Network Analysis of Multivariate Data: Mental Health Code

This is an R Markdown Notebook explaining the Mental Health code in Network Analysis of Multivariate Data

First, let's import the necessary packages

```
library(ggplot2)
library(dplyr)
library(mlVAR)
library(qgraph)
library(bootnet)
library(reshape)
library(viridis)
# install.packages("lm.beta") # Added
library(lm.beta)
```

Now, let us set our working directory to where we have our files.

```
# setwd("Insert_Here")
```

Below, we load our data table. This will automatically be called Data2.

```
load("clean_network.RData")
initial_data <- Data2
```

Most of our initial variables are not very descriptive ("Q1", "Q2",…). Let's use the names function to rename our non-descriptive variables (i.e. the ones in vars).

```
# Alpha to detrend:
# What are the trends in the data. How do I detrend it. Are able to change things in the code
alpha <- 0.05

# Variables to investigate:
vars <- c("Q1","Q2","Q3","Q4","Q5","Q6","Q7","Q8","Q9","Q10","Q11","Q12","Q13","Q14","Q15","Q16","Q17","Q18")

# Labels:
varLabs <- c("Relax","Irritable","Worry","Nervous","Future","Anhedonia", "Tired","Hungry","Alone","Angry","Social
_offline","Social_online","Music", "Procrastinate","Outdoors","C19_occupied","C19_worry","Home")

names(initial_data)[names(initial_data) %in% vars] <- varLabs
```

We only want to keep the initial_data values that satisfy not being "Hungry", "Angry",… So we'll use the %>% operator from magrittr (from dplyr). Then get the varLabs modify varLabs to reflect this change. If debugging, only run this block once, hence why it was separated.

```
# Remove items: Negative values drop.
useful_data <- initial_data %>% dplyr::select(-Hungry,-Angry,-Music,-Procrastinate,-Tired,-Outdoors,-Home,-C19_oc
cupied,-C19_worry)
rm(initial_data) # No longer needed
varLabs <- varLabs[!varLabs %in% c("Hungry","Angry","Music","Procrastinate","Tired","Outdoors","Home","C19_occupi
ed","C19_worry")]
```

seq === python's range function. Overall, we are making data frames for our future use here. Beep is the time of data collection: The label 0 is for the time epoch 9:00 - 12:00. The label 1 is for the epoch 12:00 - 15:00… Since we have three labels, this means that we are limiting data collection from 9:00 - 21:00 Overall, we are obtaining all combinations of times for each day of data collection

```
# Data frame with empty values for fitted effects (all): This goes through the minimum of beep through max of bee
p, the min and max of day, and makes a grid of all combinations
fitted_all <- expand.grid(
  beep = seq.int(min(useful_data$beep),max(useful_data$beep)),
  day = seq.int(min(useful_data$day),max(useful_data$day))
)

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

# Data frame with empty values for beeps:
fitted_beep <- data.frame(
  beep = seq.int(min(useful_data$beep),max(useful_data$beep))
)

# 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()

# Make the beep variable factor in dataset:
useful_data$beepFactor <- factor(useful_data$beep, levels = 0:3, labels = c("09:00 - 12:00","12:00 - 15:00","15:0
0 - 18:00","18:00 - 21:00"))

# Same for fitted all and fitted beep
fitted_all$beepFactor <- factor(fitted_all$beep, levels = 0:3, labels = c("09:00 - 12:00","12:00 - 15:00","15:00
 - 18:00","18:00 - 21:00"))
fitted_beep$beepFactor <- factor(fitted_beep$beep, levels = 0:3, labels = c("09:00 - 12:00","12:00 - 15:00","15:0
0 - 18:00","18:00 - 21:00"))
```

