PAIRS ESM Networks

Emorie D Beck

June 5, 2017

Contents

1 Workspace

1.1 Packages

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

```
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.

```r
#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."), esm.BH02.w1)
w4 <- dplyr::select(wave4_all, esm.IDnum.w4, esm.PRO01.w4, esm.PRO03.w4,
                    esm.PRO04.w4, esm.PRO05.w4, matches("BFI"), esm.BH02.w4)
w7 <- dplyr::select(wave7_all, esm.IDnum.w7, esm.PRO01.w7, esm.PRO03.w7,
                    esm.PRO04.w7, esm.PRO05.w7,  matches("BFI"), esm.BH02.w7)
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", "studied")

# 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", "studied")

# 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
w2 <- 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
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 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.

```
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(w2$SID)){
  mean_A_rude <- mean(w2$A_rude[w2$SID == i], na.rm = T)
  w2$A_rude[is.na(w2$A_rude) & w2$SID == i] <- mean_A_rude
  mean_A_kind <- mean(w2$A_kind[w2$SID == i], na.rm = T)
  w2$A_kind[is.na(w2$A_kind) & w2$SID == i] <- mean_A_kind
}
```

2.3 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 10 responses, lest we eliminate a large portion of our subjects.

```
# retain cases where all personality data are retained
w1_com <- w1[complete.cases(w1[,c(6:14)]),]
w2_com <- w2[complete.cases(w2[,c(6:14)]),]

# for waves 4 and 7, create a variable that combines wave and day of study
w2_com$waveDay <- paste(w2_com$Wave, w2_com$day, sep = ".")

# reorder data sets by SID, day, and block
w1_com  <- w1_com[order(w1_com$SID, w1_com$day, w1_com$hourBlock),]
w2_com <- w2_com[order(w2_com$SID, w2_com$waveDay, w2_com$hourBlock),]
```

2.4 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.

```
w1_com <- tbl_df(w1_com) %>%
  mutate(studied = as.numeric(studied)) %>%
  group_by(SID) %>%
  arrange(day, hourBlock) %>%
  mutate(beepvar3 = seq(1, n(), 1)) %>%
  group_by(SID) %>%
  mutate_each_(funs(comp = mean), vars = colnames(w1_com)[6:14]) %>%
  ungroup()

w2_com <- tbl_df(w2_com) %>%
  mutate(studied = as.numeric(studied)) %>%
  group_by(SID) %>%
  arrange(waveDay, hourBlock) %>%
  mutate(beepvar3 = seq(1, n(), 1)) %>%
  group_by(SID) %>%
  mutate_each_(funs(comp = mean), vars = colnames(w2_com)[6:14]) %>%
  ungroup()


# Make numeric subject IDs for each df because mlVAR won't run for factors #
w1_com$SID2 <- as.numeric(w1_com$SID)
w2_com$SID2 <- as.numeric(w2_com$SID)

w1_test <- w1_com %>%
  select(SID, SID2, beepvar3, A_rude:N_worried, studied) %>%
  group_by(SID) %>%
  mutate_each(funs(sd = sd(., na.rm = TRUE)), A_rude:N_worried) %>%
  mutate(count = n(), wave = "1") %>%
  filter(count > 10)
w2_test <- w2_com %>%
  select(SID, SID2, beepvar3, A_rude:N_worried, studied) %>%
  group_by(SID) %>%
```

```
  mutate_each(funs(sd = sd(., na.rm = TRUE)), A_rude:N_worried) %>%
  mutate(count = n(), wave = "2") %>%
  filter(count > 10)

for(i in 1:9){
  for(k in 1:dim(w1_test)[1]){
    if(w1_test[k, i + 13] == 0){
      w1_test[k, i + 3] <-
        jitter(as.numeric(w1_test[k, i + 3]), amount = runif(1, 0, .05))
    }
  }
}

for(i in 1:9){
  for(k in 1:dim(w2_test)[1]){
    if(w2_test[k, i + 13] == 0){
      w2_test[k, i + 3] <-
        jitter(as.numeric(w2_test[k, i + 3]), amount = runif(1, 0, .05))
    }
  }
}


keys <- c(-1,-1,-1,-1,1,1,1,1,1)
w1_composites <- w1_test %>%
  select(SID, beepvar3, A_rude:N_worried) %>%
  mutate_each(funs(reverse.code(-1,.,mini = 1, maxi = 5)), A_rude:N_relaxed) %>%
  gather(key = item, value = rating, A_rude:N_worried) %>%
  separate(item, c("trait", "item")) %>%
  group_by(SID, trait) %>%
  summarize(composite = mean(rating, na.rm = T)) %>%
  mutate(trait = mapvalues(trait, unique(trait), paste("ESM", c("Agreeableness", "Conscientiousness", "N
         wave = "1") %>%
  spread(key = trait, value = composite)
```

