PAIRS ESM Networks

Emorie D Beck

March 25, 2017

Contents

1 Workspace **2**
1.1 Packages . . . 2
1.2 Local Functions . . . 2

2 Prepare Data **2**
2.1 Clean Data . . . 2
2.2 Screen Participants . . . 4
2.3 Missing Data Handling . . . 4
2.4 Within-Person Centering . . . 5
2.5 Response Order . . . 5

3 Population Level Analyses **6**
3.1 Table Results . . . 6
3.2 Plots . . . 7
3.3 Edge Stability . . . 13
3.4 Local Network Structure . . . 14

4 Idiographic Networks **21**
4.1 Clean Data and Prepare to Run Models . . . 21
4.2 Extract Results and Save Into Data Frames . . . 22
4.3 Stability . . . 24
4.4 Centrality . . . 31

1 Workspace

1.1 Packages

```
library(lavaan)
library(qgraph)
library(igraph)
library(GPArotation)
library(mlVAR)
library(graphicalVAR)
library(knitr)
library(gridExtra)
library(Rmisc)
library(psych)
library(rlist)
library(stargazer)
library(data.table)
library(tidyverse)
library(pander)
library(RColorBrewer)
```

1.2 Local Functions

```
meanSD_r2z2r <- function(x) {
  z <- fisherz(x)
  z[is.infinite(z)] <- NA
  x_bar <- mean(z, na.rm = T)
  x_sd <- sd(z, na.rm = T)
  r_bar <- fisherz2r(x_bar)
  r_sd <- fisherz2r(x_sd)
  return(c(r_bar, r_sd))
}
```

2 Prepare Data

The data include three waves of experience sampling method data from the Personality and Intimate Relationship Study. Data were previously cleaned to remove data points that did not meet inclusion criteria. ##Load Data

```
wave1_all <- tbl_df(read.csv("~/Box Sync/network/PAIRS/Wave 1/esm_w1_RENAMED.csv"))
wave4_all <- tbl_df(read.csv("~/Box Sync/network/PAIRS/Wave 4/esm_w4_RENAMED_all.csv"))
wave7_all <- tbl_df(read.csv("~/Box Sync/network/PAIRS/Wave 7/esm_w7_RENAMED_all.csv"))
```

2.1 Clean Data

Because the data sets include data that are not being used in this study, we extract the relevant columns (Subject ID, frequency, hour block, day of study, measurement point, and personality items) from the original data frames. Next, we rename the columns for later ease of use and visualization. Finally, because of the small sample size for waves 4 and 7, we merge those data sets.

```
#Getting necessary columns
#Keeping subject ID and all esm.BFI items
w1 <- dplyr::select(wave1_all, esm.IDnum.w1, esm.PRO01.w1, esm.PRO03.w1,
                    esm.PRO04.w1, esm.PRO05.w1, dplyr::matches("BFI"),
                    -dplyr::contains(".1."))
w4 <- dplyr::select(wave4_all, esm.IDnum.w4, esm.PRO01.w4, esm.PRO03.w4,
                    esm.PRO04.w4, esm.PRO05.w4, matches("BFI"))
w7 <- dplyr::select(wave7_all, esm.IDnum.w7, esm.PRO01.w7, esm.PRO03.w7,
                    esm.PRO04.w7, esm.PRO05.w7,  matches("BFI"))
```

```r
w7 <- w7[,!(colnames(w7) %in% c("esm.BFI20.w7", "esm.BFI12.w7"))]

# column names for w1
varnames <- c("SID", "freq", "hourBlock", "day", "beepvar",
              "A_rude", "E_quiet", "C_lazy",
              "N_relaxed", "N_depressed", "E_outgoing",
              "A_kind", "C_reliable", "N_worried")

# column names for w4 and w7
varnames_w47 <- c("SID", "freq", "hourBlock", "day", "beepvar",
              "E_outgoing","E_quiet",
              "C_lazy","C_reliable",
              "N_worried","N_relaxed",
              "N_depressed", "A_rude",
              "A_kind")

# short column names (for plots)
varnames2 <- c("rude", "quiet", "lazy",
              "relaxed", "depressed", "outgoing",
              "kind", "reliable", "worried")


# rename columns
colnames(w1) <- varnames
colnames(w4) <- varnames_w47
colnames(w7) <- varnames_w47

# change subject IDs to factor
w1$SID <- factor(w1$SID)
w4$SID <- factor(w4$SID)
w7$SID <- factor(w7$SID)


# reorder w4 and w7 columns to match w1
w4 <- w4[,c(varnames,setdiff(names(w4), varnames))]
w7 <- w7[,c(varnames,setdiff(names(w7), varnames))]

# create wave variable before combining data sets.
w4$Wave <- "4"
w7$Wave <- "7"
# merge wave 4 and 7 data sets
w47 <- merge(w4, w7, all = T)
```

Variable	New Name	Description
esm.IDnum.w1	SID	numeric variable; identification number
esm.BFI37.w1	A_rude	agreeablness, negative; "During the last hour, how rude were you?" Likert scale from 1 to 5; 1 = Not a lot, 3 = Somewhat, 5 = Very
esm.BFI21.w1	E_quiet	extraversion, negative; "During the last hour, how quiet were you?" Likert scale from 1 to 5; 1 = Not a lot, 3 = Somewhat, 5 = Very
esm.BFI23.w1	C_lazy	conscientiousness, negative; "During the last hour, how lazy were you?" Likert scale from 1 to 5; 1 = Not a lot, 3 = Somewhat, 5 = Very
esm.BFI09.w1	N_relaxed	neuroticism, positive; "During the last hour, how relaxed were you?" Likert scale from 1 to 5; 1 = Not a lot, 3 = Somewhat, 5 = Very
esm.BFI04.w1	N_depressed	neuroticism, positive; "During the last hour, did you feel 'depressed, blue'?" Likert scale from 1 to 5; 1 = Not a lot, 3 = Somewhat, 5 = Very

Variable	New Name	Description
esm.BFI36.w1	E_outgoing	extraversion, positive; "During the last hour, how 'outgoing, sociable' were you?" Likert scale from 1 to 5; 1 = Not a lot, 3 = Somewhat, 5 = Very
esm.BFI32.w1	A_kind	agreeablness, positive; "During the last hour, how 'considerate, kind' were you?" Likert scale from 1 to 5; 1 = Not a lot, 3 = Somewhat, 5 = Very
esm.BFI13.w1	C_reliable	conscientiousness, positive; "During the last hour, how reliable were you?" Likert scale from 1 to 5; 1 = Not a lot, 3 = Somewhat, 5 = Very
esm.BFI19.w1	N_worried	neuroticism, positive; "During the last hour, how worried were you?" Likert scale from 1 to 5; 1 = Not a lot, 3 = Somewhat, 5 = Very

2.2 Screen Participants

To be able to construct individual networks for participants, we ideally need approximately 50 measurement points. However, for current purposes, we will keep all participants who have at least 20 responses, lest we eliminate a large portion of our subjects.

```
# get frequency count of subject's responses
problem_w1 <- plyr::count(w1$SID)
problem_w47 <- plyr::count(w47$SID)

# extract subject IDs of those with < 10 measurement points
problem_w1  <- as.character(problem_w1$x[which(problem_w1$freq < 10)])
problem_w47 <- as.character(problem_w47$x[which(problem_w47$freq < 10)])

# save excluded subjects to a separate data frame
excluded <- w1[which(w1$SID %in% problem_w1),]
excluded_w47 <- w47[which(w47$SID %in% problem_w47),]

# remove excluded subjects
w1  <- w1[!(w1$SID %in% problem_w1),]
w47 <- w47[!(w47$SID %in% problem_w47),]
```

2.3 Missing Data Handling

Participants in the study only answered Agreeableness items if they indicated they were interacting with another person during the hour block previous to responding. To retain those measurement points for use in models later, we fill in gaps using within-person means of Agreeabless items.

```
# replace missing values with person-level means
for (i in unique(w1$SID)){
  mean_A_rude <- mean(w1$A_rude[w1$SID == i], na.rm = T)
  w1$A_rude[is.na(w1$A_rude) & w1$SID == i] <- mean_A_rude
  mean_A_kind <- mean(w1$A_kind[w1$SID == i], na.rm = T)
  w1$A_kind[is.na(w1$A_kind) & w1$SID == i] <- mean_A_kind
}

for (i in unique(w47$SID)){
  mean_A_rude <- mean(w47$A_rude[w47$SID == i], na.rm = T)
  w47$A_rude[is.na(w47$A_rude) & w47$SID == i] <- mean_A_rude
  mean_A_kind <- mean(w47$A_kind[w47$SID == i], na.rm = T)
  w47$A_kind[is.na(w47$A_kind) & w47$SID == i] <- mean_A_kind
}
```

2.4 Within-Person Centering

Below, we within-person center both data sets to aid in interpretation.

```
# retain cases where all personality data are retained
w1_com <- w1[complete.cases(w1[,c(6:14)]),]
w47_com <- w47[complete.cases(w47[,c(6:14)]),]

# for waves 4 and 7, create a variable that combines wave and day of study
w47_com$waveDay <- paste(w47_com$Wave, w47_com$day, sep = ".")

# reorder data sets by SID, day, and block
w1_com  <- w1_com[order(w1_com$SID, w1_com$day, w1_com$hourBlock),]
w47_com <- w47_com[order(w47_com$SID, w47_com$waveDay, w47_com$hourBlock),]
```

2.5 Response Order

Because of the way the `mlVAR()` handles measurement points, we create a new variable that numbers participants included responses, which we will use for the final model.

```
# create variable that sequences all responses
w1_com <- w1_com %>%
  group_by(SID) %>%
  arrange(day, hourBlock) %>%
  mutate(beepvar3 = seq(1, n(), 1))

w47_com <- w47_com %>%
  group_by(SID) %>%
  arrange(waveDay, hourBlock) %>%
  mutate(beepvar3 = seq(1, n(), 1))

# Make numeric subject IDs for each df because mlVAR won't run for factors #
w1_com$SID2 <- as.numeric(w1_com$SID)
w47_com$SID2 <- as.numeric(w47_com$SID)

# calculate individual for each variable
w1_com <- w1_com %>%
  group_by(SID) %>%
  mutate_each_(funs(comp = mean), vars = colnames(w1_com)[6:14]) %>%
  ungroup()
w47_com <- w47_com %>%
  group_by(SID) %>%
  mutate_each_(funs(comp = mean), vars = colnames(w47_com)[6:14]) %>%
  ungroup()

# calculate overaall means for all variables
means_w1 <- w1_com %>%
  dplyr::select(17:25) %>%
  summarise_each(funs = funs(mean = mean))
means_w47 <- w47_com %>%
  dplyr::select(19:27) %>%
  summarise_each(funs = funs(mean = mean))



# save final data frames to files
write.csv(w1_com, "~/Box Sync/network/PAIRS/Wave 1/esm_w1_networks.csv", row.names = F)
write.csv(w47_com, "~/Box Sync/network/PAIRS/Wave 4 + 7/esm_w47_networks.csv", row.names = F)
```

3 Population Level Analyses

Bringmann et al. (2013) developed a technique for assessing ESM data using a network approach. This approach utilizes a series of univariate multilvel vector autoregressive models (mlVAR) in which all items are entered simultaneously as predictors and individually as outcomes. Below we run the models for the nine personality items at each wave. The `mlVAR()` function automatically within-centers data, so we will enter the raw data into the models. Because of the number of observations we have for each usbject we use a lag 1 factorization. Because we have more than 6 predictors, we use orthogonal estimation of temporal and contemporaneous effects.
##Univariate Multilevel Autoregressive Models ###Wave 1

```
# raw data #
# not affected by using only complete cases of the data #
fit1_w1 <-
  mlVAR_test(w1_com,
        vars = colnames(w1_com)[6:14], #4:18
        idvar = "SID2",
        lags = 1,
        #dayvar = "day",
        beepvar = "beepvar3",
        temporal = "orthogonal",
        contemporaneous = "orthogonal",
        verbose = TRUE,
        scale = FALSE)
```

3.0.1 Waves 4 + 7

```
# raw data #
fit1_w47 <-
  mlVAR_test(w47_com,
        vars = colnames(w47_com)[6:14], #4:16
        idvar = "SID2",
        lags = 1,
        #dayvar = "day",
        beepvar = "beepvar3",
        temporal = "orthogonal",
        contemporaneous = "orthogonal",
        verbose = TRUE,
        scale = FALSE)
```

3.1 Table Results

mlVAR involves running a series of multilevel models, we include all of the results for each wave in a table. ###Wave 1

```
# get modified data from mlVAR output and output all mlm's to table
augData <- fit1_w1$data
sjt.lmer(fit1_w1$output$A_rude, fit1_w1$output$E_quiet, fit1_w1$output$C_lazy,
         fit1_w1$output$N_relaxed, fit1_w1$output$N_depressed, fit1_w1$output$E_outgoing,
         fit1_w1$output$A_kind, fit1_w1$output$C_reliable, fit1_w1$output$N_worried,
         file = "~/Box Sync/network/PAIRS/Wave 1/mlVAR_raw.html",
         p.numeric = FALSE,
         show.ci = FALSE,
         sep.column = FALSE,
         pred.labels = c(paste(varnames2, "within"), paste(varnames2, "between")))


library(stargazer)
library(XML)

table <- readHTMLTable("~/Box Sync/network/PAIRS/Wave 1/mlVAR_raw.html", header=T)
table <- as.data.frame(table)
table <- table[-1, ]
colnames(table) <- c("", rep(c("B"), times = 9))
```

```r
table[,1] <- as.character(table[,1])
table[c(22:27),1] <- c("$\\sigma^2$", "$\\pi_{00, SID2}$", "$\\N_{SID2}$",
                       "$\\ICC_{SID2}$", "Observations",
                       paste("$\\R^{2}", "\\Omega_{0}^2$"))
stargazer(table, summary=F, rownames=F, float.env = "sidewaystable",
          font.size = "footnotesize")
```

3.1.1 Wave 4 + 7

```r
# get modified data from mlVAR output and output all mlm's to table
augData <- fit1_w47$data
sjt.lmer(fit1_w47$output$A_rude, fit1_w47$output$E_quiet, fit1_w47$output$C_lazy,
         fit1_w47$output$N_relaxed, fit1_w47$output$N_depressed, fit1_w47$output$E_outgoing,
         fit1_w47$output$A_kind, fit1_w47$output$C_reliable, fit1_w47$output$N_worried,
         file = "~/Box Sync/network/PAIRS/Wave 4 + 7/mlVAR_raw.html",
         p.numeric = FALSE,
         sep.column = FALSE,
         show.ci = FALSE,
         pred.labels = c(paste(varnames2, "within"), paste(varnames2, "between")))


library(stargazer)
library(XML)

table <- readHTMLTable("~/Box Sync/network/PAIRS/Wave 4 + 7/mlVAR_raw.html", header=T)
table <- as.matrix(as.data.frame(table))
table <- rbind(c("", rep(c("B"), times = 9)), table)
table <- as.data.frame(table)
colnames(table) <- c("", varnames2)
table[,1] <- as.character(table[,1])
table[c(23:28),1] <- c("$\\sigma^2$", "$\\pi_{00, SID2}$", "$\\N_{SID2}$",
                       "$\\ICC_{SID2}$", "Observations",
                       paste("$\\R^{2}", "\\Omega_{0}^2$"))
stargazer(table, summary=F, rownames=F, float.env = "sidewaystable",
          font.size = "footnotesize")
```

Save summaries of the models

```r
#model summary
sum_fit1_w1             <- summary(fit1_w1)
sum_fit1_w47            <- summary(fit1_w47)
```