Now, we want to get this in date format, and we also get the midpoint of these ranges to label this data with one overall time point in datetime format.

```
# Write as.Date
useful_data$date <- as.Date("2020-03-15") + useful_data$day
fitted_all$date <- as.Date("2020-03-15") + fitted_all$day
fitted_day$date <- as.Date("2020-03-15") + fitted_day$day

# Add the midpoints as time variable:
useful_data$midTime <- as.character(factor(useful_data$beep, levels = 0:3, labels = c("10:30","13:30","16:30","1
9:30")))

# posixct.
useful_data$midTime <- as.POSIXct(paste(useful_data$date,useful_data$midTime), format = "%Y-%m-%d %H:%M", tz = "E
urope/Amsterdam")

fitted_all$midTime <- as.character(factor(fitted_all$beep, levels = 0:3, labels = c("10:30","13:30","16:30","19:3
0")))
fitted_all$midTime <- as.POSIXct(paste(fitted_all$date,fitted_all$midTime), format = "%Y-%m-%d %H:%M", tz = "Euro
pe/Amsterdam")
```

We want to detrend the data. Mainly, some of the variables have trends (i.e. linear or other changes over time). Thus, we remove these trends before we estimate network structures. This is because if two variables have a similar trend solely because of their independent time dependence, they can just be scaled and we see a correlation that isn't truly there. Thus, we remove these time-dependent trends.

To go about this, we work with all 16 variables here that we collected. Then, we select those we want to estimate networks on. We make a linear model, get its coefficients, do an ANOVA to find significance with all these variables. Finally, we remove the trend using fitted_all.

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

# Fix curves: #seq along works with 1 var.
for (v in seq_along(varLabs)){
  formula <- as.formula(paste0(varLabs[v], " ~ 1 + day + factor(beep)")) # Get the formula from these lines
  lmRes <- lm(formula, data = useful_data)    }
```

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

# Fix curves: #seq along works with 1 var.
for (v in seq_along(varLabs)){
  formula <- as.formula(paste0(varLabs[v], " ~ 1 + day + factor(beep)")) # Get the formula from these lines
  lmRes <- lm(formula, data = useful_data)                                     # Fit a linear model to the data, a
nd the formula above. What is the formula

  # Fixed effects:
  fixed <- coef(lmRes) # Get the model coefficients

  # make zero if not significant at alpha:
  p_values[[varLabs[v]]] <- anova(lmRes)[["Pr(>F)"]][1:2] # Anova, difference of means
  if (p_values[[varLabs[v]]][1] > alpha){
    fixed[2] <- 0
  }
  if (p_values[[varLabs[v]]][2] > alpha){
    fixed[3:5] <- 0
  }

  # Add to DFs:

  # *Make a note*
  fitted_all[,varLabs[v]] <- fixed[1] + fixed[2] * fitted_all[["day"]]  +  fixed[3] * (fitted_all[["beep"]] == 1)
+
    fixed[4] * (fitted_all[["beep"]] == 2) + fixed[5] *  (fitted_all[["beep"]] == 3)

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

  fitted_beep[,varLabs[v]] <- fixed[1] + fixed[2] * median(fitted_day[["day"]]) +  fixed[3] * (fitted_beep[["bee
p"]] == 1)  +
    fixed[4] * (fitted_beep[["beep"]] == 2) + fixed[5] *  (fitted_beep[["beep"]] == 3)

  # Detrend data:
  data_detrended[,varLabs[v]] <- useful_data[,varLabs[v]] - (fixed[1] + fixed[2] * useful_data[["day"]]  +  fixed
[3] * (useful_data[["beep"]] == 1)  +
                                                           fixed[4] * (useful_data[["beep"]] == 2) + fixed[5] *
 (useful_data[["beep"]] == 3))

  ids <- rownames(anova(lmRes))
  # "day"          "factor(beep)" "Residuals"
  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)) # Standardized regression coefficients = lm.beta
  )
}
```

mlVAR computes estimates of the Multivariate Vector Autoregression model. This captures the relationship between features through time (temporal network), as well as the cross-sectional correlation of features.