2.5 Outcomes Data

2.5.1 Read in, Clean, and Create Composites for Outcomes Data

```
# read outcome variables from wave 1 #
target.ratings.initial.w1 <- read.csv("~/Box Sync/network/PAIRS/Wave 1/target_w1_RENAMED.csv")
target.ratings.initial.w4 <- read.csv("~/Box Sync/network/PAIRS/Wave 4/home_w4_RENAMED.csv")

# get column names of target subjects #
names.w1 <- colnames(target.ratings.initial.w1)
names.w4 <- colnames(target.ratings.initial.w4)

# replace subject ID variable to match centrality data #
names.w1[2] <- "SID"; colnames(target.ratings.initial.w1) <- names.w1
names.w4[2] <- "SID"; colnames(target.ratings.initial.w4) <- names.w4

target.ratings.initial.w1$wave <- "1"
target.ratings.initial.w4$wave <- "2"

short.names <- c("SID", "wave", "life_sat", "GPA")

target.ratings.w1 <- target.ratings.initial.w1 %>%
  select(SID, wave, ts.NQ34.w1, ts.AGQ03.w1, ts.BQ13.w1) %>%
  setNames(c(short.names, "procrastinate"))
target.ratings.w4 <- target.ratings.initial.w4 %>%
  select(SID, wave, ts.NQ34.w4, ts.AGQ03.w4) %>%
  setNames(short.names)

target.ratings.w4$GPA <- as.numeric(gsub(",", ".", target.ratings.w4$GPA))
```

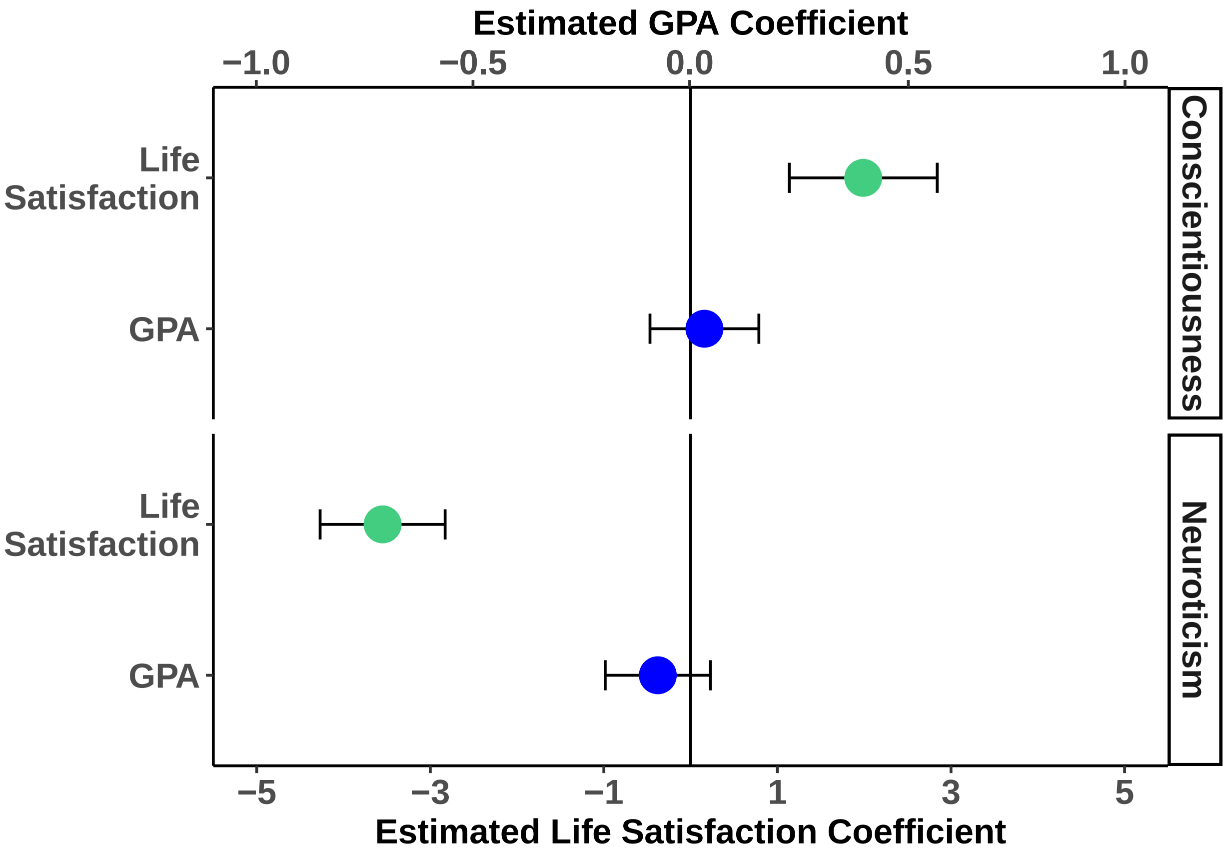
```
target.ratings <- tbl_df(target.ratings.w1) %>%
  full_join(target.ratings.w4) %>%
  mutate(SID = as.character(SID))
```