3.2 Plots

The graphs below show the raw, directed network the estimated, directed network using a univariate multilevel vector autoregressive models as (1) temporal and (2) contemporaneous netowrks

```r
# load centrality data for plotting
load("~/Box Sync/network/PAIRS/Wave 1/centrality_w1_pers.RData")
load("~/Box Sync/network/PAIRS/Wave 4/centrality_w47_pers.RData")

#load communities data for plotting
load("~/Box Sync/network/PAIRS/manuscript/R/communities.RData")

border_colors <- brewer.pal(11,"PRGn")[1:3]

######## temporal ########
# calculate quantiles of inStrength
temporal_cutoffs_w1 <-
  quantile(temporal_centrality_w1_raw$node.centrality$InStrength,
           probs = c(1/3,2/3))
temporal_cutoffs_w47 <-
  quantile(temporal_centrality_w47_raw$node.centrality$InStrength,
           probs = c(1/3,2/3))
```

Table 2:

	rude	**quiet**	**lazy**	**relaxed**	**depressed**	**outgoing**	**kind**	**reliable**	**worried**
	b	b	b	b	b	b	b	b	b
Fixed Parts									
(Intercept)	1.65 ***	3.61 ***	1.49 **	2.73 ***	0.12	2.90 ***	-0.59	2.84 ***	2.94 ***
rude within	0.03 *	0.01	0.01	-0.05 *	0.02	-0.02	-0.01	-0.02	0.03
quiet within	0.01	0.00	0.00	0.01	-0.01	-0.00	-0.00	0.01	0.00
lazy within	-0.00	0.03 **	0.06 ***	-0.01	0.02 *	-0.03 **	0.01	-0.02 *	0.01
relaxed within	0.00	-0.04 **	0.03 *	0.04 **	-0.03 **	0.04 *	0.01	0.00	-0.05 ***
depressed within	-0.01	0.01	0.05 **	-0.01	0.11 ***	0.01	0.01	-0.03 *	0.03
outgoing within	-0.00	-0.01	0.02	0.00	0.00	0.02	0.01	0.01	-0.01
kind within	0.00	0.01	-0.02	-0.03	0.01	-0.03	0.03 *	0.01	0.05 ***
reliable within	-0.01	-0.01	0.01	-0.01	-0.00	-0.01	0.02 *	0.03 *	-0.01
worried within	0.00	0.03	-0.05 ***	-0.11 ***	0.05 ***	-0.05 **	-0.02 *	0.00	0.14 ***
rude between	-0.07		0.25 ***	0.15 *	0.02	-0.73 ***	0.27 **	0.04	0.14 *
quiet between	0.12 **	0.15 ***		0.17 ***	0.23 ***	0.04	-0.06	-0.39 ***	0.02
lazy between	-0.04	0.15 **	0.27 ***		-0.19 **	0.24 ***	0.13	0.14 *	-0.67 ***
relaxed between	0.19 ***	0.00	0.28 ***	-0.13 **		-0.05	0.20 **	0.03	0.38 ***
depressed between	0.14 **	-0.69 ***	0.04	0.24 ***	-0.07		0.46 ***	0.14 *	0.14 *
outgoing between	-0.11 **	0.13 ***	-0.06	0.06	0.11 **	0.21 ***		0.30 ***	0.01
kind between	-0.14 ***	0.02	-0.39 ***	0.09 *	0.03	0.09 *	0.43 ***		0.01
reliable between	0.04	0.10 *	0.02	-0.50 ***	0.38 ***	0.08	0.03	0.03	
worried between		-0.07	0.20 **	-0.04	0.27 ***	0.12 *	-0.26 **	-0.22 **	0.04
Random Parts									
σ^2	0.289	1.523	1.217	0.932	0.530	1.407	0.488	0.644	0.877
$\pi_{00,SID2}$	0.085	0.036	0.110	0.067	0.113	0.045	0.201	0.133	0.104
N_{SID2}	349	349	349	349	349	349	349	349	349
ICC_{SID2}	0.213	0.023	0.081	0.064	0.167	0.030	0.283	0.165	0.101
Observations	10775	10775	10775	10775	10775	10775	10775	10775	10775
R^2/Ω_0^2	.378 / .373	.134 / .128	.233 / .225	.277 / .270	.406 / .403	.177 / .172	.483 / .481	.372 / .369	.343 / .340
Notes	*p<.05 **p<.01 ***p<.001								

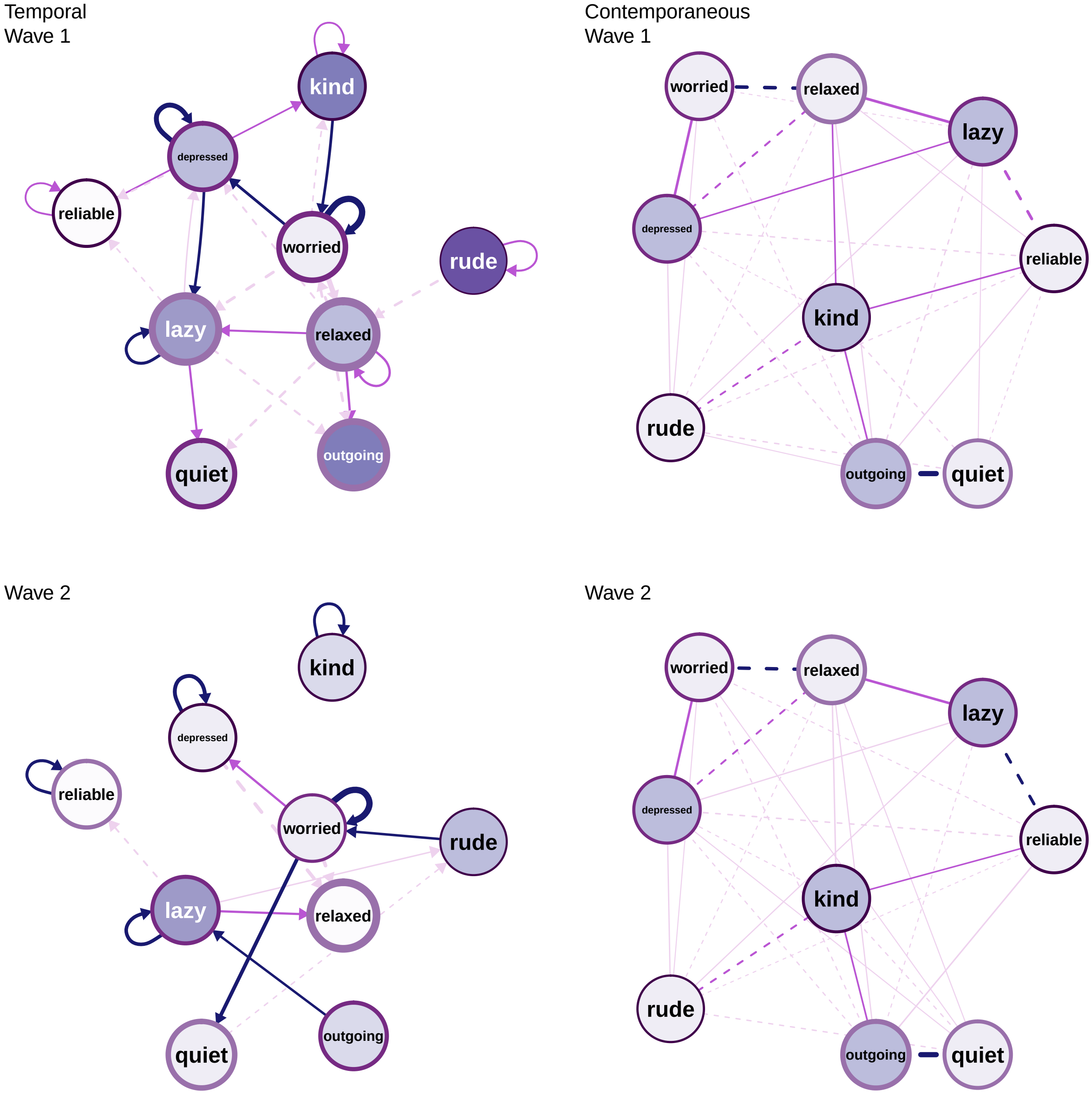
Table 3:

	rude	**quiet**	**lazy**	**relaxed**	**depressed**	**outgoing**	**kind**	**reliable**	**worried**
	b	*b*	*b*	*b*	*b*	*b*	*b*	*b*	*b*
Fixed Parts									
Fixed Parts									
(Intercept)	0.69	3.27 ***	2.33 ***	2.82 ***	0.37	2.53 ***	0.57	1.94 ***	3.39 ***
rude within	0.02	0.00	0.01	0.01	0.01	-0.05	-0.00	0.03	0.05 *
quiet within	-0.02 *	0.02	0.00	-0.01	0.00	0.00	-0.00	-0.00	0.01
lazy within	0.02 *	-0.00	0.08 ***	0.04 **	0.01	0.01	0.01	-0.03 *	-0.00
relaxed within	-0.00	0.04	0.01	0.02	-0.02	0.00	0.00	0.00	0.01
depressed within	-0.00	-0.01	0.01	-0.08 ***	0.09 ***	-0.01	0.00	-0.02	0.02
outgoing within	-0.00	0.00	0.04 *	0.00	0.01	0.02	-0.00	-0.02	0.01
kind within	0.01	-0.01	0.03	0.01	0.01	0.01	0.06 ***	0.02	-0.00
reliable within	-0.01	-0.01	0.01	0.00	-0.00	0.03	0.00	0.06 ***	0.00
worried within	0.01	0.08 ***	-0.01	-0.08 ***	0.04 **	-0.03	-0.02	-0.02	0.16 ***
rude between	0.06		0.25 *	0.07	0.19 *	-0.73 ***	-0.00	0.37 ***	-0.12
quiet between	0.18 **	0.15 *		0.10	0.10	0.09	0.17	-0.48 ***	-0.00
lazy between	-0.01	0.07	0.10		-0.19 **	0.10	0.17	0.09	-0.44 ***
relaxed between	0.19 **	0.13 *	0.13	-0.22 *		-0.03	0.16	-0.05	0.43 ***
depressed between	0.29 ***	-0.66 ***	0.11	0.12	-0.02		0.26	0.43 ***	-0.09
outgoing between	-0.14 **	-0.00	0.08	0.09	0.06	0.10 *		0.16 **	0.03
kind between	-0.15 *	0.28 ***	-0.61 ***	0.12	-0.03	0.36 ***	0.39 **		0.00
reliable between	0.08	-0.02	-0.00	-0.31 ***	0.26 ***	-0.02	0.04	0.00	
worried between		0.03	0.26 **	0.02	0.24 **	0.29 ***	-0.42 **	-0.19 *	0.17
Random Parts									
σ^2	0.290	1.421	1.144	0.851	0.440	1.340	0.490	0.663	0.786
$\pi_{00,SID2}$	0.089	0.041	0.114	0.114	0.109	0.053	0.279	0.105	0.176
N_{SID2}	157	157	157	157	157	157	157	157	157
ICC_{SID2}	0.227	0.027	0.089	0.115	0.184	0.038	0.351	0.133	0.175
Observations	6434	6434	6434	6434	6434	6434	6434	6434	6434
R^2/Ω_0^2	.383 / .382	.138 / .132	.240 / .237	.247 / .242	.410 / .408	.176 / .173	.494 / .493	.365 / .363	.351 / .347
Notes	*p<.05 **p<.01 ***p<.001								

```r
# assign colors to variables based on quantile
temporal_centrality_colors_w1 <-
  ifelse(temporal_centrality_w1_raw$node.centrality$InStrength <= temporal_cutoffs_w1[1],
         border_colors[1],
  ifelse(temporal_centrality_w1_raw$node.centrality$InStrength <= temporal_cutoffs_w1[2],
         border_colors[2],border_colors[3]))
temporal_centrality_colors_w47 <-
  ifelse(temporal_centrality_w47_raw$node.centrality$InStrength <= temporal_cutoffs_w47[1],
         border_colors[1],
  ifelse(temporal_centrality_w47_raw$node.centrality$InStrength <= temporal_cutoffs_w47[2],
         border_colors[2], border_colors[3]))


######## contemporaneous ########
# calculate quantiles of Strength
contemporaneous_cutoffs_w1 <-
  quantile(contemporaneous_centrality_w1_raw$node.centrality$Strength,
           probs = c(1/3,2/3))
contemporaneous_cutoffs_w47 <-
  quantile(contemporaneous_centrality_w47_raw$node.centrality$Strength,
           probs = c(1/3,2/3))

# assign colors to variables based on quantile
contemporaneous_centrality_colors_w1 <-
  ifelse(contemporaneous_centrality_w1_raw$node.centrality$Strength <=
           contemporaneous_cutoffs_w1[1], border_colors[1],
  ifelse(contemporaneous_centrality_w1_raw$node.centrality$Strength <=
           contemporaneous_cutoffs_w1[2], border_colors[2], border_colors[3]))
contemporaneous_centrality_colors_w47 <-
  ifelse(contemporaneous_centrality_w47_raw$node.centrality$Strength <=
           contemporaneous_cutoffs_w47[1], border_colors[1],
  ifelse(contemporaneous_centrality_w47_raw$node.centrality$Strength <=
           contemporaneous_cutoffs_w47[2], border_colors[2], border_colors[3]))


# run but don't generate temporal plot
  plot_w1 <- plot(fit1_w1,
     "temporal",
     layout = "spring",
     labels = varnames2, # can be a vector with node labels
     groups = sgc_w1_temporal,
     nonsig = 'hide',
     curve = -1,
     legend = FALSE,
     details = FALSE,
     mar = c(5,5,5,5), # controls margins
     border.color = temporal_centrality_colors_w1, # controls the border colors
     border.width = temporal_centrality_w1_raw$node.centrality$InStrength*40,
     title = "Temporal\nWave 1",
     loop = .7, # controls self-loop size, defaults to 1
     node.width = 1.6,
     edge.width = 1,
     asize = 5,
     label.font = 2,
     label.fill.vertical = 1,
     label.fill.horizontal = 1,
     edge.color = "black",
     color = t(brewer.pal(9, "Purples")[1:(length(sgc_w1_temporal))]),
     DoNotPlot = TRUE)

# run but don't generate contemporaneous plot
plot_w1_contemp <-   plot(fit1_w1,
     "contemporaneous",
     layout = "spring",
     labels = varnames2, # can be a vector with node labels
     groups = sgc_w1_contemp,
     nonsig = 'hide',
```

```
     curve = -1,
     legend = FALSE,
     details = FALSE,
     mar = c(5,5,5,5), # controls margins
     border.color = contemporaneous_centrality_colors_w1, # controls the border colors
     border.width = contemporaneous_centrality_w1_raw$node.centrality$Strength*5,
     loop = .7, # controls self-loop size, defaults to 1
     node.width = 1.6,
     edge.width = 2,
     label.font = 2,
     label.fill.vertical = 1,
     label.fill.horizontal = 1,
     title = "Contemporaneous\nWave 1",
     esize = 3,
     edge.color = "black",
     color = brewer.pal(9, "Purples")[seq(2, length(sgc_w1_contemp) * 2 ,2)],
     DoNotPlot = TRUE)

 plot_w47 <- plot(fit1_w47,
     "temporal",
     layout = plot_w1$layout,
     labels = varnames2, # can be a vector with node labels
     groups = sgc_w47_temporal,
     nonsig = 'hide',
     curve = -1,
     legend = FALSE,
     details = FALSE,
     mar = c(5,5,5,5), # controls margins
     border.color = temporal_centrality_colors_w47, # controls the border colors
     border.width = temporal_centrality_w47_raw$node.centrality$InStrength*40,
     title = "Wave 2",
     loop = .7, # controls self-loop size, defaults to 1
     node.width = 1.6,
     edge.width = 1,
     asize = 5,
     label.font = 2,
     label.fill.vertical = 1,
     label.fill.horizontal = 1,
     edge.color = "black",
     color = brewer.pal(9, "Purples")[1:length(sgc_w47_temporal)],
     DoNotPlot = TRUE)

 # run but don't generate contemporaneous plot
plot_w47_contemp <-   plot(fit1_w47,
     "contemporaneous",
     #title = "Raw Estimated temporal relationships",
     layout = plot_w1_contemp$layout,
     labels = varnames2, # can be a vector with node labels
     groups = sgc_w47_contemp,
     nonsig = 'hide',
     curve = -1,
     legend = FALSE,
     details = FALSE,
     mar = c(5,5,5,5), # controls margins
     border.color = contemporaneous_centrality_colors_w47, # controls the border colors
     border.width = contemporaneous_centrality_w47_raw$node.centrality$Strength*5,
     loop = .7, # controls self-loop size, defaults to 1
     node.width = 1.6,
     edge.width = 2,
     label.font = 2,
     label.fill.vertical = 1,
     label.fill.horizontal = 1,
     title = "Wave 2",
     esize = 3,
     edge.color = "black",
     color = rev(brewer.pal(9, "Purples")[seq(2, length(sgc_w47_contemp) * 2 ,2)]),
     DoNotPlot = TRUE)
```

```r
#change lines to dashed
plot_w1_contemp$graphAttributes$Edges$lty[plot_w1_contemp$Edgelist$weight < 0] <- 2
plot_w1$graphAttributes$Edges$lty[plot_w1$Edgelist$weight < 0] <- 2
plot_w47_contemp$graphAttributes$Edges$lty[plot_w47_contemp$Edgelist$weight < 0] <- 2
plot_w47$graphAttributes$Edges$lty[plot_w47$Edgelist$weight < 0] <- 2

#change line colors
plot_w1$graphAttributes$Edges$color <-
  ifelse(plot_w1$Edgelist$weight <.02, "thistle2",
  ifelse(plot_w1$Edgelist$weight <.04, "mediumorchid", "midnightblue"))
plot_w1_contemp$graphAttributes$Edges$color <-
  ifelse(abs(plot_w1_contemp$Edgelist$weight) < .1, "thistle2",
  ifelse(abs(plot_w1_contemp$Edgelist$weight) < .3, "mediumorchid", "midnightblue"))
plot_w47$graphAttributes$Edges$color <-
  ifelse(plot_w47$Edgelist$weight <.02, "thistle2",
  ifelse(plot_w47$Edgelist$weight <.04, "mediumorchid", "midnightblue"))
plot_w47_contemp$graphAttributes$Edges$color <-
  ifelse(abs(plot_w47_contemp$Edgelist$weight) < .1, "thistle2",
  ifelse(abs(plot_w47_contemp$Edgelist$weight) < .3, "mediumorchid", "midnightblue"))

#change variable names
plot_w1$graphAttributes$Nodes$labels <- varnames2
plot_w1_contemp$graphAttributes$Nodes$labels <- varnames2
plot_w47$graphAttributes$Nodes$labels <- varnames2
plot_w47_contemp$graphAttributes$Nodes$labels <- varnames2

#change font color for dark nodes
dark_colors <- c("#9E9AC8", "#807DBA", "#6A51A3", "#54278F", "#3F007D")
plot_w1$graphAttributes$Nodes$label.color[plot_w1$graphAttributes$Nodes$color %in% dark_colors] <- "white"
plot_w47$graphAttributes$Nodes$label.color[plot_w47$graphAttributes$Nodes$color %in% dark_colors] <- "white"
plot_w1_contemp$graphAttributes$Nodes$label.color[plot_w1_contemp$graphAttributes$Nodes$color %in% dark_colors] <- "white"
plot_w47_contemp$graphAttributes$Nodes$label.color[plot_w47_contemp$graphAttributes$Nodes$color %in% dark_colors] <- "white"


# generate modified plots
par(mfrow = c(2,2))
plot(plot_w1)
plot(plot_w1_contemp)
plot(plot_w47)
plot(plot_w47_contemp)
```



