlab = "Number of complied beeps",  
 las = 1,  
 breaks = 40,  
 main = "")

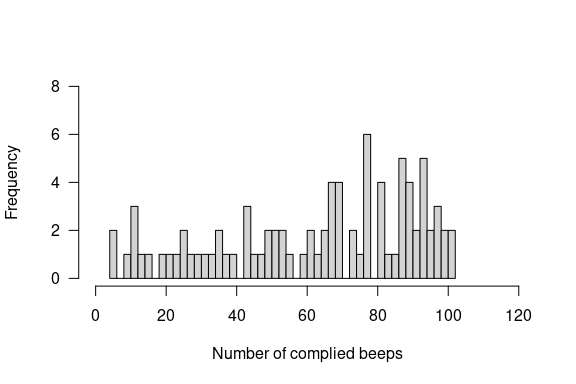


Figure 4: Compliance on variable ‘interest’

We can also see the distribution of the missing values on both variables in the following Figures:

tmp <- reshape(na.omit(data\_mlvar[, c(1, 2, 12)]),   
 direction = "wide",   
 idvar = "lifepak\_id",   
 timevar = "response\_no")  
naniar::vis\_miss(tmp[, -1], show\_perc\_col = FALSE)

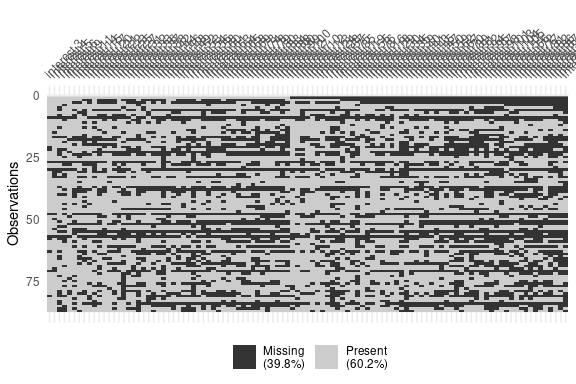


Figure 5: Missing values visualization of the variable ‘interest’

rm(tmp)

tmp <- reshape(na.omit(data\_mlvar[, c(1, 2, 14)]),   
 direction = "wide",   
 idvar = "lifepak\_id",   
 timevar = "response\_no")  
naniar::vis\_miss(tmp[, -1], show\_perc\_col = FALSE)

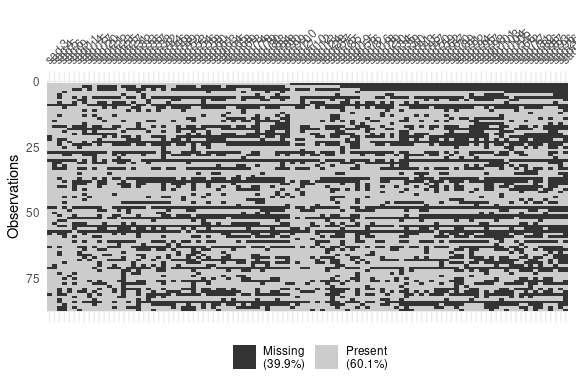


Figure 6: Missing values visualization of the variable ‘sad’

rm(tmp)

Next, Figure 7 presents the overall distribution of each variable of interest across the sample:

par(mfrow = c(1, 2))  
hist(data\_mlvar$interest,   
 main = "",  
 las = 1,  
 ylim = c(0, 1500),  
 xlab = "Interest",  
 breaks = 40)  
hist(data\_mlvar$sad,  
 main = "",  
 las = 1,  
 ylim = c(0, 1500),  
 xlab = "Sad",  
 breaks = 40)

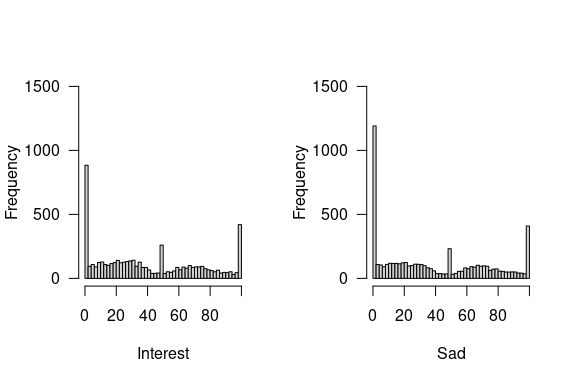


Figure 7: Overall distribution of the variables ‘interest’ and ‘sad’