You#Question 1: Networks and the Prediction of Momentary Behaviors

3 Question 0: Do Traits Predict Outcomes?

```
tidy_mod_fun <- function(model){
  tidy(model) %>%
    bind_cols(tbl_df(confint(model, method = "boot")))
}

trait_fits <- w1_composites %>%
  left_join(select(target.ratings, SID, wave, GPA, procrastinate, life_sat)) %>%
  ungroup() %>%
  gather(key = outcome, value = value, GPA:life_sat) %>%
  gather(key = trait, value = value2, ESM_Agreeableness:ESM_Neuroticism) %>%
  group_by(wave, outcome, trait) %>%
  nest() %>%
  mutate(model = map(data, possibly(~lm(value ~ value2, data = .), NA_real_)),
         tidy = map(model, possibly(tidy_mod_fun, NA_real_)))

trait.dat <- trait_fits %>%
  filter(!is.na(model) & wave == "1" & outcome != "procrastinate" ) %>%
  unnest(tidy, .drop = T) %>%
  filter(term != "(Intercept)") %>%
  mutate(trait = mapvalues(trait, unique(trait),
                c("Extraversion", "Agreeableness","Conscientiousness", "Neuroticism")),
         outcome = recode(outcome, `life_sat` = "Life\nSatisfaction"),
         outcome = factor(outcome, levels = rev(unique(outcome)))) %>%
  filter(trait %in% c("Conscientiousness", "Neuroticism"))

trait.dat %>%
  ggplot(aes(x = outcome, y = estimate)) +
    scale_color_manual(values = c("blue", "seagreen3")) +
    geom_errorbar(data = filter(trait.dat, outcome == "GPA"), width = .2,
                  aes(ymin = `2.5 %`*5, ymax = `97.5 %`*5), position = "dodge") +
    geom_errorbar(data = filter(trait.dat, outcome == "Life\nSatisfaction"),
                  aes(ymin = `2.5 %`, ymax = `97.5 %`), width = .2, position = "dodge") +
    geom_hline(yintercept = 0) +
    geom_point(data = filter(trait.dat, outcome == "GPA"),
               aes(y = estimate*5, color = outcome), size = 6) +
  geom_point(data = filter(trait.dat, outcome == "Life\nSatisfaction"),
             aes(y = estimate, color = outcome), size = 6) +
    scale_y_continuous(limits = c(-5,5), breaks = seq(-5,5,2),
                       sec.axis = sec_axis(~./5, name = "Estimated GPA Coefficient")) +
    labs(y = "Estimated Life Satisfaction Coefficient", x = NULL) +
    coord_flip() +
    facet_grid(trait~.) +
    theme_classic() +
    theme(legend.position = "none",
          axis.text = element_text(face = "bold", size = rel(1.2)),
```

```r
        axis.title = element_text(face = "bold", size = rel(1.2)),
        strip.text = element_text(face = "bold", size = rel(1.2)))
```




```r
ggsave(file = "~/Box Sync/network/PAIRS/outcomes/graphs/trait_bs2.png", width = 5, height = 5)
```

4 Question1: Does State Personality Predict Outcomes

4.1 Idiographic Predictive Networks

```r
gVAR_fun <- function(x, outcome = NULL, SID, wave){
  print(sprintf("Wave %s: S%s", wave, SID))
  colnum <- which(colnames(x) %in% outcome)
  n <- dim(x)[1]
  x <- x %>%
    arrange(beepvar3) %>%
    select(A_rude:N_worried, colnum)
  gamma <- 0
  lambda <- seq(.025, .25, .025)
  fit <-
    graphicalVAR(x, gamma = gamma, maxit.in = 1000, maxit.out = 1000,
                       lambda_beta = lambda, lambda_kappa = lambda,
                       verbose = T, scale = F, centerWithin = F)
  return(fit)
}

gVAR_fit <- w1_test %>%
  select(-contains("sd")) %>%
  full_join(select(w2_test, -contains("sd"))) %>%
  filter(!(SID %in% c("10516", "10204", "10322"))) %>% #program bug, these freeze R
  group_by(SID, wave, count) %>%
  nest()

gVAR_fit <- gVAR_fit %>%
  mutate(gVAR_fit0 = pmap(list(x = data, SID = SID, wave = wave), possibly(gVAR_fun, NA_real_)),
         gVAR_fit1 = pmap(list(x = data, outcome = "studied", SID = SID, wave = wave),
                          possibly(gVAR_fun, NA_real_)))
save(gVAR_fit, file = "~/Box Sync/network/PAIRS/outcomes/gVAR_fit.RData")
```