3.3 Edge Stability

The easiest and most straightforward way to assess the stability in responses across time is simply to correlate the regression coefficients at each time point with those at the other time points. We do this below for models generated using raw and centered data.
For these correlations, we are correlating two vectors, each of which contains 81 weights (9 x 9). We do so once for temporal and once for contemporaneous effects.

```
# extract and correlate temporal and contemporaneous effects and save to a data frame
cors <- data.frame(
  comparison = c("W1 v. W4+7"),
  type = c("temporal", "contemporaneous"),
  raw_cor =
    c(cor(sum_fit1_w1$temporal$fixed,
```

```r
        sum_fit1_w47$temporal$fixed),
    cor(sum_fit1_w1$contemporaneous$pcor,
        sum_fit1_w47$contemporaneous$pcor)),
  raw_p =
    c(cor.test(sum_fit1_w1$temporal$fixed,
             sum_fit1_w47$temporal$fixed)$p.value,
      cor.test(sum_fit1_w1$contemporaneous$pcor,
        sum_fit1_w47$contemporaneous$pcor)$p.value))

pandoc.table(cors, summary = F,
             caption = "Correlations of Temporal Fixed Effects Edges Across Waves")
```

Table 4: Correlations of Temporal Fixed Effects Edges Across Waves

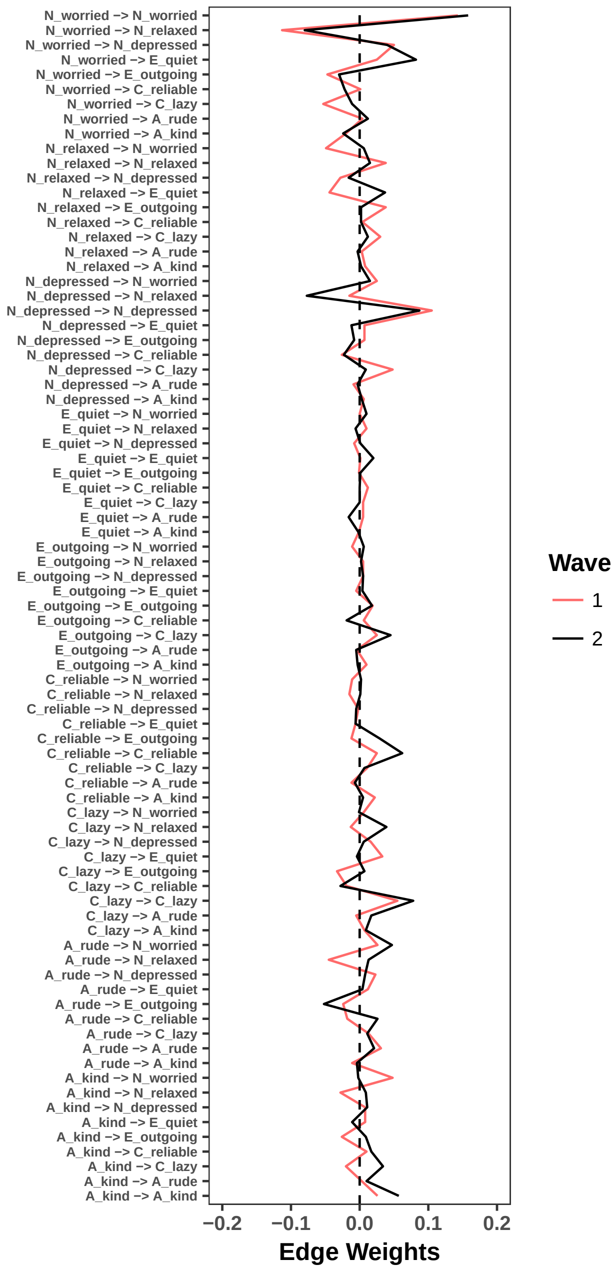
comparison	type	raw_cor	raw_p
W1 v. W4+7	temporal	0.6759	4.428e-12
W1 v. W4+7	contemporaneous	0.9922	2.575e-32

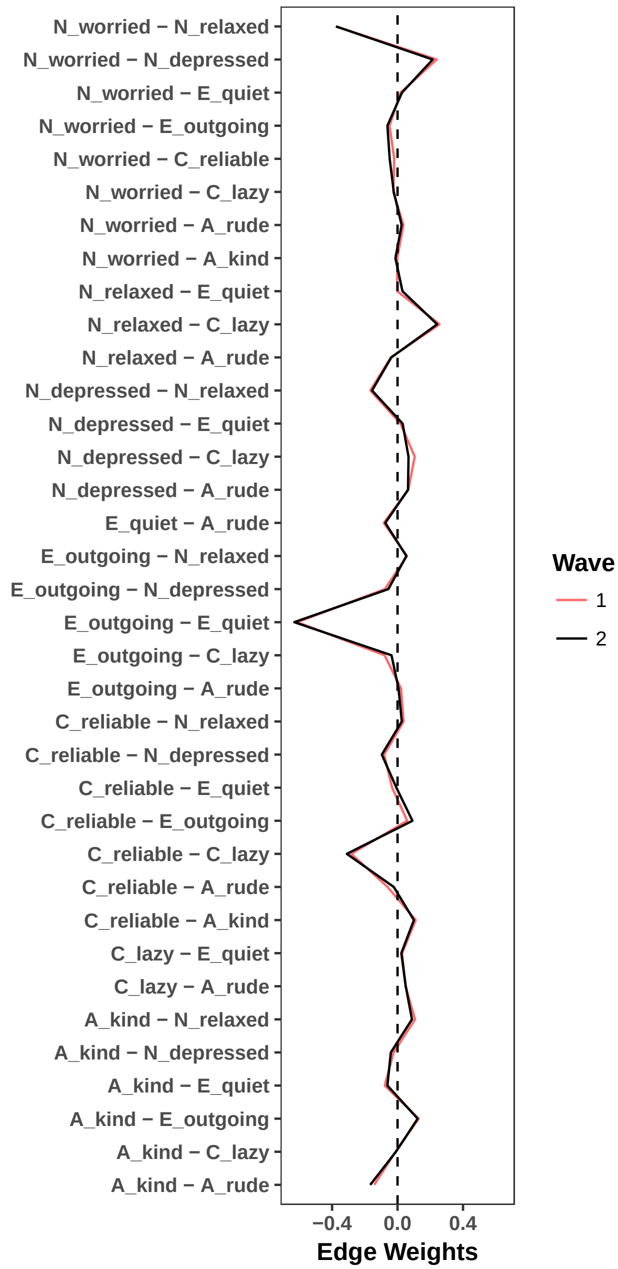
3.4 Local Network Structure

Another way of assessing network structure is locally – that is, by looking at the properties of specific nodes. Below, we test two families of measures of local network structure – centrality and clustering. ###Centrality Centrality refers to the relative importance of a focal node to the structure and dynamics of a network. In other words, it provides information about a node's role in the context of other nodes.

```r
# save temporal effects #
temporal_effects_w1 <- sum_fit1_w1$temporal
temporal_effects_w47 <- sum_fit1_w47$temporal
# save contemporaneous effects (matrix form) #
contemp_effects_w1 <- fit1_w1$results$Theta$pcor$mean
contemp_effects_w47 <- fit1_w47$results$Theta$pcor$mean
# save contemporaneous effects (long form) #
contemp_eff_w1 <- sum_fit1_w1$contemporaneous
contemp_eff_w47 <- sum_fit1_w47$contemporaneous

# create edge variable
temporal_effects_w1$V1V2 <- paste(temporal_effects_w1$from, temporal_effects_w1$to, sep = " -> ")
temporal_effects_w47$V1V2 <- paste(temporal_effects_w47$from, temporal_effects_w47$to, sep = " -> ")
contemp_eff_w1$V1V2 <- paste(contemp_eff_w1$node1, contemp_eff_w1$node2, sep = " - ")
contemp_eff_w47$V1V2 <- paste(contemp_eff_w47$node1, contemp_eff_w47$node2, sep = " - ")

# save wave information
temporal_effects_w1$wave <- "1"
temporal_effects_w47$wave <- "47"
contemp_eff_w1$wave <- "1"
contemp_eff_w47$wave <- "47"

# create profile plot of temporal effects #
temp_eff_plot <- temporal_effects_w1 %>%
  full_join(temporal_effects_w47) %>%
  mutate(wave2 = mapvalues(wave, c("1", "47"), c("1", "2"))) %>%
  ggplot(aes(x = V1V2, y = fixed)) +
    geom_line(aes(group = wave2, color = wave2)) +
    scale_color_manual(values = c("indianred1", "black")) +
    #geom_point(size = .5) +
    ylim(-.2,.2) +
    geom_hline(aes(yintercept = 0), linetype = "dashed") +
    coord_flip() +
    labs(x = NULL, y = "Edge Weights", color = "Wave") +
    theme_bw() +
    theme(axis.text.y = element_text(size = rel(.7), face = "bold"),
          axis.text.x = element_text(face = "bold"),
          axis.title = element_text(face = "bold"),
```

```
          legend.title = element_text(face = "bold"),
          panel.grid.major = element_blank(),
          panel.grid.minor.x = element_blank())
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/pop_temp_EW_ipsative.png", width = 5,  height = 8)

# create profile plot of contemporaneous effects #
contemp_eff_plot <- contemp_eff_w1 %>%
  full_join(contemp_eff_w47) %>%
  mutate(wave2 = mapvalues(wave, c("1", "47"), c("1", "2"))) %>%
  ggplot(aes(x = V1V2, y = pcor)) +
    geom_line(aes(group = wave2, color = wave2)) +
    scale_color_manual(values = c("indianred1", "black")) +
    #geom_point(size = .5) +
    ylim(-.65,.65) +
    geom_hline(aes(yintercept = 0), linetype = "dashed") +
    coord_flip() +
    labs(x = NULL, y = "Edge Weights", color = "Wave") +
    theme_bw() +
    theme(axis.text.y = element_text(size = rel(1), face = "bold"),
          axis.text.x = element_text(face = "bold"),
          axis.title = element_text(face = "bold"),
          legend.title = element_text(face = "bold"),
          panel.grid.major = element_blank(),
          panel.grid.minor.x = element_blank())
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/pop_contemp_EW_ipsative.png", width = 5,  height = 8)

# plot temporal and contemporaneous plots together
grid.arrange(temp_eff_plot, contemp_eff_plot, ncol = 2)
```






```
#temporal
#raw
temporal_centrality_w1_raw      <- centrality_auto(temporal_effects_w1[,c(1,2,4)])
temporal_centrality_w47_raw      <- centrality_auto(temporal_effects_w47[,c(1,2,4)])

#contemporaneous
#raw
contemporaneous_centrality_w1_raw      <- centrality_auto(contemp_effects_w1)
contemporaneous_centrality_w47_raw      <- centrality_auto(contemp_effects_w47)
```

3.4.0.1 Tables

```
# save centrality results into data frame

temp_cent <- temporal_centrality_w1_raw$node.centrality %>%
  mutate(wave = "1", type = "Temporal", var = varnames[6:14])
```

```r
temp_cent <- temporal_centrality_w47_raw$node.centrality %>%
  mutate(wave = "2", type = "Temporal", var = varnames[6:14]) %>%
  full_join(temp_cent)

contemp_cent <- contemporaneous_centrality_w1_raw$node.centrality %>%
  mutate(wave = "1", type = "Contemporaneous", var = varnames[6:14])
contemp_cent <- contemporaneous_centrality_w47_raw$node.centrality %>%
  mutate(wave = "2", type = "Contemporaneous", var = varnames[6:14]) %>%
  full_join(contemp_cent)

temp_cent_wide <- temp_cent %>%
  gather(key = Measure, value = Centrality, Betweenness: OutStrength) %>%
  spread(key = var, value = Centrality) %>%
  select(Measure, wave, everything(), -type) %>%
  arrange(Measure, wave)

contemp_cent_wide <- contemp_cent %>%
  gather(key = Measure, value = Centrality, Betweenness: Strength) %>%
  spread(key = var, value = Centrality) %>%
  select(Measure, wave,  everything(), -type) %>%
  arrange(Measure, wave)


stargazer(temp_cent_wide, digits = 2, summary = F, font.size = "footnotesize",
          title = "Centrality Table of Temporal Fixed Effects", rownames = FALSE,
          column.sep.width = "0pt")

stargazer(contemp_cent_wide, digits = 2, summary = F, font.size = "footnotesize",
          title = "Centrality Table of Contemporaneous Fixed Effects", rownames = FALSE,
          column.sep.width = "0pt")
```