The distributions showed very high peaks at 0, 50 and 100. In particular the frequency of each of these responses can be checked as follows:

sum(data\_mlvar$sad == 0, na.rm = TRUE)

## [1] 1095

sum(data\_mlvar$sad == 50, na.rm = TRUE)

## [1] 159

sum(data\_mlvar$sad == 100, na.rm = TRUE)

## [1] 390

sum(data\_mlvar$interest == 0, na.rm = TRUE)

## [1] 814

sum(data\_mlvar$interest == 50, na.rm = TRUE)

## [1] 202

sum(data\_mlvar$interest == 100, na.rm = TRUE)

## [1] 405

## Individuals’ Descriptives

For the VAR analyses, the sample got reduced to 53 subjects. These individuals replied to at least 60 beeps on both variables of interest. Similarly as presented before, the distribution of the missing values is presented in Figures 8 and 9.

tmp <- reshape(na.omit(data\_var[, c(1, 2, 12)]),   
 direction = "wide",   
 idvar = "lifepak\_id",   
 timevar = "response\_no")  
naniar::vis\_miss(tmp[, -1], show\_perc\_col = FALSE)

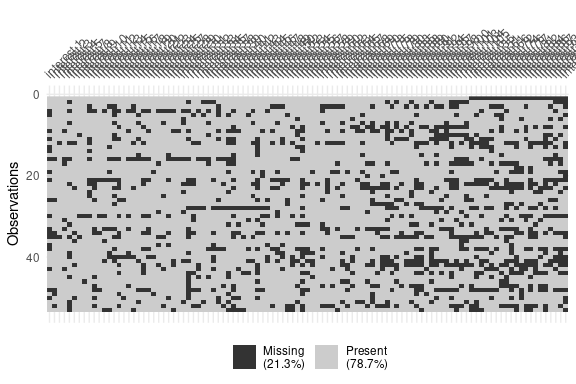


Figure 8: Missing values visualization of the variable ‘interest’, VAR subset.

rm(tmp)

tmp <- reshape(na.omit(data\_var[, c(1, 2, 14)]),   
 direction = "wide",   
 idvar = "lifepak\_id",   
 timevar = "response\_no")  
naniar::vis\_miss(tmp[, -1], show\_perc\_col = FALSE)

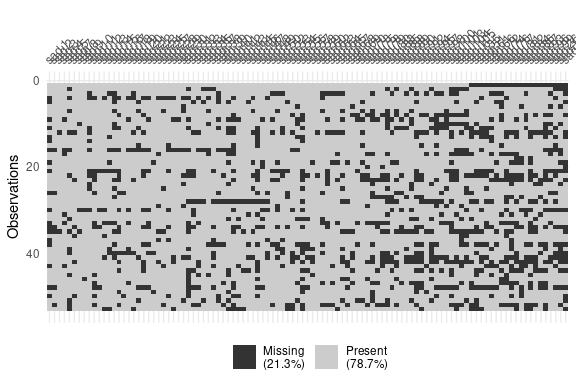


Figure 9: Missing values visualization of the variable ‘sad’, VAR subset

rm(tmp)

To have a closer look at the individual time series, we plot the time series plot of each individual for the two variables and save it on a pdf file:

ids <- unique(data\_var$lifepak\_id)  
pdf("TimeSeries.pdf")  
for (i in 1:length(ids)) {  
 plot.ts(data\_var[data\_var$lifepak\_id == ids[i], c("interest", "sad")],  
 main = paste0("Subject ID = ", ids[i]),   
 ylab = "Emotion Scores",  
 plot.type = "single", ylim = c(0, 100),   
 col = gray(c(0.1, 0.5)), las = 1, lwd = 1.5)  
 mtext(paste0("Interest: Mean = ",   
 round(mean(data\_var[data\_var$lifepak\_id == ids[i], "interest"],   
 na.rm = TRUE), 2), ", sd = ",   
 round(sd(data\_var[data\_var$lifepak\_id == ids[i], "interest"],   
 na.rm = TRUE), 2)),   
 side = 3, at = 1, line= 0.8, adj = 0, cex = 0.7)  
 mtext(paste0("Sad: Mean = ",   
 round(mean(data\_var[data\_var$lifepak\_id == ids[i], "sad"],   
 na.rm = TRUE), 2), ", sd = ",   
 round(sd(data\_var[data\_var$lifepak\_id == ids[i], "sad"],   
 na.rm = TRUE), 2)),   
 side = 3, at = 1, adj = 0, cex = 0.7)  
 legend("bottomright", c("Interest", "Sad"),   
 col = gray(c(0.1, 0.5)), lwd = 1.5, bg = "white")  
}  
rm(i)  
dev.off()

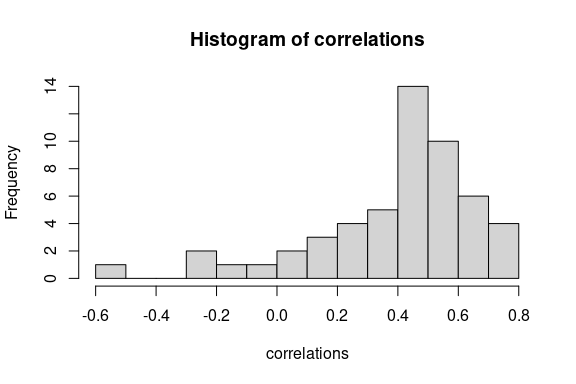
## png   
## 2

Lastly, we look at the within person correlations between *interest* and *sad*. The correlations are computed as follows:

correlations <- rep(NA, length(ids))  
  
for (i in 1:length(ids)) {  
 correlations[i] <- cor(data\_var[data\_var$lifepak\_id == ids[i], "interest"],  
 data\_var[data\_var$lifepak\_id == ids[i], "sad"],   
 use = "complete.obs")  
}  
rm(i)

The correlations have the following distribution:

hist(correlations, breaks = 15)



rm(correlations)

## VAR Analyses per Person

To run the VAR analyses per person in Mplus, we used the following for loop:

varfit <- list()  
  
if (!(dir.exists("Mplus"))) {dir.create("Mplus")}  
  
setwd("Mplus/")  
  
for (i in 1:length(ids)) {  
 tmp <- data\_var[data\_var$lifepak\_id == ids[i],   
 c("response\_no\_night", "interest", "sad")]  
 tmp <- na.omit(tmp)  
 names(tmp) <- c("time", "int", "sad")  
   
 varfit[[i]] <- var2Mplus(y = c("int", "sad"), time = "time", data = tmp,  
 filename = paste0("id", ids[i], ".dat"),  
 runmodel = !file.exists(paste0("id", ids[i], ".out")))  
}  
rm(i, tmp)  
setwd("..")

This loop also uses a customized function that allows exporting the data to Mplus, writing the model’s syntax, run the model in Mplus, and read the Mplus output into R. In this set of analyses, we also included time (represented in response\_no\_night) as a predictor to detrend the time series.

Once the analyses were finished, we extracted the estimated parameters of interest (lagged and cross-lagged effects and linear trends) from the VAR analyses with the following code:

parameters <- paste0("par", 1:9)  
results <- lapply(varfit, function(x) {  
 out <- cbind(parameters,   
 x$parameters$unstandardized[c(1:7, 10:11),   
 c(3, 6, 7)])  
 return(out)  
})  
  
results <- do.call(rbind, results)  
rm(parameters)

Next, we plot the estimated parameters with their 95% credibility intervals per person.

tmp <- results[results$parameters == "par1", ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:53,   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

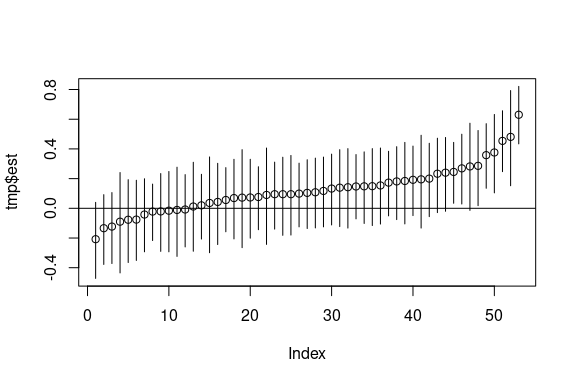


Figure 10: VAR with linear trend: Autoregressive effect of interest.