4.1.1 Extract Results and Save Into Data Frames

```
beta_fun <- function(fit, SID){
  PDC <- fit$PDC
  from <- row.names(PDC)
  nvar <- length(from)
  PDC.long <- tbl_df(PDC) %>%
    mutate(from = from,
           type = "Temporal") %>%
    gather(key = to, value = value, 1:nvar) %>%
    mutate(sign = ifelse(value < 0, -1, ifelse(value > 0, 1, 0))) %>%
    unite(var, from, to, sep = ".", remove = F)
}

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

kappa_long_fun <- function(fit){
  PCC <- fit$PCC
  PCC <- PCC[,order(colnames(PCC))]
  PCC <- PCC[order(rownames(PCC)),]
  PCC[lower.tri(PCC, diag = T)] <- NA
  vars <- rownames(PCC)
  nvar <- length(vars)
  PCC.long <- tbl_df(PCC) %>%
    mutate(Var1 = vars,
           type = "Contemporaneous") %>%
    gather(key = Var2, value = value, 1:nvar) %>%
    filter(!is.na(value)) %>%
    mutate(sign = ifelse(value < 0, -1, 1)) %>%
    unite(var, Var1, Var2, sep = ".", remove = F)
}

gVAR_fit <- gVAR_fit %>%
  #filter(!is.na(gVAR_fit)) %>%
  mutate(beta0 = map2(gVAR_fit0, SID, possibly(beta_fun, NA_real_)),
         beta1 = map2(gVAR_fit1, SID, possibly(beta_fun, NA_real_)),
         kappa_mat0 = map(gVAR_fit0, possibly(kappa_mat_fun, NA_real_)),
         kappa_mat1 = map(gVAR_fit1, possibly(kappa_mat_fun, NA_real_)),
         kappa0 = map(gVAR_fit0, possibly(kappa_long_fun, NA_real_)),
         kappa1 = map(gVAR_fit0, possibly(kappa_long_fun, NA_real_)))

beta_long0 <- gVAR_fit %>% filter(!is.na(beta0)) %>% unnest(beta0)
beta_long1 <- gVAR_fit %>% filter(!is.na(beta1)) %>% unnest(beta1)

kappa_long0 <- gVAR_fit %>% filter(!is.na(kappa0)) %>% unnest(kappa0)
kappa_long1 <- gVAR_fit %>% filter(!is.na(kappa1)) %>% unnest(kappa1)

# save fit information
beta_long0$fit <- "personality"
kappa_long0$fit <- "personality"
beta_long1$fit <- "studied"
kappa_long1$fit <- "studied"

#get variable names from models
varnames_fit0 <- row.names(gVAR_fit$gVAR_fit0[[1]]$beta)
varnames_fit1 <- row.names(gVAR_fit$gVAR_fit1[[1]]$beta)
# extract PCC information
PCC_fit0 <- gVAR_fit$kappa_mat0; names(PCC_fit0) <- gVAR_fit$SID
PCC_fit1 <- gVAR_fit$kappa_mat1; names(PCC_fit1) <- gVAR_fit$SID
```

4.1.2 Plots

4.2 Split Half Networks (Reliability Check)

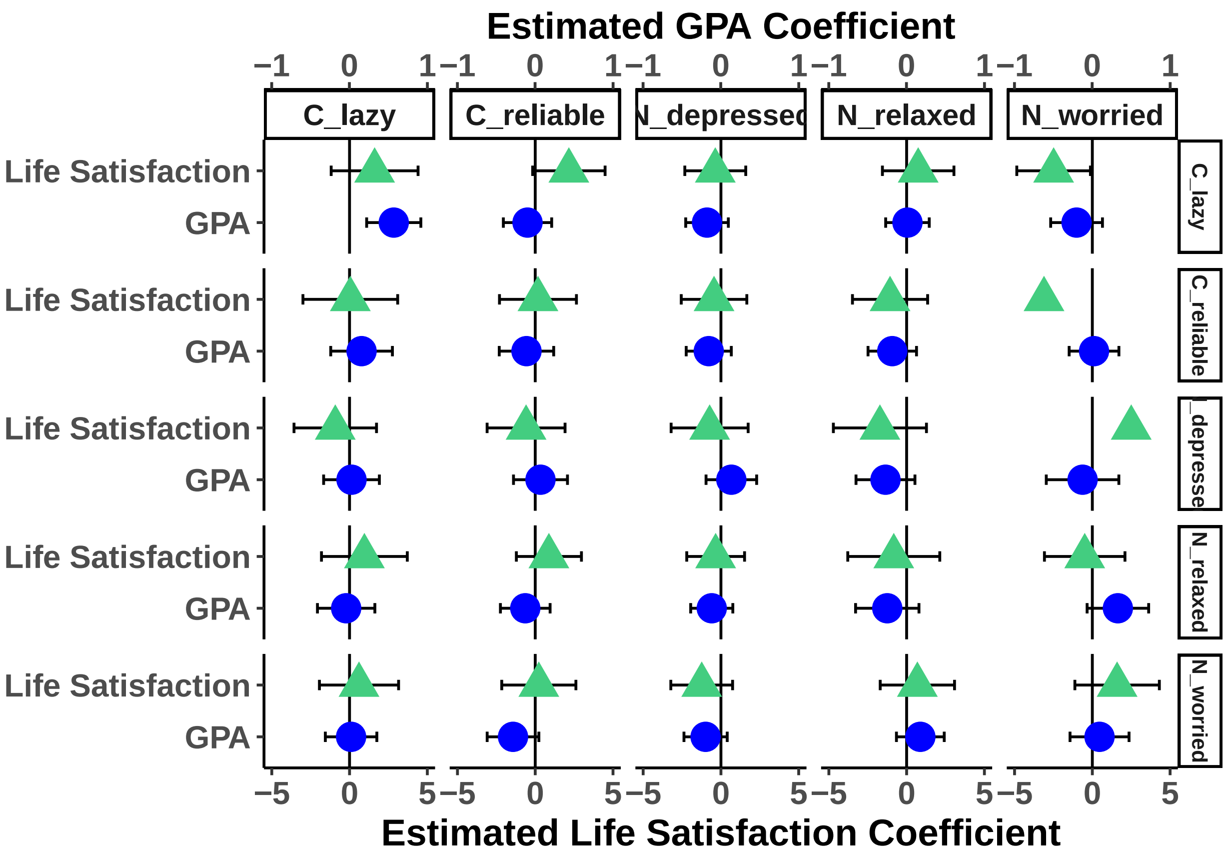
To test the reliability of the networks, we split each person's responses in half and calculate a network of eah and then compare the two using profile correlations.

```
gVAR_fun <- function(x, outcome = NULL, SID, wave){
  print(sprintf("Wave %s: S%s", wave, SID))
  colnum <- which(colnames(x) %in% outcome)
  n <- dim(x)[1]
  x <- x %>%
    arrange(beepvar3) %>%
    select(A_rude:N_worried, colnum)
  gamma <- 0
  lambda <- seq(.025, .25, .025)
  fit <-
    graphicalVAR(x, gamma = gamma, maxit.in = 1000, maxit.out = 1000,
                 lambda_beta = lambda, lambda_kappa = lambda,
                 verbose = T, scale = F, centerWithin = F)
  return(fit)
}

gVAR_fit_split <- w1_test %>%
  select(-contains("sd")) %>%
  full_join(select(w2_test, -contains("sd"))) %>% ungroup() %>%
  mutate(split = ifelse(count %% 2 == 0, count/2, count%/%2),
         split = ifelse(beepvar3 <= split, 1, 2)) %>%
  filter(!((wave == "1" & SID %in% c("68", "10148", "10160", "10455", "10187",
                           "10273","10428", "10259", "10304", "10345", "10512")) |
             (wave == "2" & SID %in% c("12", "87", "102", "10248", "10280",
                                       "10287", "10444")))) %>%
  group_by(SID, wave, count, split) %>%
  nest()

gVAR_fit_split <- gVAR_fit_split %>%
  mutate(gVAR_fit0 = pmap(list(x = data, SID = SID, wave = wave), possibly(gVAR_fun, NA_real_)))
save(gVAR_fit, gVAR_fit_split, file = "~/Box Sync/network/PAIRS/outcomes/gVAR_fit.RData")
```