Table 5: Centrality Table of Temporal Fixed Effects

Measure	wave	A_rude	E_quiet	C_lazy	N_relaxed	N_depressed	E_outgoing	A_kind	C_reliable	N_worried
Betweenness	1	0	0	6	9	6	0	3	6	12
Betweenness	2	14	6	16	10	1	10	0	3	9
Closeness	1	0.003	0.001	0.002	0.002	0.002	0.001	0.002	0.002	0.003
Closeness	2	0.003	0.001	0.002	0.001	0.002	0.002	0.002	0.001	0.003
InStrength	1	0.04	0.14	0.20	0.24	0.14	0.19	0.08	0.10	0.18
InStrength	2	0.07	0.16	0.13	0.23	0.09	0.14	0.05	0.14	0.09
OutStrength	1	0.17	0.04	0.13	0.20	0.14	0.07	0.15	0.09	0.31
OutStrength	2	0.16	0.04	0.11	0.08	0.15	0.09	0.10	0.06	0.30

Table 6: Centrality Table of Contemporaneous Fixed Effects

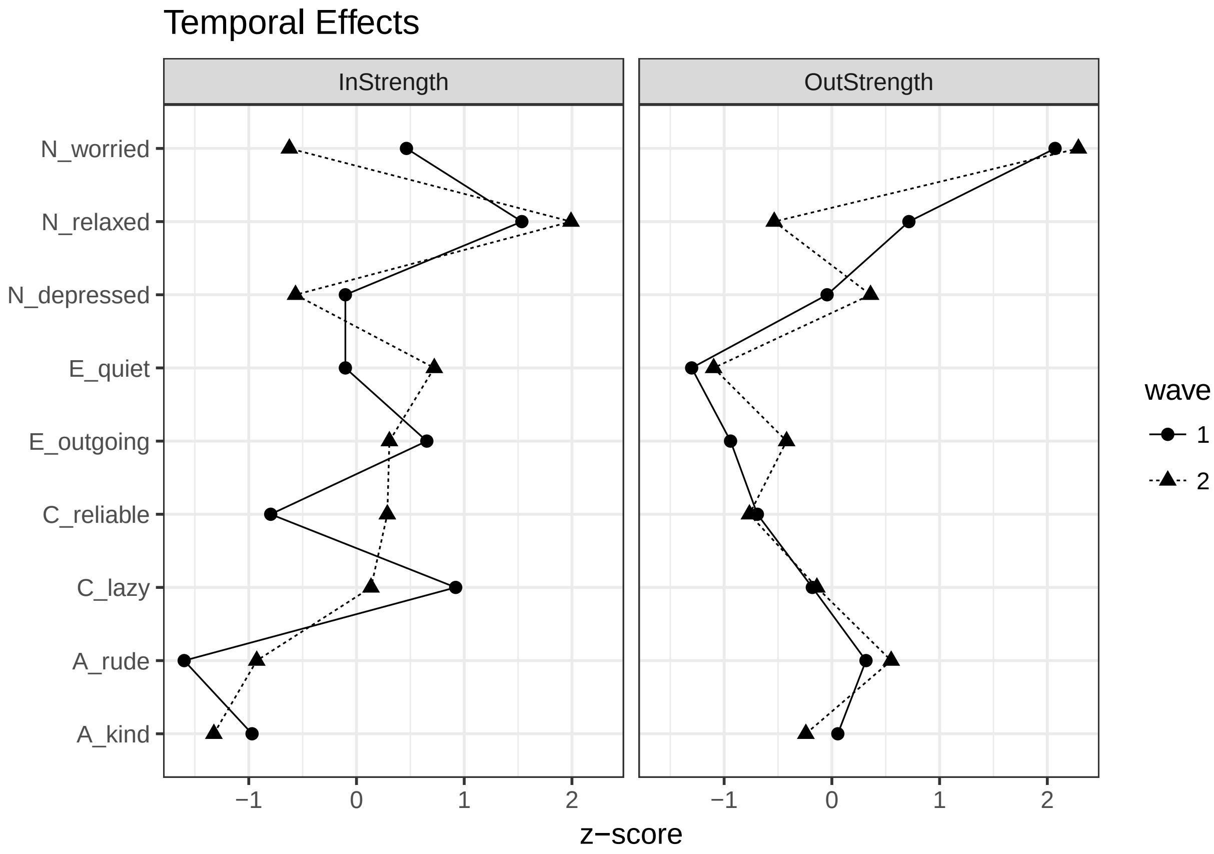
Measure	wave	A_rude	E_quiet	C_lazy	N_relaxed	N_depressed	E_outgoing	A_kind	C_reliable	N_worried
Betweenness	1	0	1	6	5	2	6	4	2	0
Betweenness	2	0	0	2	6	0	6	5	4	0
Closeness	1	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Closeness	2	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Strength	1	0.50	0.87	0.84	1.05	0.78	1.08	0.61	0.68	0.77
Strength	2	0.45	0.89	0.76	1.01	0.73	1.06	0.60	0.70	0.79

3.4.0.2 Plots

3.4.0.2.1 Temporal

```r
temp_cent_long <- temp_cent %>%
  gather(key = Measure, value = Centrality, Betweenness:OutStrength) %>%
  group_by(wave, Measure) %>%
  mutate(z = scale(Centrality))
```

```
temp_cent_long %>%
  filter(!(Measure == "Closeness") & !(Measure == "Betweenness")) %>%
  ggplot(aes(x = var, y = z, group = wave))+
    geom_line(aes(linetype = wave), color = "black", size = .3) +
    geom_point(aes(shape = wave), size = 2) +
    labs(x = NULL, y = "z-score", title = "Temporal Effects") +
    coord_flip() +
    facet_grid(.~Measure) +
    theme_bw()
```

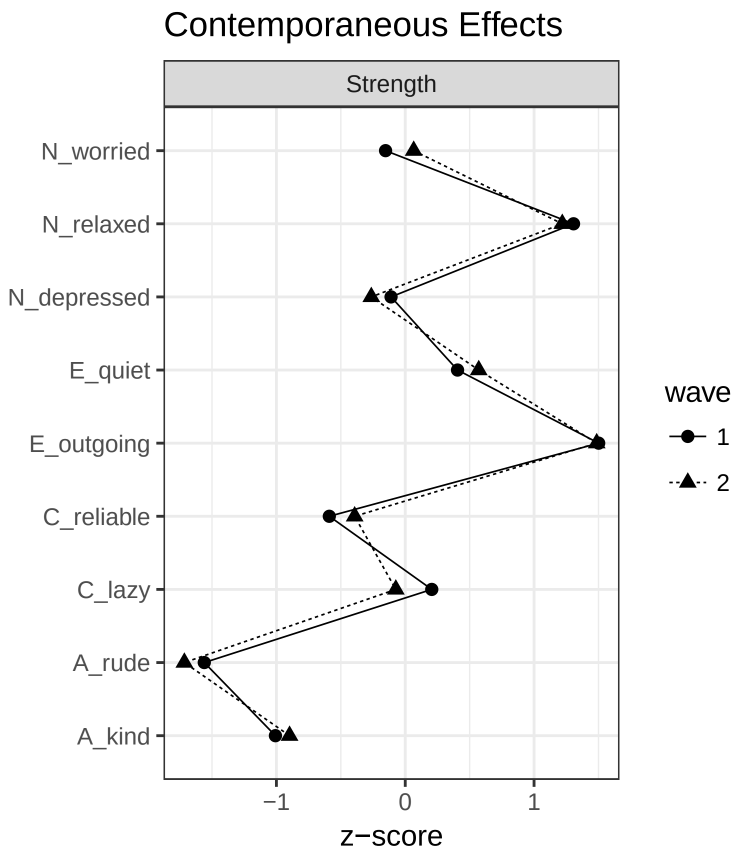



```
w1_cent <- temp_cent_long %>%
  filter(wave == "1" & (Measure == "InStrength" | Measure == "OutStrength")) %>%
ggplot(aes(x = var, y = z, group = wave))+
  geom_line(aes(linetype = wave), color = "black", size = .3) +
  geom_point(aes(shape = wave), size = 2) +
  geom_hline(aes(yintercept = 0), linetype = "dashed") +
  labs(x = NULL, y = "z-score",
       linetype = "Wave", shape = "Wave") +
  coord_flip() +
  facet_grid(.~Measure) +
  theme_bw() +
  theme(axis.text = element_text(face = "bold"),
        axis.title = element_text(face = "bold"),
        legend.title = element_text(face = "bold"))
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/pop_temporal_centrality_w1.png",
       width = 6, height = 5)

w47_cent <- temp_cent_long %>%
  filter(wave == "2"& (Measure == "InStrength" | Measure == "OutStrength")) %>%
ggplot(aes(x = var, y = z, group = wave)) +
  geom_line(aes(linetype = wave), linetype = "dashed", color = "black", size = 1) +
  geom_point( shape = 15, aes(shape = wave),size = 2) +
  geom_point(aes(shape = wave)) +
```

```r
  geom_hline(aes(yintercept = 0), linetype = "dashed") +
  labs(x = NULL, y = "z-score",
       linetype = "Wave", shape = "Wave") +
  coord_flip() +
  facet_grid(.~Measure) +
  theme_bw()+
  theme(axis.text = element_text(face = "bold"),
        axis.title = element_text(face = "bold"),
        legend.title = element_text(face = "bold"))
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/pop_temporal_centrality_w1.png",
       width = 6, height = 5)
```

3.4.0.2.2 Contemporaneous

```r
# Contemporaneous #

contemp_cent_long <- contemp_cent %>%
  gather(key = Measure, value = Centrality, Betweenness:Strength) %>%
  group_by(wave, Measure) %>%
  mutate(z = scale(Centrality))


contemp_cent_plot <- ggplot(contemp_cent_long, aes(x = var, y = z, group = wave))+
  geom_line(aes(linetype = wave), color = "black", size = .3) +
  geom_point(aes(shape = wave), size = 2) +
  labs(x = NULL, y = "z-score", title = "Contemporaneous Effects") +
  coord_flip() +
  facet_grid(.~Measure) +
  theme_bw()

contemp_cent_long %>%
  filter(Measure == "Strength") %>%
  ggplot(aes(x = var, y = z, group = wave))+
  geom_line(aes(linetype = wave), color = "black", size = .3) +
  geom_point(aes(shape = wave), size = 2) +
  labs(x = NULL, y = "z-score", title = "Contemporaneous Effects") +
  coord_flip() +
  facet_grid(.~Measure) +
  theme_bw()
```




```
w1_cent <- contemp_cent_long %>%
  filter(wave == "1" & Measure == "Strength") %>%
  ggplot(aes(x = var, y = z, group = wave))+
  geom_line(aes(linetype = wave),  color = "black", size = .8) +
  geom_point( aes(shape = wave),size = 2) +
  labs(x = NULL, y = "z-score",
       linetype = "Wave", shape = "Wave") +
  coord_flip() +
  facet_grid(.~Measure) +
  theme_bw() +
  theme(axis.text = element_text(face = "bold"),
        axis.title = element_text(face = "bold"),
        legend.title = element_text(face = "bold"))
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/pop_contemp_centrality_w1.png",
       width = 4, height = 5)

w47_cent <- contemp_cent_long %>%
  filter(wave == "2" & Measure == "Strength") %>%
  ggplot(aes(x = var, y = z, group = wave))+
  geom_line(aes(linetype = wave), linetype = "dashed", color = "black", size = 1) +
  geom_point( shape = 15, aes(shape = wave),size = 2) +
  geom_point(aes(shape = wave)) +
  labs(x = NULL, y = "z-score", title = "Contemporaneous Effects") +
  coord_flip() +
  facet_grid(.~Measure) +
  theme_bw()
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/pop_contemp_centrality_w1.png",
       width = 4, height = 5)

combined_centrality <- temp_cent_long %>% full_join(contemp_cent_long) %>% ungroup()

combined <- ggplot(combined_centrality, aes(x = var, y = z, group = wave))+
  geom_line(aes(linetype = wave), color = "black", size = .3) +
  geom_point(aes(shape = wave), size = 2) +
```

```r
  labs(x = NULL, y = "z-score") +
  coord_flip() +
  facet_wrap(~type + Measure, nrow= 2) +
  theme_bw() +
  theme(legend.position = c(.87, .25))
ggsave(plot = combined, "~/Box Sync/network/PAIRS/manuscript/centrality_plot.png", width = 6, height = 5)
```

3.4.0.3 Centrality Edge Correlations

One simple way to assess the centrality or importance of a node in a network over time is to correlate centrality indices across the waves. Below, we calculate profile correlations of node centrality across the waves for 4 (temporal) or 3 (contemporaneous) measures of centrality. #####Temporal

```r
cor_fun <- function(x){cor(x$`1`, x$`2`, use = "pairwise")}

pop_cors <- tbl_df(combined_centrality) %>%
  select(-z) %>%
  spread(key = wave, value = Centrality) %>%
  group_by(type, Measure) %>%
  nest() %>%
  mutate(r = map(data, cor_fun)) %>%
  unnest(r, .drop = T) %>%
  spread(key = Measure, value = r)
```

4 Idiographic Networks

Although mlVAR includes both population and subject level effects, it represents subject level effects as deviations from population effects rather than exmaning unique subject-level patterns. To assess such unique effects, below we construct individual networks for all subjects at each wave.