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

# Estimate network using multilevel VAR model
res <- mlVAR(data_detrended,
             vars=varLabs,
             idvar="id",
             dayvar="day",
             beepvar="beep",
             lags = 1,
             temporal = "orthogonal",
             contemporaneous = "orthogonal",
             nCores = 8)
```

```
'estimator' argument set to 'lmer'
```

id <int>	day <int>	beep <dbl>
3	1	1
11	1	1
21	4	3
40	3	3
40	4	0
40	4	1
40	4	2
40	4	3
40	5	0
40	5	1

1-10 of 33 rows Previous **1** 2 3 4 Next

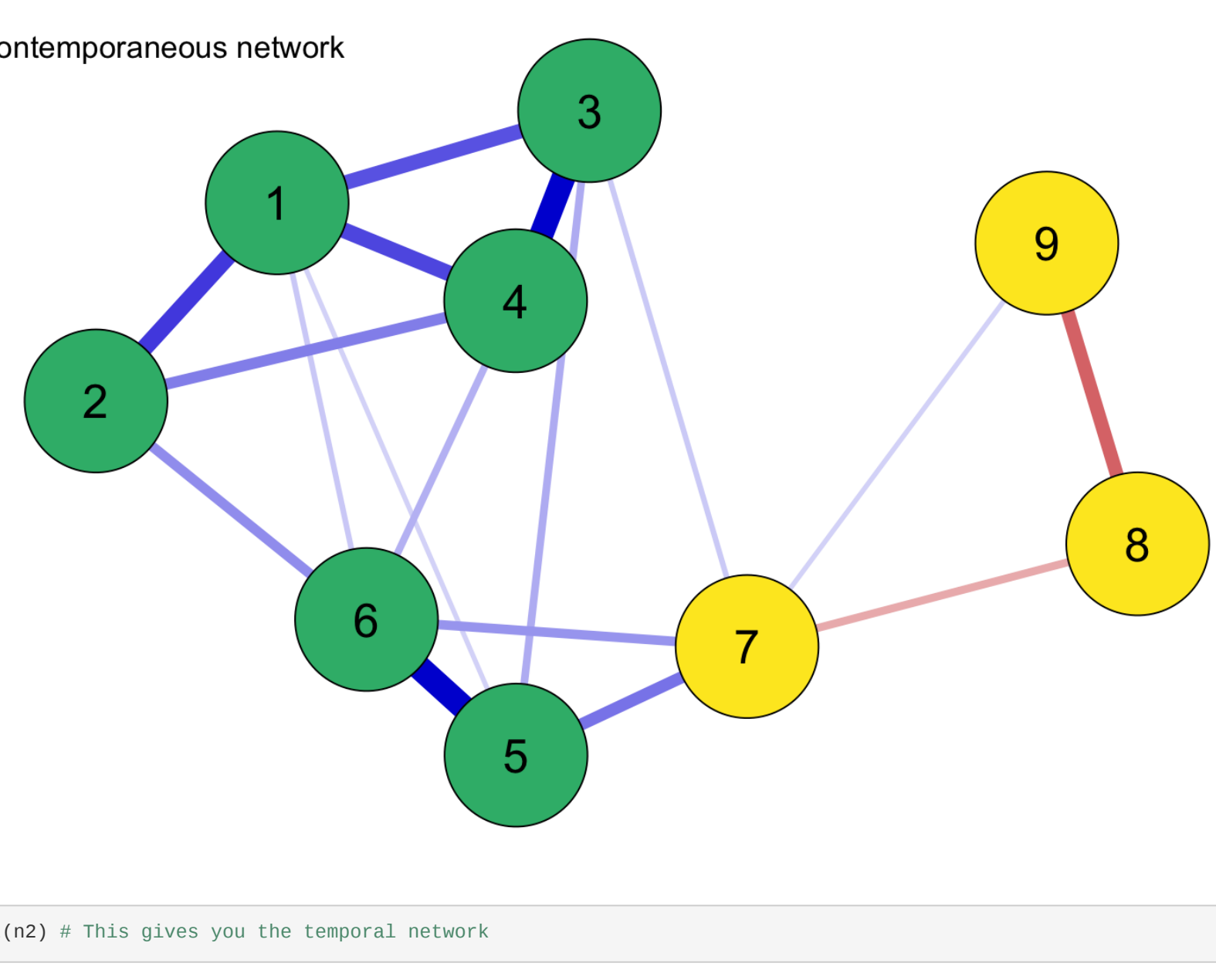
```
Warning in mlVAR(data_detrended, vars = varLabs, idvar = "id", dayvar = "day",  :
  Some beeps are recorded more than once! Results are likely unreliable.