4.3 Regressions

4.3.1 Temporal

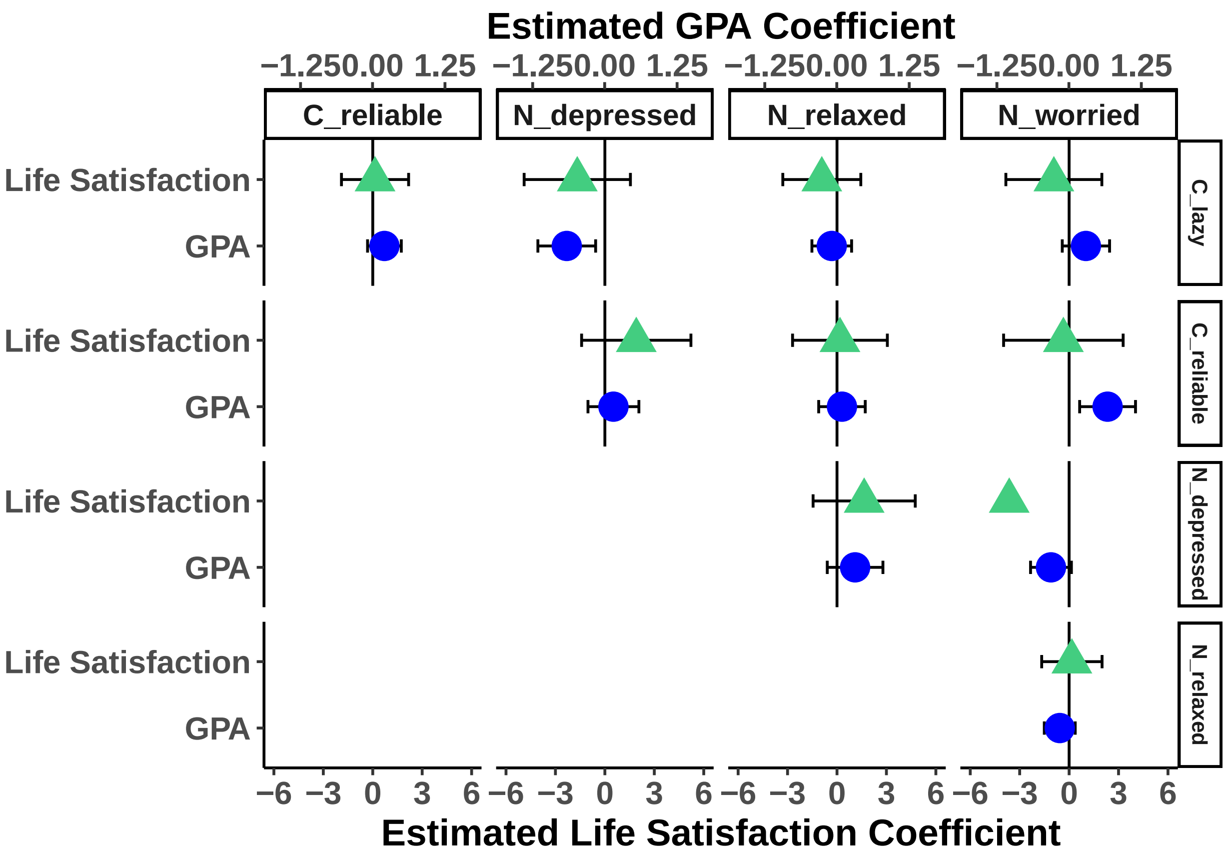
```
beta <- tbl_df(beta_long0) %>%
  full_join(beta_long1) %>%
  left_join(select(target.ratings, SID, wave,
                   GPA, procrastinate, life_sat)) %>%
  left_join(w1_composites) %>%
  filter(!is.na(value))

beta_long <- beta %>%
  select(SID:fit, GPA, procrastinate, life_sat, ESM_Conscientiousness, ESM_Neuroticism) %>%
  gather(key = outcome, value = value2, GPA:life_sat) %>%
  mutate(Edge_Trait1 = ifelse(grepl("C_",from) == T, "C",
                       ifelse(grepl("N_", from) == T, "N", NA)),
         Edge_Trait2 = ifelse(grepl("C_",to) == T, "C",
                       ifelse(grepl("N_", to) == T, "N", NA))) %>%
  filter((Edge_Trait1 == "C" & Edge_Trait2 == "C") |
         (Edge_Trait1 == "N" & Edge_Trait2 == "N") |
         (Edge_Trait1 == "C" & Edge_Trait2 == "N") |
         (Edge_Trait1 == "N" & Edge_Trait2 == "C"))

PDC_model <- function(df){
    mod0 <- lm(value2 ~ ESM_Neuroticism, data = df)
    mod1 <- lm(value2 ~ value, data = df)
    modC <- lm(value2 ~ value + ESM_Conscientiousness, data = df)
    modN <- lm(value2 ~ value + ESM_Neuroticism, data = df)
    modT <- lm(value2 ~ value + ESM_Conscientiousness + ESM_Neuroticism, data = df)
    results <- list(mod0, mod1, modC, modN, modT)
    return(results)
}

PDC_tidy <- function(mod_list){
  tidy_mod0 <- tidy(mod_list[[1]]) %>% mutate(covar = "Nonly") %>%
    bind_cols(tbl_df(confint(mod_list[[1]], method = "boot")))
  tidy_mod1 <- tidy(mod_list[[2]]) %>% mutate(covar = "EW") %>%
    bind_cols(tbl_df(confint(mod_list[[2]], method = "boot")))
  tidy_modC <- tidy(mod_list[[3]]) %>% mutate(covar = "C") %>%
    bind_cols(tbl_df(confint(mod_list[[3]], method = "boot")))
  tidy_modN <- tidy(mod_list[[4]]) %>% mutate(covar = "N") %>%
    bind_cols(tbl_df(confint(mod_list[[4]], method = "boot")))
  tidy_modT <- tidy(mod_list[[5]]) %>% mutate(covar = "C+N") %>%
    bind_cols(tbl_df(confint(mod_list[[5]], method = "boot")))
  tidy_mods <- tidy_mod1 %>%
    full_join(tidy_modC) %>%
    full_join(tidy_modN) %>%
    full_join(tidy_modT) %>%
    full_join(tidy_mod0)
  return(tidy_mods)
}

PDC_fits <- beta_long %>%