rm(tmp)

tmp <- results[results$parameters == "par2", ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:53,   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

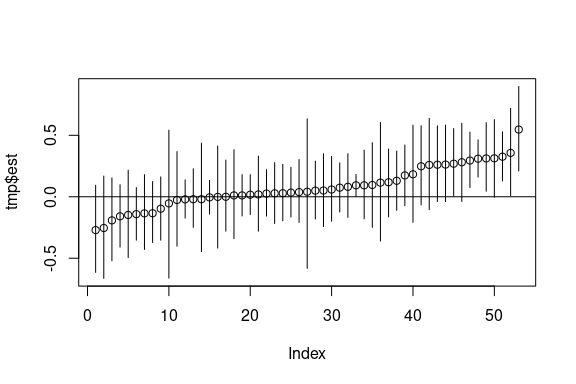


Figure 11: VAR with linear trend: Cross-regressive effect of interest on sad.

rm(tmp)

tmp <- results[results$parameters == "par3", ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:53,   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

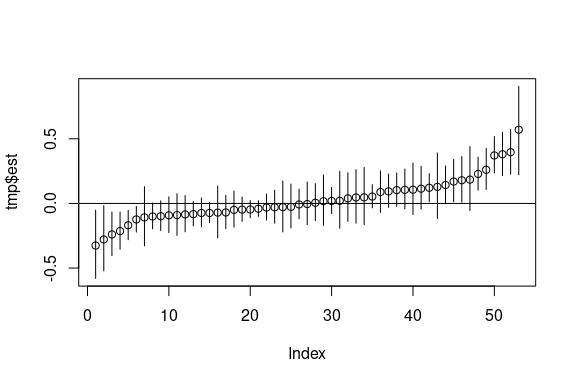


Figure 12: VAR with linear trend: Linear trend of interest.

rm(tmp)

tmp <- results[results$parameters == "par4", ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:53,   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

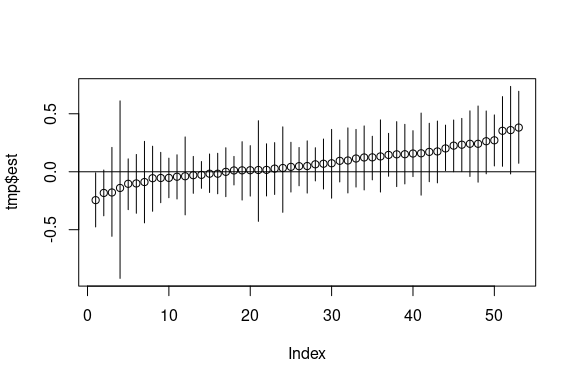


Figure 13: VAR with linear trend: Cross-regressive effect of sad on interest.

rm(tmp)

tmp <- results[results$parameters == "par5", ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:53,   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)



Figure 14: VAR with linear trend: Autoregressive effect of sad.

rm(tmp)

tmp <- results[results$parameters == "par6", ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:53,   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

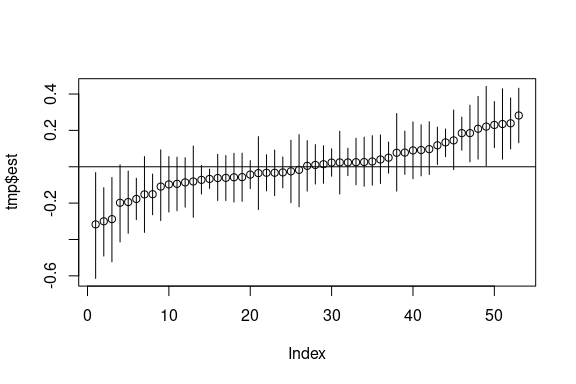


Figure 15: VAR with linear trend: Linear trend of sad.