4.1 Clean Data and Prepare to Run Models

```r
# get subjects
subs_w1  <- as.character(unique(w1_com$SID))
subs_w47 <- as.character(unique(w47_com$SID))

# find subjects' SDs of responses
# need to flat subjects with SD's = 0
w1_test <- w1_com[,c(1,6:14)] %>%
  group_by(SID) %>%
  mutate_each(funs(sd = sd(., na.rm = TRUE)))
w47_test <- w47_com[,c(1,6:14)] %>%
  group_by(SID) %>%
  mutate_each(funs(sd = sd(., na.rm = TRUE)))

for(i in 1:9){
  for(k in 1:dim(w1_test)[1]){
    if(w1_test[k, i + 10] == 0){
      w1_test[k, i + 1] <-
        jitter(as.numeric(w1_test[k, i + 1]), amount = runif(1, 0, .05))
    }
  }
}

for(i in 1:9){
  for(k in 1:dim(w47_test)[1]){
    if(w47_test[k, i + 10] == 0){
      w47_test[k, i + 1] <-
        jitter(as.numeric(w47_test[k, i + 1]), amount = runif(1, 0, .05))
    }
```

```
  }
}
```

4.1.1 Run Models

For idiographic networks, we estimate a Gaussian graphical model (GGM) variation of the vector autoregressive model (VAR), which estimates a partial correlation network in which correlations represent the correlation between variables after conditioning on all other variables. These models are regularized using a variant of the least absolute shrinkage and selection operator (LASSO), graphical LASSO (glasso). In addition, glasso includes a tuning parameter that can be set to control the sparsity of the network. Different values of the parameter can be chosen to optimize prediction accuracy to minimize an information criterion, such as the Bayesian information criterion (BIC) or the extended BIC (EBIC; Chen & Chen, 2008).

```
w1_test <- read.csv("~/Box Sync/network/PAIRS/Wave 1/esm_w1_jittered.csv")
w47_test <- read.csv("~/Box Sync/network/PAIRS/Wave 4 + 7/esm_w47_jittered.csv")

# save those subjects to a vector
subs2_w1 <- as.character(unique(w1_test$SID))
subs2_w47 <- as.character(unique(w47_test$SID))


gVAR_fun <- function(x){
  n <- dim(x)[1]
  gamma <- .005 * n
  lambda <- .003 * n
  fit <-
    graphicalVAR_test(x, gamma = gamma, maxit.in = 1000, maxit.out = 1000,
                      lambda_beta = lambda, lambda_kappa = lambda,
                      verbose = F, scale = F)
  return(fit)
}

gVAR_fit_w1 <- w1_test %>%
  select(-contains("sd")) %>%
  group_by(SID) %>%
  nest() %>%
  mutate(gVAR_fit = map(data, possibly(gVAR_fun, NA_real_)))
save(gVAR_fit_w1, file = "~/Box Sync/network/PAIRS/Wave 1/graphicalVAR_allSubs.RData")

gVAR_fit_w47 <- w47_test %>%
  select(-contains("sd")) %>%
  group_by(SID) %>%
  nest() %>%
  mutate(gVAR_fit = map(data, possibly(gVAR_fun, NA_real_)))

save(gVAR_fit_w47,
     file = "~/Box Sync/network/PAIRS/Wave 4 + 7/graphicalVAR_allSubs.RData")
```

4.2 Extract Results and Save Into Data Frames

4.2.1 Temporal: Partial Directed Correlations

```
load("~/Box Sync/network/PAIRS/Wave 1/graphicalVAR_allSubs.RData")
load("~/Box Sync/network/PAIRS/Wave 4 + 7/graphicalVAR_allSubs.RData")
beta_fun <- function(fit, SID){
  PDC <- fit$PDC
  from <- row.names(PDC)
  PDC.long <- tbl_df(PDC) %>%
    mutate(from = from,
           type = "Temporal") %>%
    gather(key = to, value = weight, A_rude:N_worried)
```

```
}

kappa_mat_fun <- function(fit){fit$PCC}

kappa_long_fun <- function(fit){
  PCC <- fit$PCC
  PCC[lower.tri(PCC, diag = T)] <- NA
  vars <- rownames(PCC)
  PCC.long <- tbl_df(PCC) %>%
    mutate(Var1 = vars,
           type = "Contemporaneous") %>%
    gather(key = Var2, value = weight, A_rude:N_worried) %>%
    filter(!is.na(weight)) %>%
    unite(var, Var1, Var2, sep = ".", remove = F)
}

gVAR_fit_w1 <- gVAR_fit_w1 %>%
  filter(!is.na(gVAR_fit)) %>%
  mutate(wave = "1",
         beta = map2(gVAR_fit, SID, beta_fun),
         kappa_mat = map(gVAR_fit, kappa_mat_fun),
         kappa = map(gVAR_fit, kappa_long_fun))
gVAR_fit_w47 <- gVAR_fit_w47 %>%
  filter(!is.na(gVAR_fit)) %>%
  mutate(wave = "47",
         beta = map2(gVAR_fit, SID, beta_fun),
         kappa_mat = map(gVAR_fit, kappa_mat_fun),
         kappa = map(gVAR_fit, kappa_long_fun))

beta_long <- gVAR_fit_w1 %>%
  unnest(beta)
beta_long <- gVAR_fit_w47 %>%
  unnest(beta) %>%
  full_join(beta_long)

# get SIDs from models
SID_w1  <- as.character(unique(gVAR_fit_w1$SID))
SID_w47 <- as.character(unique(gVAR_fit_w47$SID))

#get variable names from models
varnames <- unique(beta_long$from)

# find subjects in both waves
w1w47_subs <- unique(SID_w1)[unique(SID_w1) %in% unique(SID_w47)]
```

4.2.2 Contemporaneous: Partial Contemporaneous Correlations

```
kappa_w1  <- gVAR_fit_w1$kappa
kappa_w47 <- gVAR_fit_w47$kappa

kappa_long <- gVAR_fit_w1 %>%
  unnest(kappa)
kappa_long <- gVAR_fit_w47 %>%
  unnest(kappa) %>%
  full_join(kappa_long) %>%
  mutate(Var1 = factor(Var1, levels = varnames),
         Var2 = factor(Var2, levels = varnames))

kable(kappa_long %>%
  group_by(wave, var) %>%
  summarise(r = meanSD_r2z2r(weight)[1]) %>%
  arrange(desc(r)), caption = "Average Idiographic Partial Contemporaneous Correlations")
```

4.3 Stability

4.3.1 Rank-Order

One way to conceptualize edge stability is to examine the stability of specific edges across all subjects over time. That is, are some edges more stable than others? ####PDC's

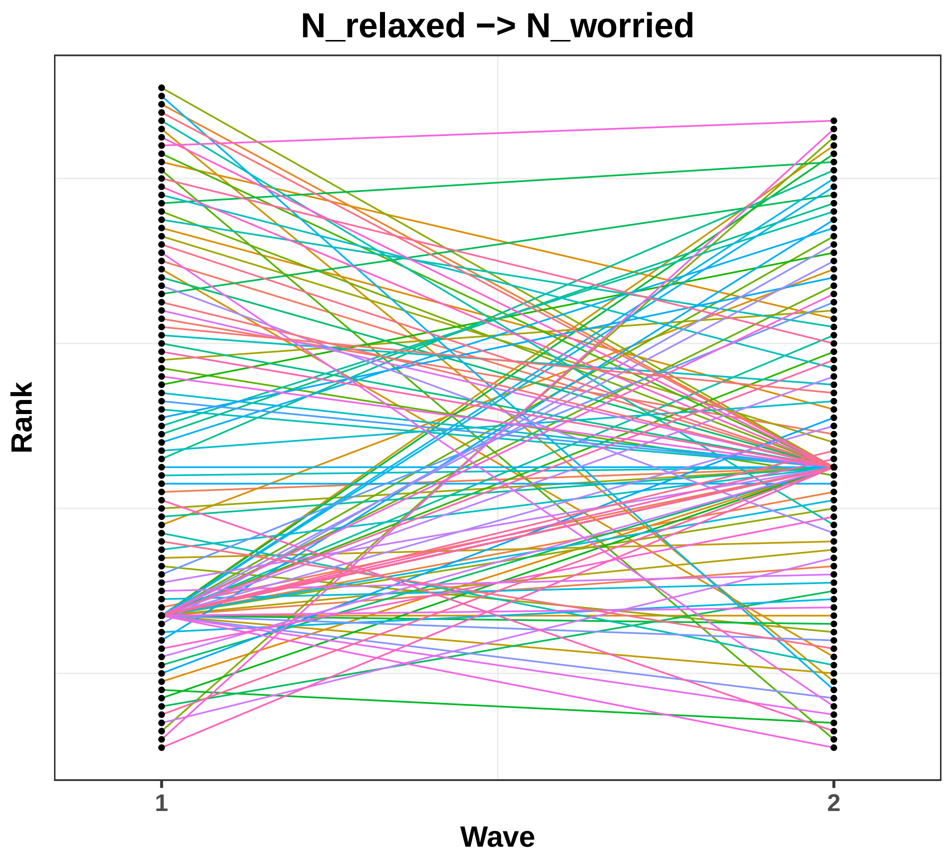
```
cor_fun <- function(x){
    results <- cor(x$`1`, x$`47`, use = "pairwise", method = "spearman")
}

beta_long <- beta_long %>%
  dplyr::filter(SID %in% w1w47_subs) %>%
  group_by(wave, from, to) %>%
  mutate(rank = dense_rank(desc(weight))) %>%
  ungroup() %>%
  mutate(x = ifelse(wave == "1", -1, 1))


beta_cors_long <- beta_long %>%
  select(-x, -weight) %>%
  spread(key = wave, value = rank) %>%
  group_by(from, to) %>%
  nest() %>%
  mutate(r = map(data, cor_fun))

beta_cors_long.df <- beta_cors_long %>% unnest(r, .drop = T)

kable(beta_cors_long.df %>%
  summarize(mean = meanSD_r2z2r(r)[1],
            sd = meanSD_r2z2r(r)[2]), caption = "Average Idiographic Partial Contemporaneous Correlations")
```

Table 7: Average Idiographic Partial Contemporaneous Correlations

mean	sd
0.0036964	0.0914211

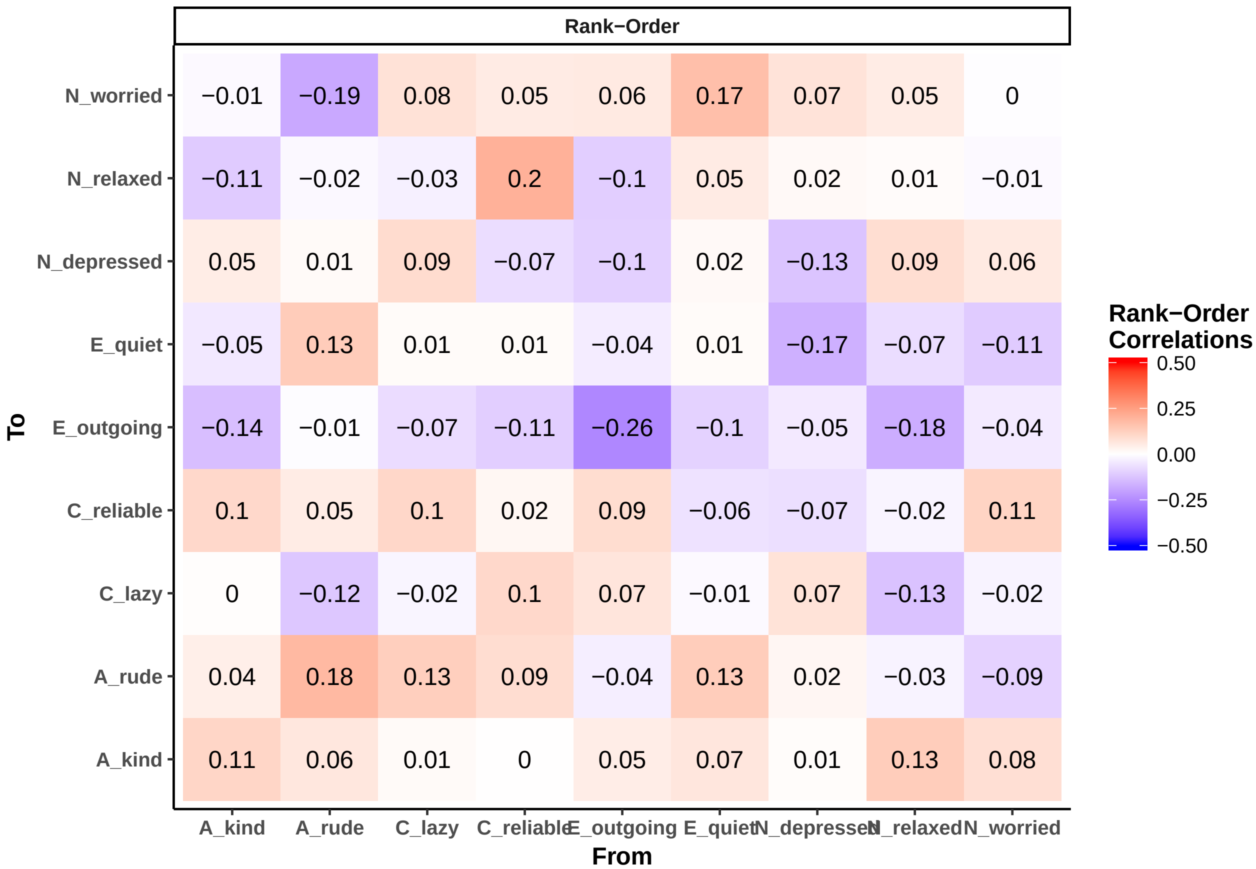
```
PDC_cors_heatmap <- beta_cors_long.df %>%
  mutate(measure = "Rank-Order") %>%
  # mutate(Var1 = factor(Var1, levels = sort(as.character(unique(Var1)))),
  #        Var2 = factor(Var2, levels = sort(as.character(unique(Var2))))) %>%
  filter(measure == "Rank-Order") %>%
  ggplot(aes(x = from, y = to, fill = r)) +
    geom_raster() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-.5,.5), space = "Lab",
   name="Rank-Order\nCorrelations") +
  geom_text( aes(label = round(r,2))) +
  facet_grid(.~measure) +
  labs(x = "From", y = "To") +#, title = "Temporal Edge Stability Correlations") +
  theme_classic() +
  theme(axis.text = element_text(face = "bold"),
        axis.title = element_text(face = "bold"),
        strip.text = element_text(face = "bold"),
        legend.title = element_text(face = "bold"))
ggsave(file = "~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/PDC_Edge_Cors_heatmap.png", width = 9, height = 6)

vars <- rep(NA, 81)
k <- 1
for(i in varnames[1:9]){for(j in varnames[1:9]){vars[k] <- paste(i, j, sep = "."); k <- k+1}}


draw.a.plot <-  function(x){
  var <- separate(as.data.frame(x), x, into = c("from", "to"), sep = "[.]")
  data <- beta_long %>%
```

```r
    filter(from == var$from & to == var$to &
             SID %in% w1w47_subs) %>%
    mutate(V1V2 = paste(from, to, sep = " -> "),
           wave = recode(wave, `1` = "1", `47` = "2"))
  ggplot(data, aes(x = as.numeric(wave), y = rank, group = factor(SID))) +
    geom_line(aes(color = factor(SID)), size = .3) +
    scale_x_continuous(limits = c(.9,2.1), breaks = 1:2) +
    #scale_color_manual(values = c("indianred1", "black")) +
    geom_point(size = .5) +
    labs(x = "Wave", y = "Rank", color = "Subject",
         title = data$V1V2) +
    theme_bw() +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank(),
          panel.grid.major = element_blank(),
          plot.title = element_text(hjust = .5, face = "bold"),
          axis.title = element_text(face = "bold"),
          axis.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold"),
          legend.position = "none")
}

draw.a.plot("N_relaxed.N_worried")
```