  group_by(outcome, wave, fit, var) %>%
  nest() %>%
  mutate(model = map(data, possibly(PDC_model, NA_real_)),
         tidy = map(model, possibly(PDC_tidy, NA_real_)))

PDC.dat <- PDC_fits %>%
  filter(!is.na(tidy)) %>%
  unnest(tidy) %>%
  filter(fit == "studied" & term == "value" & covar == "EW" & outcome != "procrastinate") %>%
  separate(var, into = c("from", "to"), sep = "[.]") %>%
  mutate(outcome = recode(outcome, `life_sat` = "Life Satisfaction")) %>%
  unite(comb, outcome, wave, sep = ": ", remove = F) %>%
  mutate(comb = factor(comb, levels = rev(unique(comb))),
         outcome = factor(outcome, levels = rev(unique(outcome))))

PDC.dat %>%
  ggplot(aes(x = outcome, y = estimate)) +
    geom_errorbar(data = filter(PDC.dat, outcome == "GPA"), width = .2,
                  aes(ymin = `2.5 %`*5, ymax = `97.5 %`*5), position = "dodge") +
    geom_errorbar(data = filter(PDC.dat, outcome != "GPA"),
                  aes(ymin = `2.5 %`, ymax = `97.5 %`), position = "dodge", width = .2) +
    geom_hline(aes(yintercept = 0)) +
    geom_point(data = filter(PDC.dat, outcome == "GPA"),
               aes(y = estimate * 5, color = outcome, shape = outcome), size = 5) +
    geom_point(data = filter(PDC.dat, outcome != "GPA"),
               aes(color = outcome, shape = outcome), size = 5) +
    scale_color_manual(values = c("blue", "seagreen3")) +
    scale_y_continuous(limits = c(-5,5), breaks = seq(-5, 5, 5),
                       sec.axis = sec_axis(~./5, name = "Estimated GPA Coefficient",
                                           breaks = seq(-1,1,1))) +
    #scale_color_manual(values = c("red", "blue", "green")) +
    labs(y = "Estimated Life Satisfaction Coefficient", x = NULL) +
    coord_flip() +
    facet_grid(from~to) +
    theme_classic() +
    theme(legend.position = "none",
          axis.text = element_text(face = "bold", size = rel(1.1)),
          axis.title = element_text(face = "bold", size = rel(1.3)),
          strip.text.x = element_text(face = "bold", size = rel(1.3)),
          strip.text.y = element_text(face = "bold", size = rel(.9)))
```




```
ggsave(file = "~/Box Sync/network/PAIRS/outcomes/graphs/temp_ew_bs2.png", width = 8, height = 5)
```

