

R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.
Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

```
library(ggplot2)
library(dplyr)
library(mlVAR)
library(qgraph)
library(bootnet)
library(reshape)
library(viridis)
library(lm.beta)
library(Rcpp)
library(lubridate)
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

```
#setwd("Insert Here")
```

```
figs <- "../figures/" # figure directory
datapath <- "../data/" # data directory

# -----
# ----- 2. Load data -----
# -----

# load data
preprocessed_data = read.csv("patient_3_Crosscheck.csv")
```

Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, : embedded nul(s) found in input

```
# Alpha to detrend:
alpha <- 0.05

# Data frame with empty values for fitted effects (all):
fitted_all <- expand_grid(
  day = seq(min(preprocessed_data$day),max(preprocessed_data$day))
)

# Data frame with empty values for day trends:
fitted_day <- data.frame(
  day = seq(min(preprocessed_data$day),max(preprocessed_data$day))
)

# Data frame to store p-values:
p_values <- data.frame(
  var = c("day", "beep")
)

# Also empty data frame list for test statistics:
testStatistics <- list()
coefficients <- list()
stdcoefficients <- list()
```

```
# Data frame to store detrended data:
data_detrended <- preprocessed_data

varlabs <- c('ema_CALM', 'ema_SOCIAL', 'ema_SLEEPING', 'ema_THINK', 'ema_HOPEFUL',
            'ema_VOICES', 'ema_SEEING_THINGS', 'ema_STRESSED', 'ema_HARM',
            'ema_DEPRESSED')
```

```
# Fix curves:
for (v in seq_along(varlabs)){
  # formula <- as.formula(paste0(varlabs[v], " ~ 1 + day + factor(beep)"))
  # contrasts can be applied only to factors with 2 or more levels
  formula <- as.formula(paste0(varlabs[v], " ~ 1 + day "))
  lmRes <- lm(formula, data = preprocessed_data)

  # Fixed effects:
  fixed <- coef(lmRes)

  # make zero if not significant at alpha:
  p_values[[varlabs[v]]] <- anova(lmRes)[["Pr(>F)"]][1:2]
  if (p_values[[varlabs[v]]][1] > alpha){
    fixed[2] <- 0
  }

  # Add to DFs:
  fitted_all[,varlabs[v]] <- fixed[1] + fixed[2] * fitted_all[["day"]]

  fitted_day[,varlabs[v]] <- fixed[1] + fixed[2] * fitted_day[["day"]]

# Detrend data:
data_detrended[,varlabs[v]] <- preprocessed_data[,varlabs[v]] - (fixed[1] + fixed[2] * preprocessed_data[["day"]
])

ids <- rownames(anova(lmRes))
testStatistics[[v]] <- cbind(data.frame(var = varlabs[v], effect = ids), anova(lmRes))

coefficients[[v]] <- data.frame(
  var = varlabs[v],
  type = names(coef(lmRes)),
  coef = coef(lmRes),
  std = coef(lm.beta(lmRes))
)
}
```

```
repetitions = 4
data_detrended <- data_detrended[rep(seq_len(nrow(data_detrended)), each = repetitions), ]
data_detrended[["id"]] <- rep(seq(repetitions), times=134)
```

```
# res <- mlVAR(data_detrended,
#             vars=varlabs,
#             idvar="id",
#             dayvar="day",
#             estimator="lm",
#             temporal = "unique")
```

```
# -----
# ----- 4. Here we estimate network models -----
# -----

# Estimate network using multilevel VAR model
res <- mlVAR(data_detrended,
             vars=varlabs,
             idvar="id",
             lags = 1,
             temporal = "orthogonal",
             contemporaneous = "orthogonal",
             nCores = 2)
```

'estimator' argument set to 'lmer'
Estimating temporal and between-subjects effects
Warning in lmer_mlVAR(PredModel, augData, idvar, verbose = verbose, contemporaneous = contemporaneous, : Zero SD found in mean of following variables: ema_CALM, ema_SOCIAL, ema_SLEEPING, ema_THINK, ema_HOPEFUL, ema_VOICES, ema_SEEING_THINGS, ema_STRESSED, ema_HARM, ema_DEPRESSED - Between-subject effects could not be estimated
Estimating contemporaneous effects
Computing random effects



```
# this is how we can save the object after the estimation; careful, over 100mb
#save(res, file=paste0(datapath, "network_orthogonal.Rdata"))

# you can later load it via
#res <- load(paste0(datapath, "network_orthogonal.Rdata"))
```

```
#summary(data_detrended)

#sum(is.na(data_detrended))

table(is.na(data_detrended))