rm(tmp)

We also ran the VAR analyses using a feature of DSEM that tries to approximate unequal spaced longitudinal observations to equal spaced by adding missing values. This is achieved by using the **TINTERVAL** option in Mplus. While this approach has been suggested to deliver similar results as continuous time models, finding the correct time interval can be difficult in empirical applications given the frequency of the beeps and whether the beeps prompted randomly throughout the day. Here, we used a **TINTERVAL** of 2.5 hours, which should result in adding three missing observations during the night. Note that some risks of using this feature is that multiple values might be on the same interval if the interval is not small enough. This is addressed automatically in Mplus, but if it happens too many times, Mplus warns that the approximation is not accurate. For more information about this, see Asparouhov et al. ([2018](#ref-Asparouhov2018)).

varfit <- list()  
  
if (!(dir.exists("Mplus"))) {dir.create("Mplus")}  
  
setwd("Mplus/")  
  
for (i in 1:length(ids)) {  
 tmp <- data\_var[data\_var$lifepak\_id == ids[i],   
 c("response\_time", "interest", "sad")]  
 tmp <- na.omit(tmp)  
 names(tmp) <- c("time", "int", "sad")  
   
 varfit[[i]] <- var2Mplus(y = c("int", "sad"), data = tmp,  
 variable\_options = list(timevar = "time", tinterval = 2.5),  
 analysis\_options = list(biterations.min = 10000,  
 chains = 3),  
 filename = paste0("id", ids[i], "tinterval.dat"),  
 runmodel = !file.exists(paste0("id", ids[i],  
 "tinterval.out"))  
 )  
}  
rm(i, tmp)  
setwd("..")

Here, we also extract the estimated parameters of interest with the following code:

parameters <- paste0("par", c(1:2, 4:5, 7:9))  
results <- lapply(varfit, function(x) {  
 out <- cbind(parameters,   
 x$parameters$unstandardized[c(1:5, 8, 9),   
 c(3, 6, 7)])  
 return(out)  
})  
  
results <- do.call(rbind, results)  
rm(parameters)

The estimated parameters and their credibility intervals are shown in the following figures:

tmp <- results[results$parameters == "par1", ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:53,   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

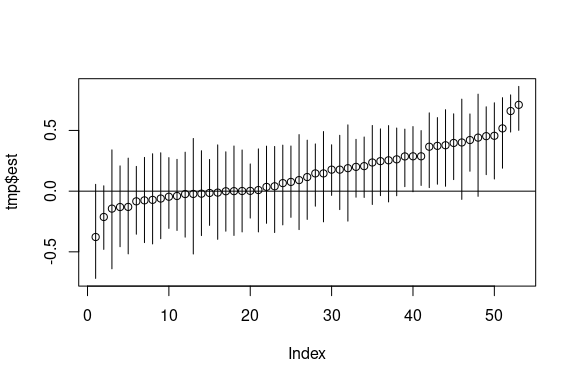


Figure 16: VAR with TINTERVAL: Autoregressive effect of interest.

rm(tmp)

tmp <- results[results$parameters == "par2", ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:53,   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

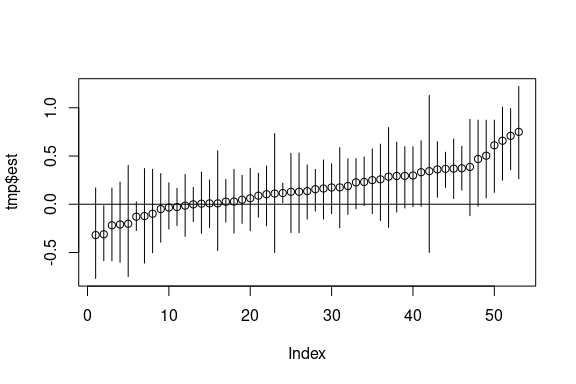


Figure 17: VAR with TINTERVAL: Cross-regressive effect of interest on sad.

rm(tmp)

tmp <- results[results$parameters == "par4", ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:53,   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