4.3.2 Contemporaneous

```r
PCC <- tbl_df(kappa_long0) %>%
  full_join(kappa_long1) %>%
  left_join(select(target.ratings, SID, wave, GPA, procrastinate, life_sat)) %>%
  left_join(w1_composites) %>%
  filter(!is.na(value))

PCC_long <- PCC %>%
  select(SID:fit, GPA, procrastinate, life_sat, ESM_Conscientiousness, ESM_Neuroticism) %>%
  gather(key = outcome, value = value2, GPA:life_sat) %>%
   mutate(Edge_Trait1 = ifelse(grepl("C_",Var1) == T, "C",
                        ifelse(grepl("N_", Var1) == T, "N", NA)),
         Edge_Trait2 = ifelse(grepl("C_",Var2) == T, "C",
                       ifelse(grepl("N_", Var2) == T, "N", NA))) %>%
  filter((Edge_Trait1 == "C" & Edge_Trait2 == "C") |
         (Edge_Trait1 == "N" & Edge_Trait2 == "N") |
         (Edge_Trait1 == "C" & Edge_Trait2 == "N") |
         (Edge_Trait1 == "N" & Edge_Trait2 == "C"))

PCC_fits <- PCC_long %>%
  group_by(outcome, wave, fit, var) %>%
  nest() %>%
  mutate(model = map(data, possibly(PDC_model, NA_real_)),
         tidy = map(model, possibly(PDC_tidy, NA_real_)))

PCC.dat <- PCC_fits %>%
  filter(!is.na(tidy)) %>%
  unnest(tidy) %>%
  filter(fit == "personality" & term == "value" & covar == "EW" &
           outcome != "procrastinate" & wave == "1") %>%
  separate(var, into = c("from", "to"), sep = "[.]") %>%
  mutate(outcome = recode(outcome, `life_sat` = "Life Satisfaction")) %>%
  unite(comb, outcome, wave, sep = ": ", remove = F) %>%
  mutate(comb = factor(comb, levels = rev(unique(comb))),
         outcome = factor(outcome, levels = rev(unique(outcome))))

PCC.dat %>%
  ggplot(aes(x = outcome, y = estimate)) +
    geom_errorbar(data = filter(PCC.dat, outcome == "GPA"), width = .2,
                  aes(ymin = `2.5 %`*3.5, ymax = `97.5 %`*3.5), position = "dodge") +
    geom_errorbar(data = filter(PCC.dat, outcome != "GPA"),
                  aes(ymin = `2.5 %`, ymax = `97.5 %`), position = "dodge", width = .2) +
    geom_hline(aes(yintercept = 0)) +
    geom_point(data = filter(PCC.dat, outcome == "GPA"),
               aes(y = estimate * 3.5, color = outcome, shape = outcome), size = 5) +
    geom_point(data = filter(PCC.dat, outcome != "GPA"),
               aes(color = outcome, shape = outcome), size = 5) +
    scale_y_continuous(limits = c(-6,6), breaks = seq(-6, 6, 3),
                       sec.axis = sec_axis(~./3.5, name = "Estimated GPA Coefficient",
                                           breaks = seq(-1.25,1.25,1.25))) +
    scale_color_manual(values = c("blue", "seagreen3")) +
    labs(y = "Estimated Life Satisfaction Coefficient", x = NULL) +
    coord_flip() +
    facet_grid(from~to) +
    theme_classic() +
    theme(legend.position = "none",
          axis.text = element_text(face = "bold", size = rel(1.1)),
          axis.title = element_text(face = "bold", size = rel(1.3)),
          strip.text.x = element_text(face = "bold", size = rel(1.3)),
          strip.text.y = element_text(face = "bold", size = rel(.9)))
```




```
ggsave(file = "~/Box Sync/network/PAIRS/outcomes/graphs/contemp_ew_bs2.png", width = 8, height = 5)
```