Warning in mlVAR(data_detrended, vars = varLabs, idvar = "id", dayvar = "day",  :
  4 subjects detected with < 20 measurements. This is not recommended, as within-person centering with too few ob
servations per subject will lead to biased estimates (most notably: negative self-loops).
Estimating temporal and between-subjects effects
Estimating contemporaneous effects
Computing random effects
```

```
  |
  |
|   0%
  |
  |=
|   1%
  |
  |===
|   3%
  |
  |====
|   4%
  |
  |======
|   5%
  |
  |=======
|   6%
  |
  |=========
|   8%
  |
  |==========
|   9%
  |
  |============
|  10%
  |
  |=============
|  11%
  |
  |===============
|  13%
  |
  |================
|  14%
  |
  |==================
|  15%
  |
  |===================
|  16%
  |
  |=====================
|  18%
  |
  |======================
|  19%
  |
  |========================
|  20%
  |
  |=========================
|  22%
  |
  |===========================
|  23%
  |
  |============================
|  24%
  |
  |==============================
|  25%
  |
  |===============================
|  27%
  |
  |=================================
|  28%
  |
  |==================================
|  29%
  |
  |====================================
|  30%
  |
  |=====================================
|  32%
  |
  |=======================================
|  33%
  |
  |========================================
|  34%
  |
  |=========================================
|  35%
  |
  |===========================================
|  37%
  |
  |============================================
|  38%
  |
  |==============================================
|  39%
  |
  |===============================================
|  41%
  |
  |=================================================
|  42%
  |
  |==================================================
|  43%
  |
  |====================================================
|  44%
  |
  |=====================================================
|  46%
  |
  |=======================================================
|  47%
  |
  |========================================================
|  48%
  |
  |==========================================================
|  49%
  |
  |===========================================================
|  51%
  |
  |=============================================================
|  52%
  |
  |==============================================================
|  53%
  |
  |================================================================
|  54%
  |
  |=================================================================
|  56%
  |
  |===================================================================
|  57%
  |
  |====================================================================
|  58%
  |
  |======================================================================
|  59%
  |
  |=======================================================================
|  61%
  |
  |=========================================================================
|  62%
  |
  |==========================================================================
|  63%
  |
  |============================================================================
|  65%
  |
  |=============================================================================
|  66%
  |
  |==============================================================================
|  67%
  |
  |================================================================================
|  68%
  |
  |=================================================================================
|  70%
  |