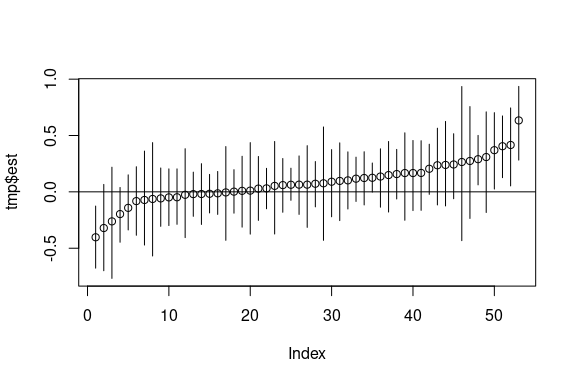


Figure 18: VAR with TINTERVAL: Cross-regressive effect of sad on interest.

rm(tmp)

tmp <- results[results$parameters == "par5", ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:53,   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

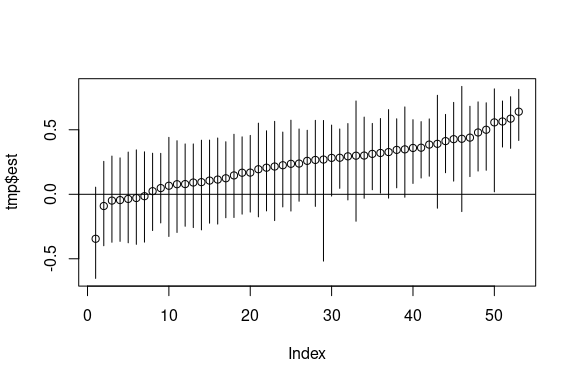


Figure 19: VAR with TINTERVAL: Autoregressive effect of sad.

rm(tmp)

## Multilevel VAR analysis

For the multilevel VAR analysis, we also fit the model adding time as a within covariate to account for linear trends and using the **TINTERVAL** option of Mplus. To fit the model using a linear trend, we used the following code:

if (!(dir.exists("Mplus"))) {dir.create("Mplus")}  
  
setwd("Mplus/")  
  
tmp <- data\_mlvar[, c("lifepak\_id", "response\_no\_night", "interest", "sad")]  
tmp <- na.omit(tmp)  
names(tmp) <- c("id", "time", "int", "sad")  
  
mlvarfit <- mlvar2Mplus(y = c("int", "sad"), id = "id", time = "time", data= tmp,  
 random.effects = list(lagged = TRUE, slopes = FALSE,  
 trend = TRUE, rvar = FALSE),  
 output\_options = list(standardized = TRUE),  
 filename = "mlvar\_trend.dat",  
 runmodel = !file.exists("mlvar\_trend.out"))  
  
rm(tmp)  
setwd("..")

In this case, we requested the standardized parameters within persons. This allows to extract the standardized effects for each individual.

results <- mlvarfit$parameters$wilevel.standardized$stdyx.standardized  
parameters <- c("S11|INT.ON", "S12|INT.ON", "S21|SAD.ON", "S22|SAD.ON")

Next, we plot the standardized lagged effects of each individual with their credibility intervals.

tmp <- results[results$paramHeader == parameters[1], ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:nrow(tmp),   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

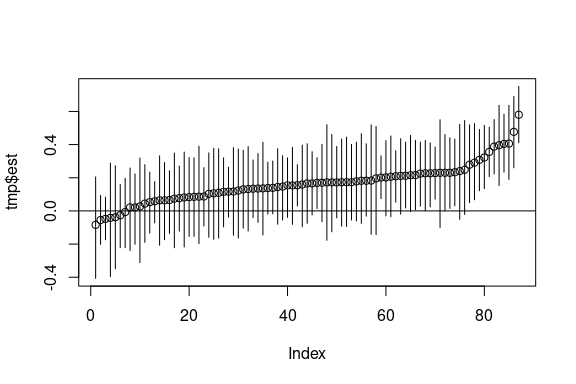


Figure 20: ML-VAR with linear trend: Autoregressive effect of interest.

rm(tmp)

tmp <- results[results$paramHeader == parameters[2], ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:nrow(tmp),   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