FALSE
6432
```

```
# Plot :

names <- c("Calm","Social","Sleeping","Think","Hopeful","Voices","Seeing Things","Stressed","Harm", "Depressed")

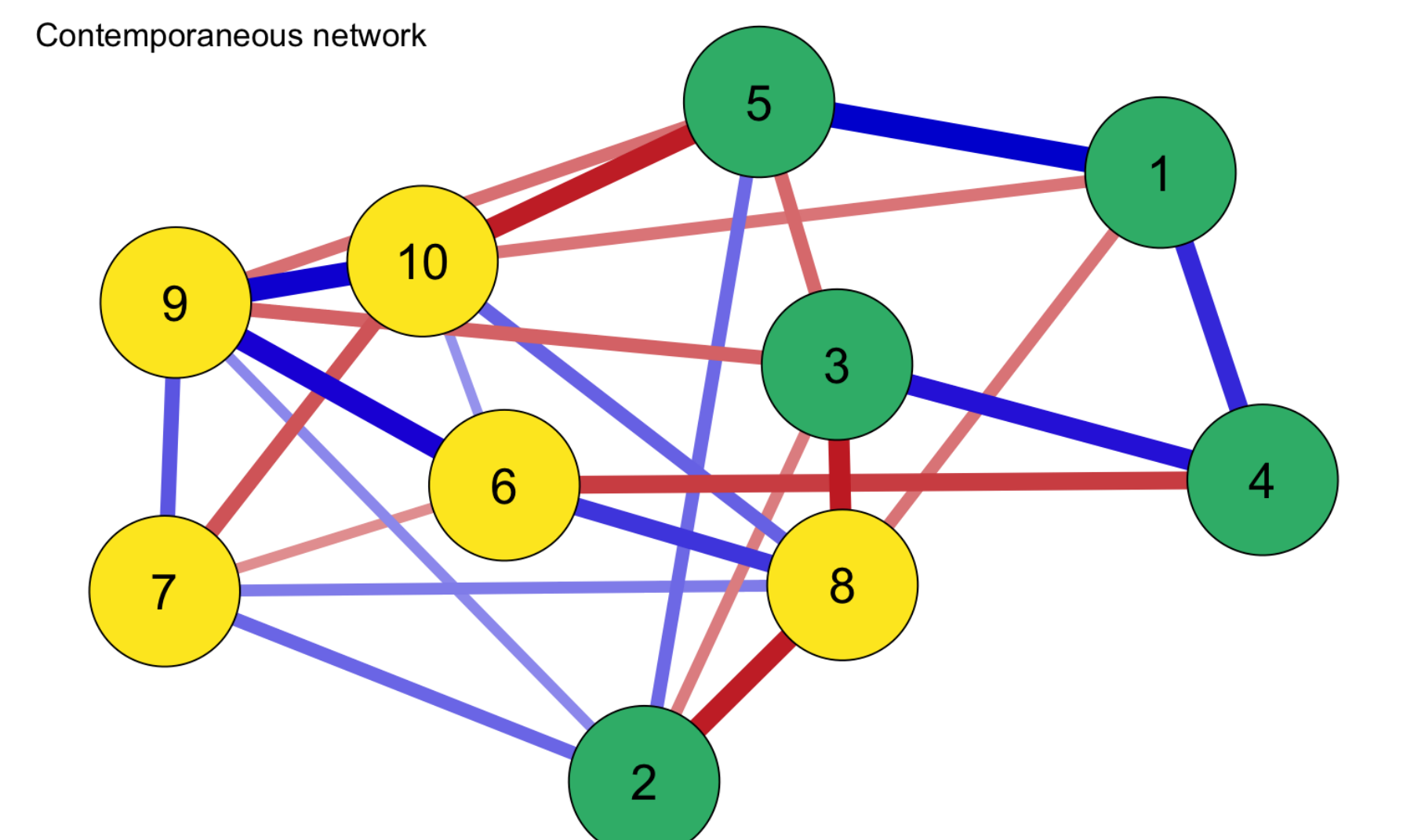
gr <- list('Positive'=c(1:5), 'Negative'=c(6:10))

# Get networks:
cont <- getNet(res, "contemporaneous", layout = "spring", nonsig = "hide", rule = "and")
#bet <- getNet(res, "between", nonsig = "hide", rule = "and")
temp <- getNet(res, "temporal", nonsig = "hide")