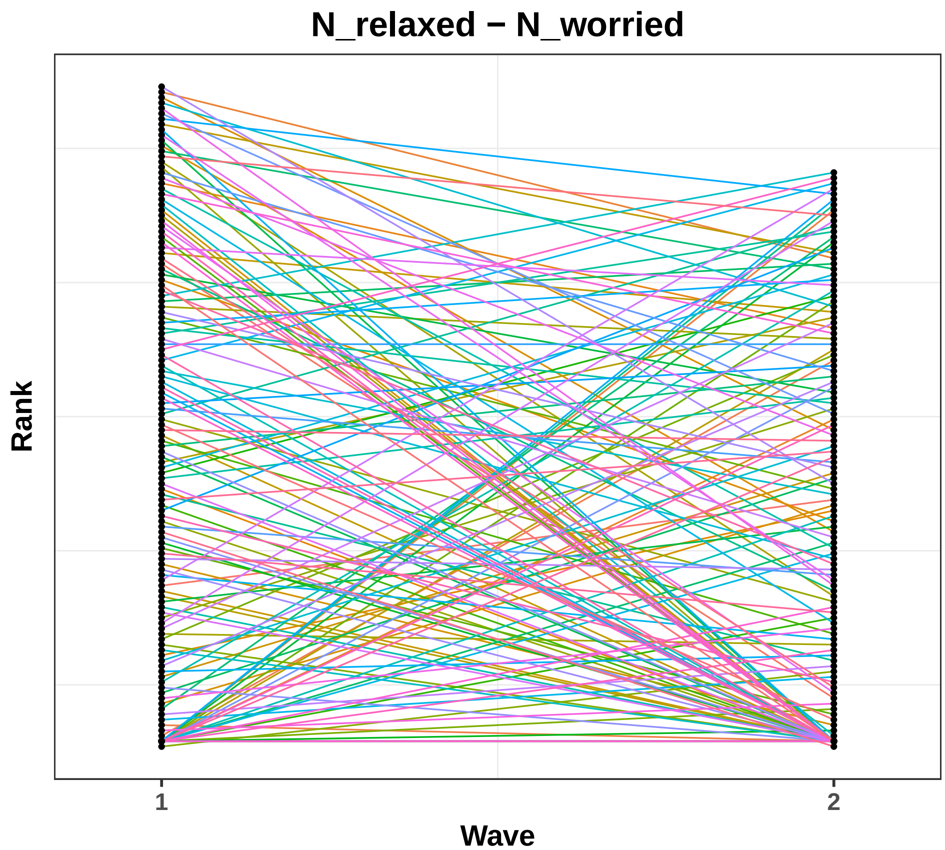

```r
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/PDC_vars_denserank.png", width = 5, height = 8)


# saveGIF(loop.animate(), interval = .5, movie.name="PDC_vars_denserank.gif", ani.width = 250, ani.height = 400,
#         imgdir = "~/Box Sync/network/PAIRS/Brown Bag 3.31/")
```

```r
PDC_cors_heatmap
```



4.3.1.1 PCC's

```
#### rank ####

draw.a.plot <-  function(x){
  var <- separate(as.data.frame(x), x, into = c("Var1", "Var2"), sep = "[ ]")
  data <- kappa_long %>%
    filter(Var1 == var$Var1 & Var2 == var$Var2 &
             SID %in% w1w47_subs) %>%
    mutate(V1V2 = paste(Var1, Var2, sep = " - "),
           wave = recode(wave, `1` = "1", `47` = "2")) %>%
    group_by(wave, V1V2) %>%
    mutate(rank = dense_rank(desc(weight)))
  ggplot(data, aes(x = as.numeric(wave), y = rank, group = factor(SID))) +
    geom_line(aes(color = factor(SID)), size = .3) +
    scale_x_continuous(limits = c(.9,2.1), breaks = 1:2) +
    #scale_color_manual(values = c("indianred1", "black")) +
    geom_point(size = .5) +
    labs(x = "Wave", y = "Rank", color = "Subject",
         title = data$V1V2) +
    theme_bw() +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank(),
          panel.grid.major = element_blank(),
          plot.title = element_text(hjust = .5, face = "bold"),
          axis.title = element_text(face = "bold"),
          axis.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold"),
          legend.position = "none")
}

draw.a.plot("N_relaxed N_worried")
```



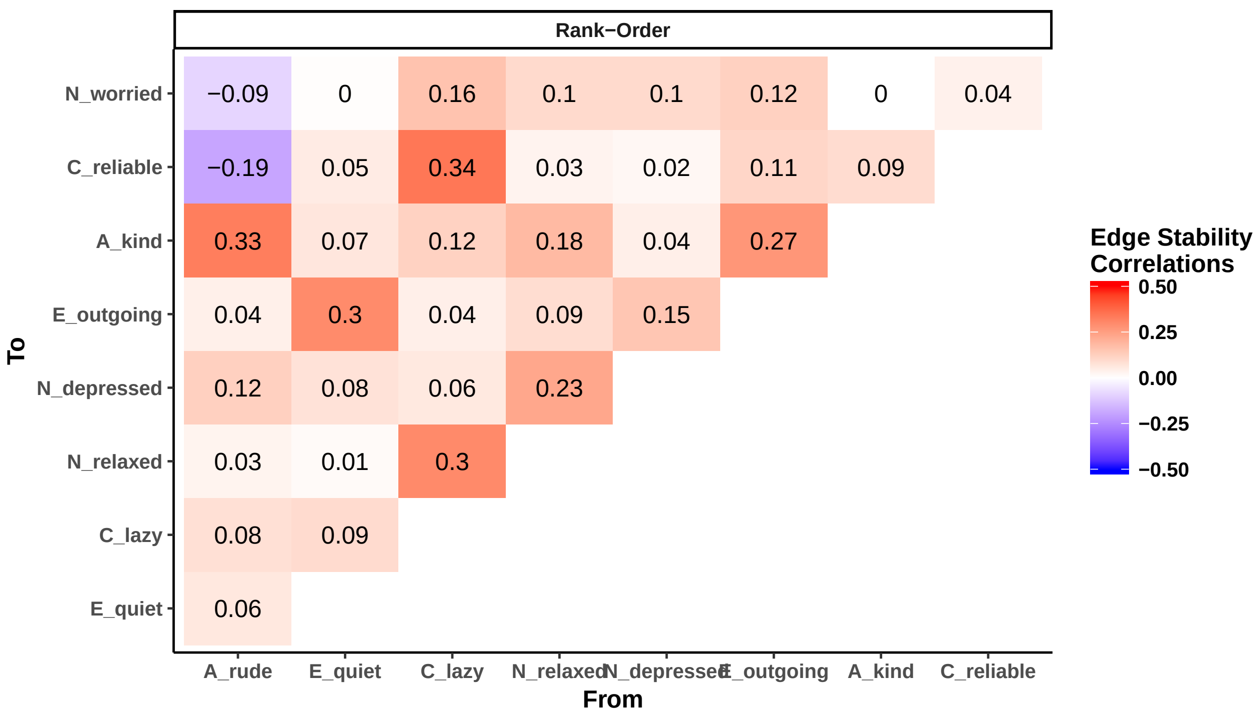

```r
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/PCC_vars_denserank.png", width = 5, height = 8)


# saveGIF(loop.animate(), interval = .5, movie.name="PCC_vars_denserank.gif", ani.width = 250, ani.height = 400,
#          imgdir = "~/Box Sync/network/PAIRS/Brown Bag 3.31/")

cor_fun <- function(x){
    results <- cor(x$`1`, x$`47`, use = "pairwise", method = "spearman")
}

kappa_long <- kappa_long %>%
  dplyr::filter(SID %in% w1w47_subs) %>%
  group_by(wave, Var1, Var2) %>%
  mutate(rank = min_rank(desc(weight))) %>%
  ungroup() %>%
  mutate(x = ifelse(wave == "1", -1, 1))

kappa_cors_long <- kappa_long %>%
  select(-x, -weight) %>%
  spread(key = wave, value = rank) %>%
  group_by(Var1, Var2) %>%
  nest() %>%
  mutate(r = map(data, cor_fun))

kappa_cors_long.df <- kappa_cors_long %>% unnest(r, .drop = T) %>%
  mutate(measure = "Rank-Order", "Absolute Change")

kable(kappa_cors_long.df %>%
  group_by(measure) %>%
  summarize(mean = meanSD_r2z2r(r)[1],
            sd = meanSD_r2z2r(r)[2]), caption = "Average Idiographic Partial Contemporaneous Rank-Order Correlations")
```

Table 8: Average Idiographic Partial Contemporaneous Rank-Order Correlations

measure	mean	sd
Rank-Order	0.1001744	0.1139535

```
PCC_cors_heatmap <- kappa_cors_long.df %>%
  # mutate(Var1 = factor(Var1, levels = sort(as.character(unique(Var1)))),
  #        Var2 = factor(Var2, levels = sort(as.character(unique(Var2))))) %>%
  filter(measure == "Rank-Order") %>%
  ggplot(aes(x = Var1, y = Var2, fill = r)) +
    geom_raster() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-.5,.5), space = "Lab",
   name="Edge Stability\nCorrelations") +
  geom_text(aes(label = round(r,2))) +
  #geom_text(data = filter(PCC_cors_long.df, measure == "Absolute Change"), aes(label = round(sd,2))) +
  facet_grid(.~measure) +
  labs(x = "From", y = "To") +
  theme_classic() +
  theme(strip.text = element_text(face = "bold"),
        axis.text = element_text(face = "bold"),
        axis.title = element_text(face = "bold"),
        legend.title = element_text(face = "bold"),
        legend.text = element_text(face = "bold"))
ggsave(file = "~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/PCC_Edge_Cors_heatmap.png", width = 9, height = 6)
```

```
PCC_cors_heatmap
```



4.3.2 Ipsative Edge Stability

Another way to conceptualize stability is to assess a single person's stability over time. Below, we take profile correlations of all edges for each subject individually for both temporal and contemporaneous effects. The results are then displayed in a histogram.

```
ip_beta_cors <- beta_long %>%
  dplyr::filter(SID %in% w1w47_subs) %>%
  arrange(wave, SID, from, to) %>%
```

```
  select(-x, -rank) %>%
  spread(wave, weight) %>%
  group_by(SID) %>%
  summarize(cors = cor(`1`, `47`, use = "pairwise.complete.obs")) %>%
  mutate(type = "PDC")

ip_cors <- kappa_long %>%
  dplyr::filter(SID %in% w1w47_subs) %>%
  arrange(wave, SID, var) %>%
  select(-x, -rank) %>%
  spread(wave, weight) %>%
  group_by(SID) %>%
  summarize(cors = cor(`1`, `47`, use = "pairwise.complete.obs")) %>%
  mutate(type = "PCC") %>%
  full_join(ip_beta_cors)

ip_cors$type <- factor(ip_cors$type, levels = c("PDC", "PCC"))
ip_cors$type2 <- factor(ifelse(ip_cors$type == "PDC", "Temporal", "Contemporaneous"),
                        levels = c("Temporal", "Contemporaneous"))

kable(ip_cors %>%
  group_by(type2) %>%
  summarize(mean_cor = meanSD_r2z2r(cors)[1],
            sd_cor = meanSD_r2z2r(cors)[2]), caption = "Average Ipsative Correlations")

kable(ip_cors %>%
  filter(SID == "90"), "Ipsative Correlation for Subject 90")

ggplot(ip_cors, aes(x = cors)) +
  geom_histogram(color = "black", fill = "gray") +
  labs(x = "Ipsative Correlations", y = "Frequency",
       title = "Ipsative Correlations of graphicalVAR Edge Weights") +
  xlim(c(-1, 1)) +
  facet_grid(.~type2) +
  theme_bw()
ggsave("~/Box Sync/network/PAIRS/manuscript/ipsative_cor.png", width  = 6, height = 3)

PDC_ip_cors <- ip_cors %>%
  filter(type2 == "Temporal") %>%
  ggplot(aes(x = cors)) +
  geom_histogram(color = "black", fill = "gray") +
  labs(x = "Ipsative Correlations", y = "Frequency",
       title = "Ipsative Correlations of graphicalVAR Edge Weights") +
  xlim(c(-1, 1)) +
  facet_grid(.~type2) +
  theme_bw() +
  theme(plot.title = element_text(hjust = .5))

PCC_ip_cors <- ip_cors %>%
  filter(type2 == "Contemporaneous") %>%
  ggplot(aes(x = cors)) +
  geom_histogram(color = "black", fill = "gray") +
  labs(x = "Ipsative Correlations", y = "Frequency",
       title = "Ipsative Correlations of graphicalVAR Edge Weights") +
  xlim(c(-1, 1)) +
  facet_grid(.~type2) +
  theme_bw() +
  theme(plot.title = element_text(hjust = .5))

draw.a.plot <-  function(sub){
  data <- beta_long %>%
    filter(SID == sub) %>%
    mutate(V1V2 = paste(from, to, sep = " -> "),
           wave = recode(wave, `1` = "1", `47` = "2"))
  ggplot(data, aes(x = V1V2, y = weight, group = wave)) +
    geom_line(aes(color = wave), size = .3) +
    #geom_point(size = .5) +
```

```r
    scale_color_manual(values = c("indianred1", "black")) +
    labs(x = NULL, y = "Edge Weight", color = "Wave", title = sprintf("Subject %s", sub)) +
    geom_hline(aes(yintercept = 0), linetype = "dashed") +
    coord_flip() +
    theme_bw() +
    theme(axis.text.y = element_text(size = rel(.7), face = "bold"),
            axis.text.x = element_text(face = "bold"),
            axis.title = element_text(face = "bold"),
            legend.title = element_text(face = "bold"),
            panel.grid.major = element_blank(),
            panel.grid.minor.x = element_blank(),
            plot.title = element_text(hjust = .5))
}

draw.a.plot("90")
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/PDC_EW_ipsative.png", width = 5,  height = 8)


#set up function to loop through the draw.a.plot() function
loop.animate <- function() {
    lapply(as.character(w1w47_subs[10:30]), function(i) {
        print(draw.a.plot(i))
    })
}

# saveGIF(loop.animate(), interval = .5, movie.name="PDC_EW_ipsative.gif", ani.width = 300, ani.height = 450,
#         imgdir = "~/Box Sync/network/PAIRS/Brown Bag 3.31/")


draw.a.plot <-  function(z){
  data <- kappa_long %>%
    filter(SID == z) %>%
    mutate(V1V2 = paste(Var1, Var2, sep = " - "),
           wave = recode(wave, `1` = "1", `47` = "2"))
  ggplot(data, aes(x = V1V2, y = weight, group = wave)) +
    geom_line(aes(color = wave), size = .3) +
    scale_color_manual(values = c("indianred1", "black")) +
    geom_hline(aes(yintercept = 0), linetype = "dashed") +
    #geom_point(size = .5) +
    labs(x = NULL, y = "Edge Weight", color = "Wave", title = sprintf("Subject %s", z)) +
    coord_flip() +
    theme_bw() +
    theme(axis.text.y = element_text(size = rel(.7), face = "bold"),
            axis.text.x = element_text(face = "bold"),
            axis.title = element_text(face = "bold"),
            legend.title = element_text(face = "bold"),
            panel.grid.major = element_blank(),
            panel.grid.minor.x = element_blank(),
            plot.title = element_text(hjust = .5))
    }

draw.a.plot("90")
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/PCC_EW_ipsative.png", width = 5,  height = 8)

#set up function to loop through the draw.a.plot() function
loop.animate <- function() {
    lapply(as.character(w1w47_subs[10:30]), function(i) {
        print(draw.a.plot(i))
    })
}

# saveGIF(loop.animate(), interval = .5, movie.name="PCC_ipsative.gif", ani.width = 300, ani.height = 450,
#         imgdir = "~/Box Sync/network/PAIRS/Brown Bag 3.31/")
```

4.4 Centrality

As with between-person effects, we can calculate centrality for individuals.

```
varnames <- varnames[-c(10:13)]
# create function to save both centrality measure and variable names to a data frame.
centrality_fun <- function(x) {
  data <- x %>%
    select(from, to, weight) %>%
    mutate(weight = as.numeric(weight))
  centrality <- centrality_auto(data.frame(data))
  df <- data.frame(var = varnames,
                   centrality$node.centrality)
}

kappa_cen_fun <- function(x){
  centrality <- centrality_auto(x)$node.centrality
  centrality$var <- varnames
  return(centrality)
}


####### PDC's #######
# calculate centrality for each subject for each wave and save them to a list #
gVAR_fit_w1 <- gVAR_fit_w1 %>%
  mutate(beta_centrality = map(beta, centrality_fun),
         kappa_centrality = map(kappa_mat, kappa_cen_fun))
gVAR_fit_w47 <- gVAR_fit_w47 %>%
  mutate(beta_centrality = map(beta, centrality_fun),
         kappa_centrality = map(kappa_mat, kappa_cen_fun))

beta_centrality <- gVAR_fit_w1 %>%
  unnest(beta_centrality)
beta_centrality <- gVAR_fit_w47 %>%
  unnest(beta_centrality) %>%
  full_join(beta_centrality) %>%
  select(-Degree) %>%
  mutate(type = "Temporal")

kappa_centrality <- gVAR_fit_w1 %>%
  unnest(kappa_centrality)
kappa_centrality <- gVAR_fit_w47 %>%
  unnest(kappa_centrality) %>%
  full_join(kappa_centrality) %>%
  select(-Degree) %>%
  mutate(type = "Contemporaneous")

save(kappa_centrality, beta_centrality,
     file = "~/Box Sync/network/PAIRS/centrality/graphicalVAR_centrality.RData")
```