  |===================================================================================
|  71%
  |
  |====================================================================================
|  72%
  |
  |======================================================================================
|  73%
  |
  |=======================================================================================
|  75%
  |
  |=========================================================================================
|  76%
  |
  |==========================================================================================
|  77%
  |
  |============================================================================================
|  78%
  |
  |=============================================================================================
|  80%
  |
  |===============================================================================================
|  81%
  |
  |================================================================================================
|  82%
  |
  |==================================================================================================
|  84%
  |
  |===================================================================================================
|  85%
  |
  |=====================================================================================================
|  86%
  |
  |======================================================================================================
|  87%
  |
  |========================================================================================================
|  89%
  |
  |=========================================================================================================
|  90%
  |
  |===========================================================================================================
|  91%
  |
  |============================================================================================================
|  92%
  |
  |==============================================================================================================
|  94%
  |
  |==============================================================================================================
=      |  95%
  |
  |==============================================================================================================
===    |  96%
  |
  |==============================================================================================================
====   |  97%
  |
  |==============================================================================================================
====== |  99%
  |
  |==============================================================================================================
=======| 100%
```

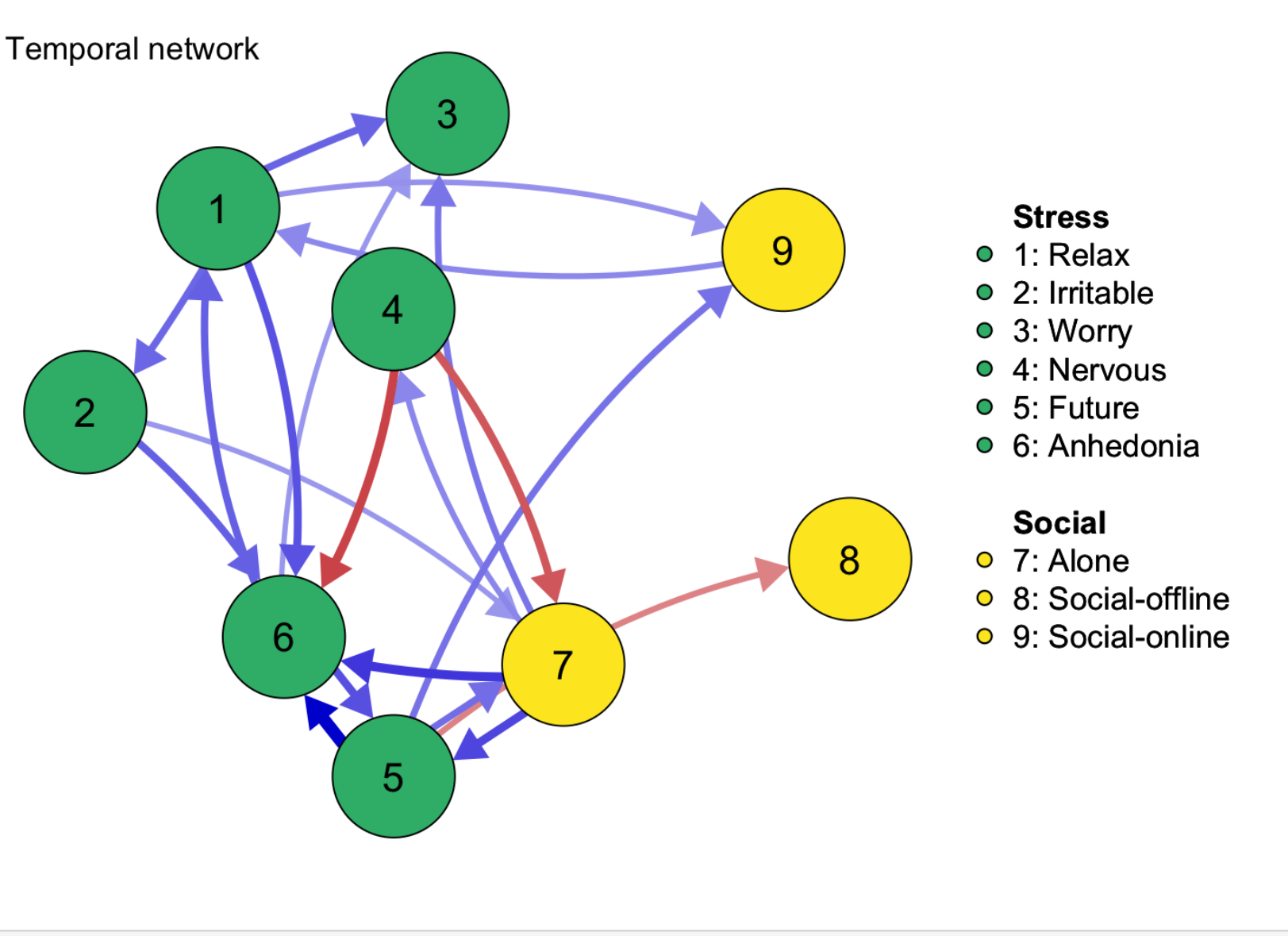
```
# Plot :
names <- c("Relax","Irritable","Worry","Nervous","Future", "Anhedonia","Alone","Social-offline", "Social-online")

gr <- list('Stress'=c(1:6), 'Social'=c(7:9))
# Is the algorithm conside

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

L <- averageLayout(cont, temp) # A layout matrix with the joint layout

pdf("figure.pdf", 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(1:9),
             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(1:9),
             vsize=10,color=viridis_pal()(4)[3:4], asize=6, curve=0.75, curveAll=T)
```

Now we plot the contemporaneous and temporal networks

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



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



```
#dev.off()
```