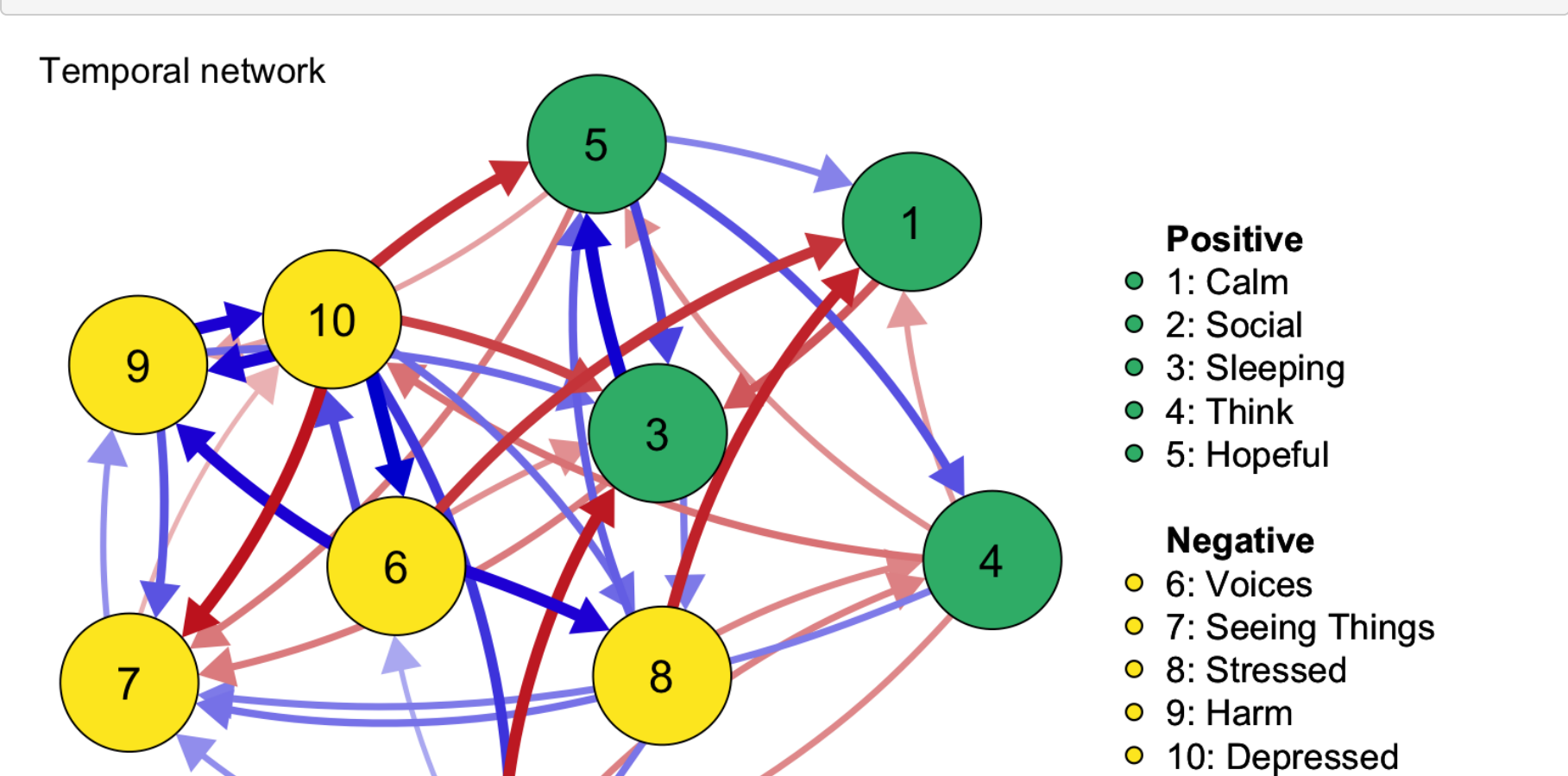
L <- averageLayout(cont, temp)

pdf(paste0("patient_3_Crosscheck.csv"), width=6, height=2.5)
layout(matrix(c(1,1,2,2,2), nc=5, byrow = TRUE)) # 40% vs 60% widths
n1 <- qgraph(cont, layout = L,
             title="Contemporaneous network", theme='colorblind', negDashed=FALSE,
             groups=gr, legend=FALSE, nodeNames = names, labels=c(1:10),
             vsize=12,color=viridis_pal()(4)[3:4])
n2 <- qgraph(temp, layout = L,
             title="Temporal network", theme='colorblind', negDashed=FALSE, diag=FALSE,
             groups=gr, legend.cex=0.5, legend=TRUE, nodeNames = names, labels=c(1:10),
             vsize=10,color=viridis_pal()(4)[3:4], asize=6, curve=0.75, curveAll=T)
```

```
plot(n1) # This gives you the contemporaneous network
```



```
plot(n2) # This gives you the temporal network
```



When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).
The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.