4.4.1 Sample Centrality Plot

```
load("~/Box Sync/network/PAIRS/centrality/graphicalVAR_centrality.RData")

centrality_Plot_fun <- function(x, type2){
  dat <- centrality_all  %>%
    filter(SID == x, type == type2) %>%
    arrange(measure, wave)
  dat %>%
    ggplot(aes(x = var, y = z, group = wave))+
      geom_line(aes(linetype = wave), color = "black", size = .3) +
      geom_point(aes(shape = wave), size = 2) +
      labs(x = NULL, y = "z-score") +
      ylim(-2,2) +
      coord_flip() +
      labs(title = x) +
```

```
      facet_wrap(~type + measure, nrow = 1) +
      theme_bw() +
      theme(plot.title = element_text(hjust = .5),
            legend.position = "bottom")
}



centrality <- beta_centrality %>%
  gather(key = measure, value = centrality,
         Betweenness, Closeness, InStrength, OutStrength) %>%
  group_by(SID, wave, measure) %>%
  mutate(z = scale(centrality)) %>%
  ungroup() %>%
  mutate(wave = mapvalues(wave, from = c("1", "47"), to = c("1","2")))


centrality_all <- kappa_centrality %>%
  gather(key = measure, value = centrality,
         Betweenness, Closeness, Strength) %>%
  group_by(SID, wave, measure) %>%
  mutate(z = scale(centrality)) %>%
  ungroup() %>%
  mutate(wave = mapvalues(wave, from = c("1", "47"), to = c("1","2"))) %>%
  full_join(centrality)


draw.a.plot <-  function(x,y){
  centrality <- centrality_all %>%
    filter(SID == x) %>%
    arrange(measure, wave)
  centrality %>%
    filter(type == y & (measure != "Betweenness" & measure != "Closeness")) %>%
    ggplot(aes(x = var, y = z, group = wave))+
      geom_line(aes(linetype = wave), color = "black", size = .3) +
      geom_point(aes(shape = wave), size = 2) +
      labs(x = NULL, y = "z-score", title = sprintf("Subject %s", x),
           shape = "Wave", linetype = "Wave") +
      ylim(-2.5,2.5) +
      geom_hline(aes(yintercept = 0), linetype = "dashed") +
      coord_flip() +
      facet_wrap(~type + measure, nrow = 1) +
      theme_bw() +
      theme(plot.title = element_text(hjust = .5, face = "bold"),
            axis.text = element_text(face = "bold"),
            axis.title = element_text(face = "bold"),
            legend.title = element_text(face = "bold"))
  }

#set up function to loop through the draw.a.plot() function
loop.animate <- function() {
    lapply(w1w47_subs[10:30], function(i) {
        print(draw.a.plot(i, "Temporal"))
    })
}

# saveGIF(loop.animate(), interval = .5, movie.name="PDC_centrality.gif", ani.width = 400, ani.height = 350,
#          imgdir = "~/Box Sync/network/PAIRS/Brown Bag 3.31/")

loop.animate <- function() {
    lapply(w1w47_subs[10:30], function(i) {
        print(draw.a.plot(i, "Contemporaneous"))
    })
}

# saveGIF(loop.animate(), interval = .5, movie.name="PCC_centrality.gif", ani.width = 275, ani.height = 350,
#          imgdir = "~/Box Sync/network/PAIRS/Brown Bag 3.31/")
```

```r
centrality$type <- factor(centrality$type, levels = c("Temporal", "Contemporaneous"))

PCC_centralityPlot_90 <- centrality_all %>%
  filter(type == "Contemporaneous" & measure == "Strength" & SID == "90") %>%
ggplot(aes(x = var, y = z, group = wave))+
  geom_line(aes(linetype = wave), color = "black", size = .3) +
  geom_point(aes(shape = wave), size = 2) +
  labs(x = NULL, y = "z-score", linetype = "Wave", shape = "Wave") +
  scale_y_continuous(limits = c(-2.5,2.5), breaks = seq(-2.5,2.5,.5)) +
  geom_hline(aes(yintercept = 0), linetype = "dashed") +
  coord_flip() +
  facet_wrap(~type + measure, nrow = 1) +
  theme_bw() +
  theme(axis.text = element_text(face = "bold"),
        axis.title = element_text(face = "bold"),
        legend.title = element_text(face = "bold"))
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/PCC_centrality_plot_90.png", width = 4, height = 5)

PDC_centralityPlot_90 <- centrality_all %>%
  filter(type == "Temporal" & (measure != "Betweenness" & measure != "Closeness") & SID == "90") %>%
ggplot(aes(x = var, y = z, group = wave))+
  geom_line(aes(linetype = wave), color = "black", size = .3) +
  geom_point(aes(shape = wave), size = 2) +
  labs(x = NULL, y = "z-score", linetype = "Wave", shape = "Wave") +
  scale_y_continuous(limits = c(-2.5,2.5), breaks = seq(-2.5,2.5,.5)) +
  geom_hline(aes(yintercept = 0), linetype = "dashed") +
  coord_flip() +
  facet_wrap(~type + measure, nrow = 1) +
  theme_bw()+
  theme(axis.text = element_text(face = "bold"),
        axis.title = element_text(face = "bold"),
        legend.title = element_text(face = "bold"))

ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/PDC_centrality_plot_90.png", width = 6, height = 5)
```

4.4.2 Correlations Across Waves

4.4.2.1 Rank Order Correlations

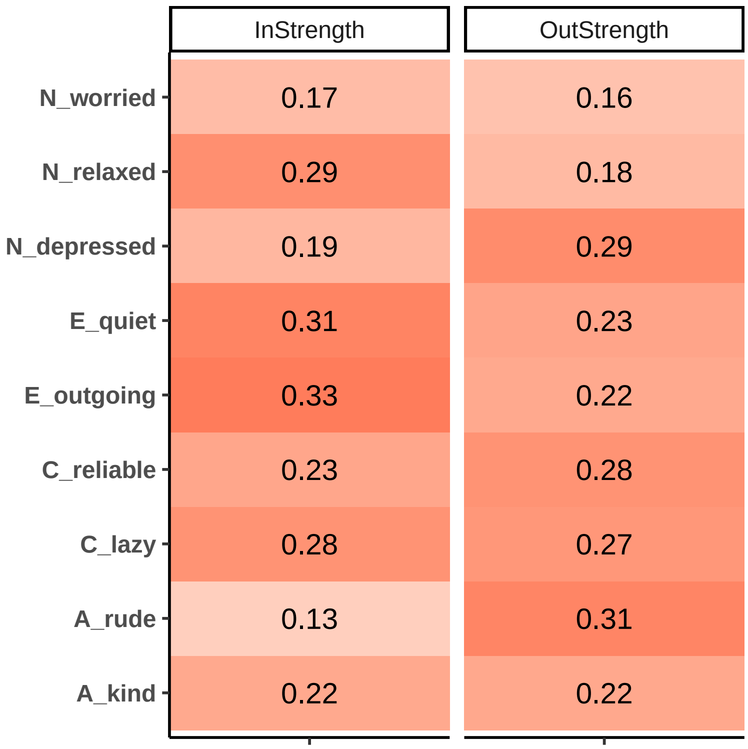
We can also look at the stability of the centrality of specific variables across the waves. That is, do variables maintain relatively the same importance in a subject's network across the waves.

4.4.2.1.1 PDC's

```r
subs_w1 <- as.character(unique(kappa_centrality$SID[kappa_centrality$wave == "1"]))
subs_w47 <- as.character(unique(kappa_centrality$SID[kappa_centrality$wave == "47"]))

w1w47_subs2 <- as.character(subs_w1[subs_w1 %in% subs_w47])

cor_fun <- function(x){
  x <- x %>% separate(measure3, into = c("Measure", "cor_type"), remove = F, "[.]")
  cor(x$`1`, x$`47`, use = "pairwise", method = "spearman")
}

beta_centrality_rank <- tbl_df(beta_centrality) %>%
  filter(SID %in% w1w47_subs2) %>%
  gather(key = measure, value = Centrality, Betweenness:OutStrength) %>%
  group_by(measure, var, type, wave) %>%
  mutate(rank = min_rank(desc(Centrality))) %>%
  ungroup() %>%
  gather(key = measure2, value = value, Centrality, rank) %>%
  unite(measure3, measure, measure2, remove = F, sep = ".")  %>%
  spread(key = wave, value = value)
```

```r
beta_centrality_cor <- beta_centrality_rank %>%
  filter(measure2 == "rank") %>%
  group_by(var, type, measure, measure2) %>%
  nest() %>%
  mutate(r = map(data, cor_fun))

beta_centrality_cor.df <- beta_centrality_cor %>%
  unnest(r, .drop = T)

kable(beta_centrality_cor.df %>%
  group_by(measure2, measure) %>%
  summarize(mean = meanSD_r2z2r(r)[1],
            sd = meanSD_r2z2r(r)[2]),
  caption = "Average Rank Order Centrality Partial Directed Correlation")
```

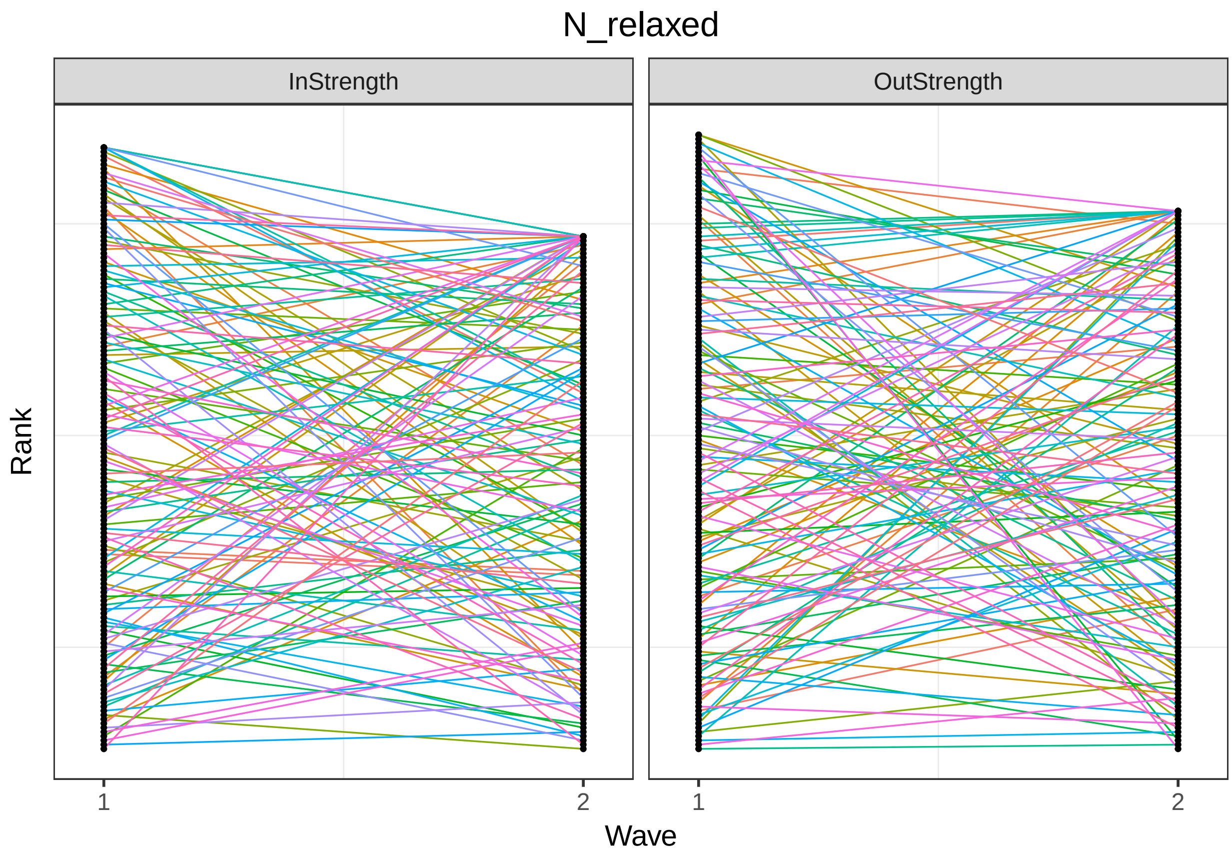
```r
beta_centrality_cor.df %>%
  filter(measure == "InStrength" | measure == "OutStrength") %>%
  select(var, measure, measure2, r) %>%
  mutate(measure2 = "Rank-Order") %>%
  ggplot(aes(x = measure, y = var, fill = r)) +
    geom_raster() +
    scale_fill_gradient2(low = "blue", high = "red", mid = "white",
     midpoint = 0, limit = c(-.5,.5), space = "Lab",
     name="Centrality\nCorrelations") +
    geom_text(aes(label = round(r,2))) +
    labs(x = NULL, y = NULL) +
    facet_grid(.~measure, scale = "free") +
    theme_classic() +
    theme(axis.text = element_text(face = "bold"), legend.position = "none",
          axis.text.x = element_blank())
```




```r
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/PDC_centrality_var_cor.png",
       width = 4, height = 4)

draw.a.plot <-  function(x){
  data <- beta_centrality %>%
    filter(var == x & SID %in% w1w47_subs) %>%
```

```r
    select(-Betweenness, -Closeness) %>%
    mutate(wave = recode(wave, `1` = "1", `47` = "2")) %>%
    gather(key = type2, value = Centrality, InStrength, OutStrength) %>%
    spread(key = wave, value = Centrality) %>%
    mutate(diff = `1` - `2`) %>%
    gather(key = wave, value = Centrality, `1`, `2`)
  max_cen <- ceiling(max(data$Centrality, na.rm = T))
  break_size <- max_cen/2
  plot <- ggplot(data, aes(x = SID, y = Centrality, group = wave)) +
    geom_hline(aes(yintercept = 0), linetype = "dashed") +
    geom_line(aes(color = wave), size = .3) +
    scale_color_manual(values = c("indianred1", "black")) +
    scale_y_continuous(limits = c(-1*max_cen, max_cen), breaks = seq(-max_cen, max_cen, break_size)) +
    #geom_point(size = .5) +
    labs(x = "Subject", y = "Centrality", color = "Wave",
         title = x) +
    coord_flip() +
    theme_bw() +
    facet_grid(.~type2) +
    theme(axis.text.y = element_blank(),
          axis.ticks.y = element_blank(),
          panel.grid.major = element_blank(),
          plot.title = element_text(hjust = .5))
  plot2 <- plot + geom_line(aes(y = diff), color = "blue", size = 1)
  return(list(plot, plot2))
  }

#set up function to loop through the draw.a.plot() function
loop.animate <- function() {
    lapply(varnames[1:9], function(i) {
        print(draw.a.plot(i))
      #ggsave(sprintf("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/%s.png", i), height = 6, width = 4)
    })
}

# saveGIF(loop.animate(), interval = .5, movie.name="PDC_centrality_var.gif", ani.width = 400, ani.height = 400)
#
##### rank order #####
```

```r
draw.a.plot <-  function(x){
  data <- beta_centrality %>%
    filter(var == x & SID %in% w1w47_subs) %>%
    select(-Betweenness, -Closeness) %>%
    mutate(wave = recode(wave, `1` = "1", `47` = "2")) %>%
    gather(key = type2, value = Centrality, InStrength, OutStrength) %>%
    group_by(wave, type2, var) %>%
    mutate(rank = dense_rank(desc(Centrality)))
  ggplot(data, aes(x = as.numeric(wave), y = rank, group = factor(SID))) +
    geom_line(aes(color = factor(SID)), size = .3) +
    geom_point(size = .5) +
    scale_x_continuous(limits = c(.95, 2.05), breaks = c(1,2)) +
    labs(x = "Wave", y = "Rank", title = x) +
    theme_bw() +
    facet_grid(.~type2, scale = "free") +
    theme(panel.grid.major = element_blank(),
          axis.text.y = element_blank(),
          axis.ticks.y = element_blank(),
          plot.title = element_text(hjust = .5),
          legend.position = "none")
}

draw.a.plot("N_relaxed")
```