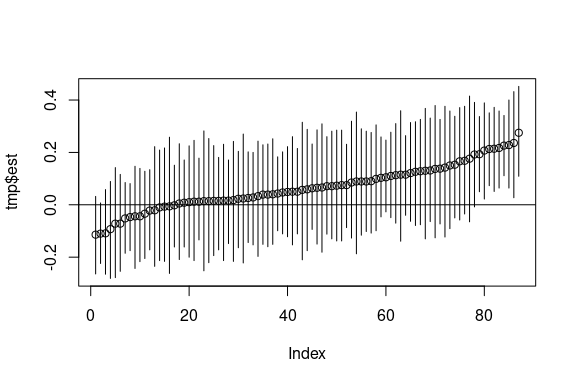


Figure 21: ML-VAR with linear trend: Cross-regressive effect of interest on sad.

rm(tmp)

tmp <- results[results$paramHeader == parameters[3], ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:nrow(tmp),   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

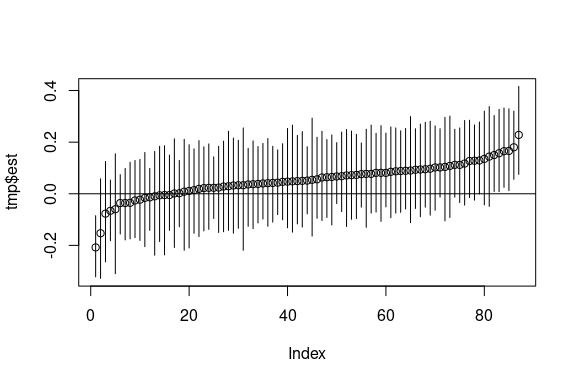


Figure 22: ML-VAR with linear trend: Cross-regressive effect of sad on interest.

rm(tmp)

tmp <- results[results$paramHeader == parameters[4], ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:nrow(tmp),   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

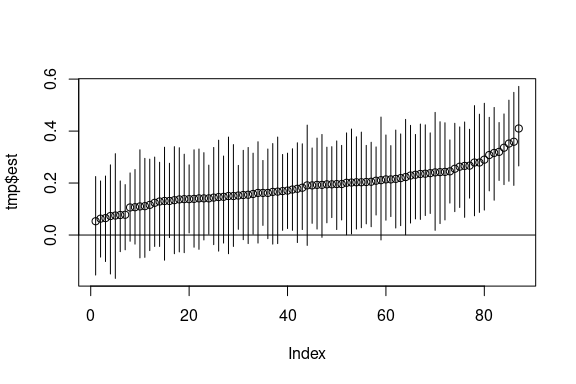


Figure 23: ML-VAR with linear trend: Autoregressive effect of sad.

rm(tmp)

We also fit the ML-VAR model using the **TINTERVAL** feature of Mplus. We indicated a time interval of 2.5 hours.

if (!(dir.exists("Mplus"))) {dir.create("Mplus")}  
  
setwd("Mplus/")  
  
tmp <- data\_mlvar[, c("lifepak\_id", "response\_time", "interest", "sad")]  
tmp <- na.omit(tmp)  
names(tmp) <- c("id", "time", "int", "sad")  
  
mlvarfit <- mlvar2Mplus(y = c("int", "sad"), id = "id", data= tmp,  
 random.effects = list(lagged = TRUE, slopes = FALSE,  
 trend = TRUE, rvar = FALSE),  
 variable\_options = list(timevar = "time", tinterval = 2.5),  
 analysis\_options = list(biterations.min = 10000,  
 chains = 3),  
 output\_options = list(standardized = TRUE),  
 filename = "mlvar\_tinterval.dat",  
 runmodel = !file.exists("mlvar\_tinterval.out"))  
  
rm(tmp)  
setwd("..")