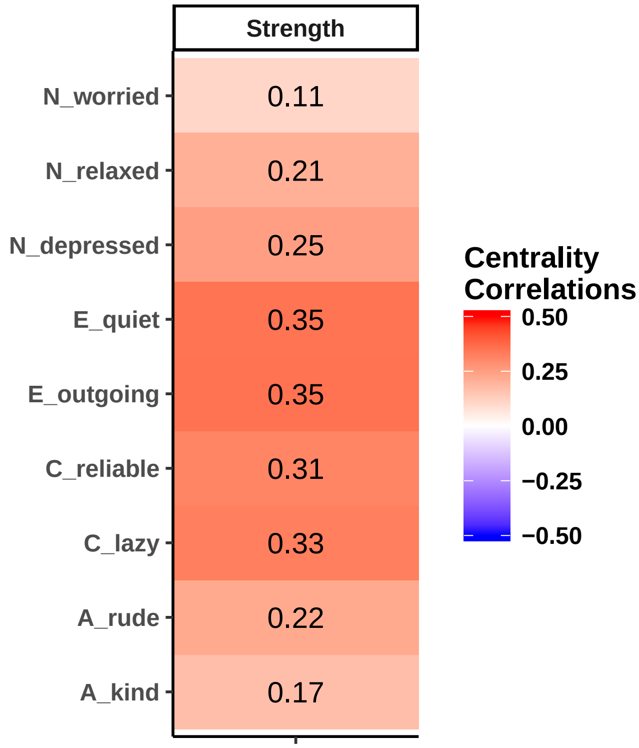

```
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/PDC_centrality_rank.png", width = 5, height = 5)
# saveGIF(loop.animate(), interval = .5, movie.name="PDC_centrality_rank.gif", ani.width = 400, ani.height = 400)
```

4.4.2.1.2 PCC's

```
kappa_centrality_cor <- tbl_df(kappa_centrality) %>%
  filter(SID %in% w1w47_subs2) %>%
  gather(key = measure, value = Centrality, Betweenness:Strength) %>%
  group_by(measure, var, type, wave) %>%
  mutate(rank = min_rank(desc(Centrality))) %>%
  ungroup() %>%
  gather(key = measure2, value = value, Centrality, rank) %>%
  unite(measure3, measure, measure2, remove = F, sep = ".") %>%
  spread(key = wave, value = value) %>%
  group_by(var, type, measure, measure2) %>%
  nest() %>%
  mutate(r = map(data, cor_fun))

kappa_centrality_cor.df <- kappa_centrality_cor %>%
  filter(measure2 == "rank") %>%
  unnest(r, .drop = T)

kable(kappa_centrality_cor.df %>%
  group_by(measure2, measure) %>%
  summarize(mean = meanSD_r2z2r(r)[1],
            sd = meanSD_r2z2r(r)[2]),
  caption = "Average Centrality Partial Contemporaneous Correlation")
```

```
kappa_centrality_cor.df %>%
  filter(measure == "Strength") %>%
  mutate(measure2 = "Rank-Order") %>%
  filter(measure2 == "Rank-Order") %>%
  ggplot(aes(x = measure, y = var, fill = r)) +
```

```r
    geom_raster() +
    scale_fill_gradient2(low = "blue", high = "red", mid = "white",
     midpoint = 0, limit = c(-.5,.5), space = "Lab",
     name="Centrality\nCorrelations") +
    geom_text(aes(label = round(r,2))) +
    labs(x = NULL, y = NULL) +
    facet_grid(.~measure) +
    theme_classic() +
    theme(axis.text = element_text(face = "bold"),
          axis.text.x = element_blank(),
          legend.text = element_text(face = "bold"),
          legend.title = element_text(face = "bold"),
          strip.text = element_text(face = "bold"))
```




```r
ggsave("~/Box Sync/network/PAIRS/Brown Bag Presentation 3.31/PCC_centrality_var_cor.png",
       width = 3.5, height = 4)

draw.a.plot <-  function(x){
  data <- kappa_centrality %>%
    select(-Betweenness, -Closeness) %>%
    filter(var == x & SID %in% w1w47_subs) %>%
    mutate(wave = recode(wave, `1` = "1", `47` = "2")) %>%
    spread(key = wave, value = Strength) %>%
    mutate(diff = `1` - `2`) %>%
    gather(key = wave, value = Strength, `1`, `2`)
  max_cen <- ceiling(max(data$Strength, na.rm = T))
  break_cen <- max_cen/2
  plot <- ggplot(data, aes(x = SID, y = Strength, group = wave)) +
    geom_line(aes(color = wave), size = .3) +
    scale_y_continuous(limits = c(-max_cen, max_cen), breaks = seq(-max_cen, max_cen, break_cen)) +
    scale_color_manual(values = c("indianred1", "black")) +
    #geom_point(size = .5) +
    labs(x = "Subject", y = "Centrality", color = "Wave",
         title = x) +
    coord_flip() +
    theme_bw() +
    theme(axis.text.y = element_blank(),
```

```
         axis.ticks.y = element_blank(),
         panel.grid.major = element_blank(),
         plot.title = element_text(hjust = .5))
  plot2 <- plot + geom_line(aes(y = diff), color = "blue", size = 1)
  return(list(plot, plot2))
  }
```

4.4.2.2 Profile Correlations by Centrality Measure

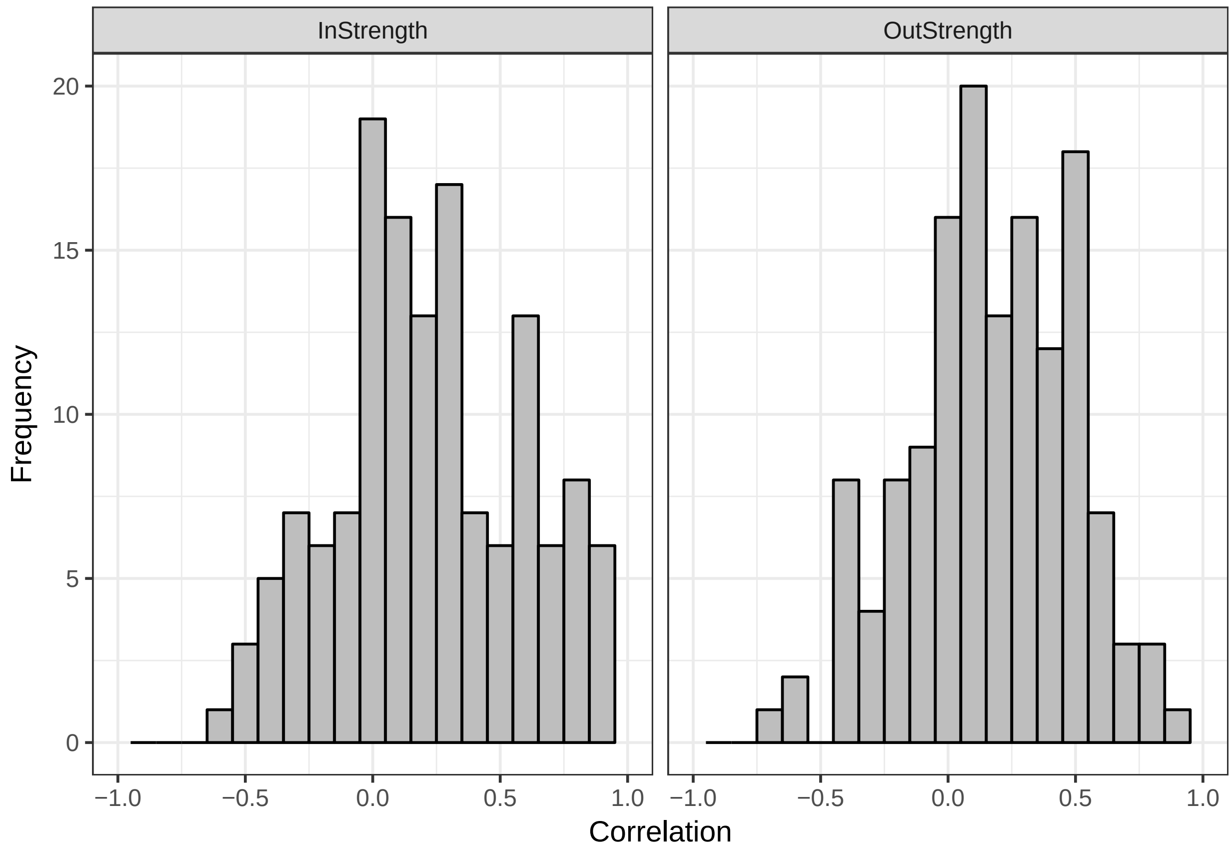
We can also examine profile correlations of the subjects across waves. That is, for a single subject, does the relative importance of a node in a network remain stable over time. We calculate a correlation for each subject and plot the results in a histogram.
#####PDCs

```
beta_profile_cors <- tbl_df(beta_centrality) %>%
  dplyr::filter(SID %in% w1w47_subs2) %>%
  gather(key = measure, value = Centrality, Betweenness:OutStrength) %>%
  spread(key = wave, value = Centrality) %>%
  arrange(SID, type, measure, var) %>%
  group_by(SID, type, measure) %>%
  nest() %>%
  mutate(r = map(data, ~cor(.$`1`, .$`47`, use = "pairwise")))

beta_profile_cors.df <- beta_profile_cors %>% unnest(r, .drop = T)

kable(beta_profile_cors.df %>%
  group_by(type, measure) %>%
  summarize(mean = meanSD_r2z2r(r)[1],
            sd = meanSD_r2z2r(r)[2]),
  caption = "Average Profile Correlation for Partial Directed Correlations")
```

Table 9: Average Profile Correlation for Partial Directed Correlations

type	measure	mean	sd
Temporal	Betweenness	0.0385562	0.3948826
Temporal	Closeness	0.1128220	0.6567782
Temporal	InStrength	0.2812713	0.4454195
Temporal	OutStrength	0.1935835	0.3622928

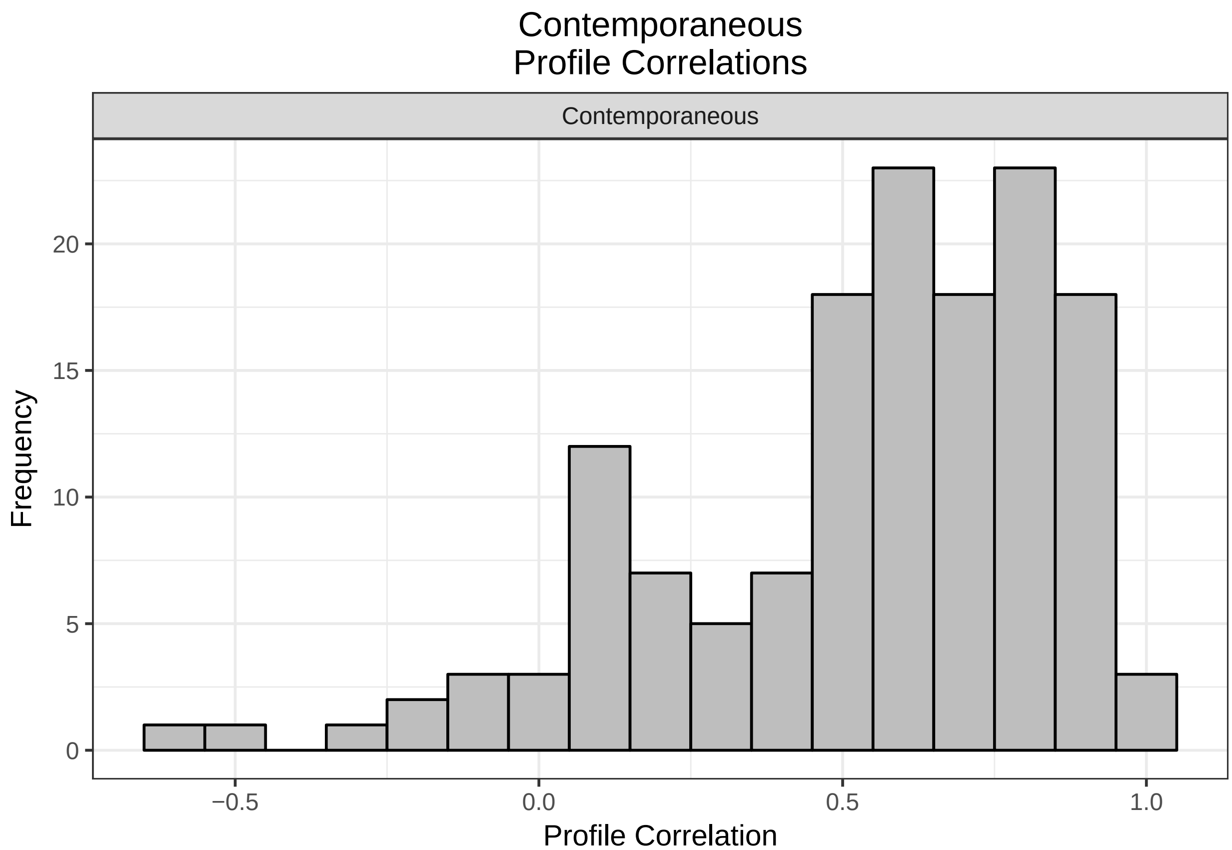
```
beta_profile_cors.df %>%
  filter(measure %in% c("InStrength", "OutStrength")) %>%
  ggplot(aes(x = r)) +
  geom_histogram(binwidth = .1, color = "black", fill = "gray") +
  xlim(-1,1) +
  labs(x = "Correlation", y = "Frequency") +
  facet_grid(.~measure) +
  theme_bw()
```




```
ggsave("~/Box Sync/network/PAIRS/manuscript/R/ipsative_centralty.png", width = 6, height = 4)
```

4.4.2.2.1 PCCs

```
kappa_profile_cors <- kappa_centrality %>%
  filter(SID %in% w1w47_subs2) %>%
  gather(key = measure, value = Centrality, Betweenness:Strength) %>%
  spread(key = wave, value = Centrality) %>%
  arrange(SID, type, measure, var) %>%
  group_by(SID, type, measure) %>%
  nest() %>%
  mutate(r = map(data, ~cor(.$`1`, .$`47`, use = "pairwise")))

kappa_profile_cors.df <- kappa_profile_cors %>% unnest(r, .drop = T)

kappa_profile_cors.df %>%
  filter(measure == "Strength") %>%
  ggplot(aes(x = r))+
  geom_histogram(binwidth = .1, color = "black", fill = "gray")+
  labs(y = "Frequency", x = "Profile Correlation",
       title = "Contemporaneous\nProfile Correlations") +
  facet_wrap(~type) +
  theme_bw() +
  theme(plot.title = element_text(hjust = .5))
```


Contemporaneous
Profile Correlations
Contemporaneous
Frequency
0
5
10
15
20
−0.5
0.0
0.5
1.0
Profile Correlation