Here, we also extracted the within person standardized parameters.

results <- mlvarfit$parameters$wilevel.standardized$stdyx.standardized  
parameters <- c("S11|INT.ON", "S12|INT.ON", "S21|SAD.ON", "S22|SAD.ON")

In the following Figures, we present the estimated standardized lagged effects with their corresponding credibility intervals.

tmp <- results[results$paramHeader == parameters[1], ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:nrow(tmp),   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

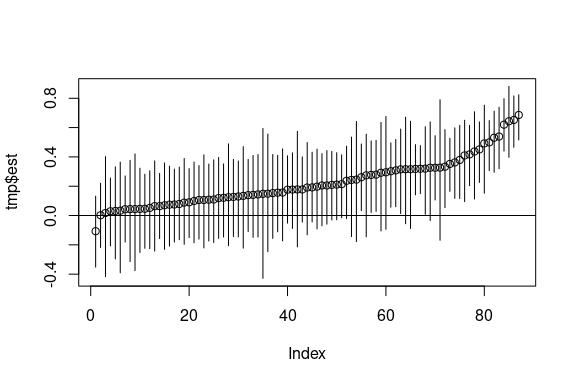


Figure 24: ML-VAR with linear trend: Autoregressive effect of interest.

rm(tmp)

tmp <- results[results$paramHeader == parameters[2], ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:nrow(tmp),   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

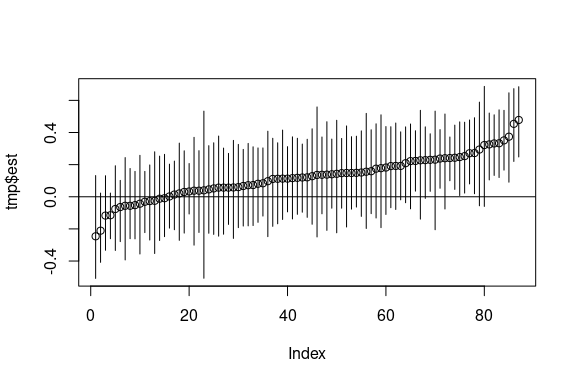


Figure 25: ML-VAR with linear trend: Cross-regressive effect of interest on sad.

rm(tmp)

tmp <- results[results$paramHeader == parameters[3], ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:nrow(tmp),   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

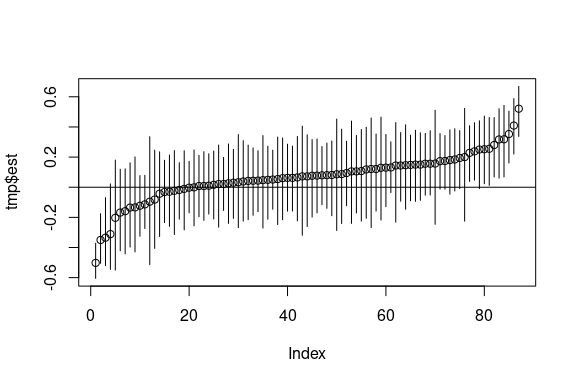


Figure 26: ML-VAR with linear trend: Cross-regressive effect of sad on interest.

rm(tmp)

tmp <- results[results$paramHeader == parameters[4], ]  
tmp <- tmp[order(tmp$est), ]  
plot(tmp$est, ylim = c(min(tmp$lower\_2.5ci), max(tmp$upper\_2.5ci)))  
segments(x0 = 1:nrow(tmp),   
 y0 = tmp$lower\_2.5ci,  
 y1 = tmp$upper\_2.5ci)  
abline(h = 0)

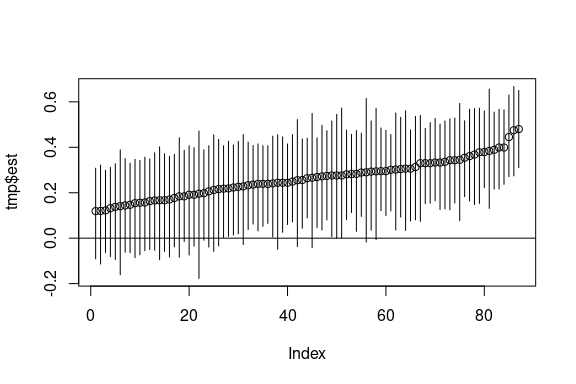


Figure 27: ML-VAR with linear trend: Autoregressive effect of sad.

rm(tmp)

# References

Asparouhov, T., Hamaker, E. L., & Muthén, B. (2018). Dynamic Structural Equation Models. *Structural Equation Modeling: A Multidisciplinary Journal*, *25*(3), 359–388. <https://doi.org/10.1080/10705511.2017.1406803>