5 Question 2: Does centrality relate to outcomes?

5.1 Run Centrality Analyses

5.1.1 Idiographic

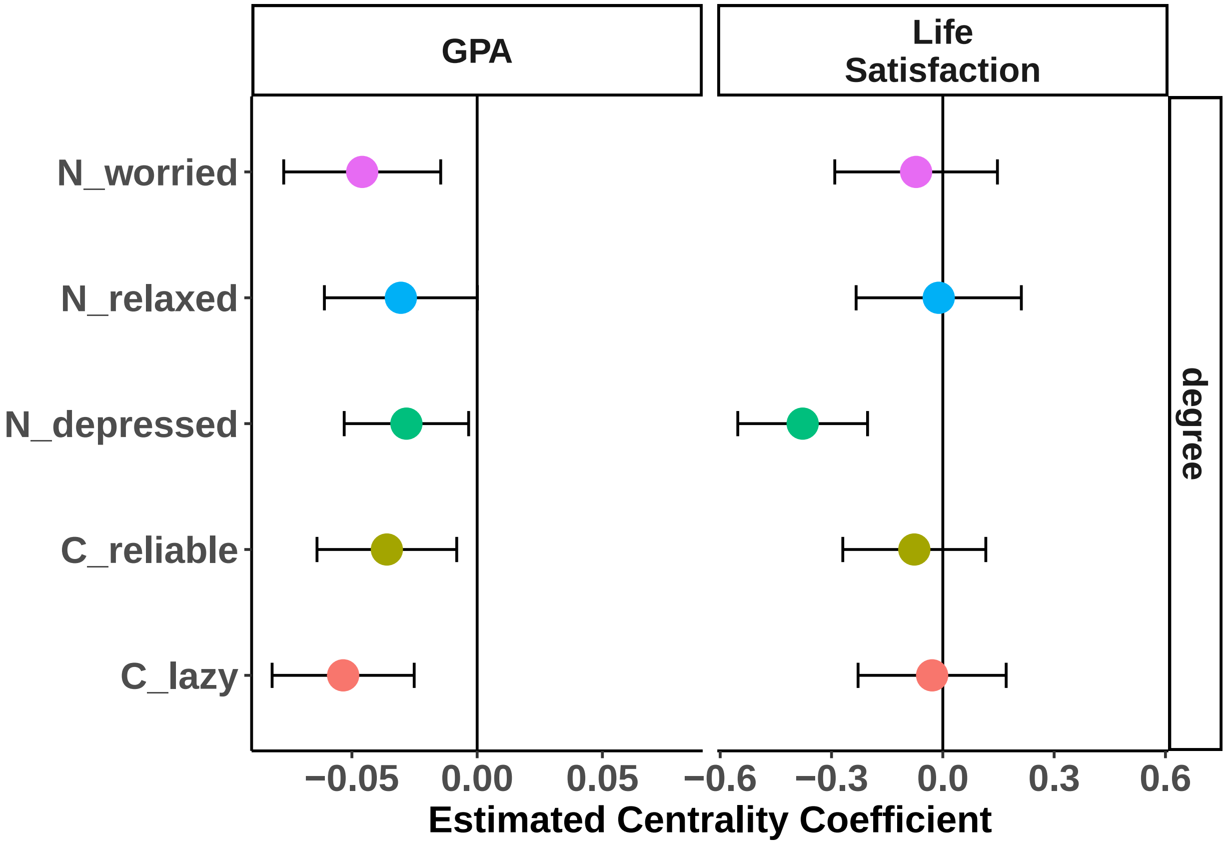
```
# create function to save both centrality measure and variable names to a data frame.
centralityList <- function(x, type, result, gVAR) {
  if(type == "temporal"){
      centrality <- centrality_auto(data.frame(select(x, from, to, value)))
      mat <- gVAR$PDC
      diag(mat) <- 0
      varnames <- as.character(unique(x$from))
      degree <- tibble(var = row.names(mat),
                       InDegree = colSums(mat != 0),
                       OutDegree = rowSums(mat != 0))
      spl <- centrality$ShortestPathLengths}
    else{
      centrality <- centrality_auto(x)
      mat <- gVAR$PCC
      diag(mat) <- 0
      degree <- tibble(var = rownames(x),
                       degree = colSums(mat != 0))
      varnames <- rownames(x)
      spl <- centrality$ShortestPathLengths
      spl <- spl[,order(colnames(spl))]
      spl <- spl[order(rownames(spl)),]
      spl[upper.tri(spl, diag = T)] <- NA}
  if(result == "centrality"){
    df <- tbl_df(centrality$node.centrality %>% mutate(var = varnames)) %>%
      full_join(degree)
  } else{
    if(type == "temporal"){
    df <- tbl_df(spl) %>%
       mutate(from = colnames(.)) %>%
       gather(key = to, value = value, 1:length(varnames)) %>%
       mutate(value = ifelse(is.infinite(value) == T, NA, value))
    } else {
      df <- tbl_df(spl) %>%
       mutate(Var2 = colnames(.)) %>%
       gather(key = Var1, value = value, 1:length(varnames)) %>%
       filter(!is.na(value)) %>%
       mutate(value = ifelse(is.infinite(value) == T, NA, value))
```

```r
    }
  }
  return(df)
}

gVAR_fit <- gVAR_fit %>%
  mutate(beta_centrality0 = map2(beta0, gVAR_fit0, possibly(~centralityList(.x,
                                      "temporal", "centrality", .y), NA_real_)),
         beta_centrality1 = map2(beta1, gVAR_fit1, possibly(~centralityList(.x,
                                      "temporal", "centrality", .y), NA_real_)),
         kappa_centrality0 = map2(kappa_mat0, gVAR_fit0, possibly(~centralityList(.x,
                                      "contemporaneous", "centrality", .y), NA_real_)),
         kappa_centrality1 = map2(kappa_mat1, gVAR_fit1, possibly(~centralityList(.x,
                                      "contemporaneous", "centrality", .y), NA_real_)))

# let's reshape and merge the results within models
centrality_fit0_PDC_long <- gVAR_fit %>%
  filter(!is.na(beta_centrality0)) %>%
  unnest(beta_centrality0, .drop = T) %>%
  mutate(fit = "personality", type = "Temporal")

centrality_fit1_PDC_long <- gVAR_fit %>%
  filter(!is.na(beta_centrality1)) %>%
  unnest(beta_centrality1, .drop = T) %>%
  mutate(fit = "studied", type = "Temporal")

centrality_fit0_PCC_long <- gVAR_fit %>%
  filter(!is.na(kappa_centrality0)) %>%
  unnest(kappa_centrality0, .drop = T) %>%
  mutate(fit = "personality", type = "Contemporaneous")

centrality_fit1_PCC_long <- gVAR_fit %>%
  filter(!is.na(kappa_centrality1)) %>%
  unnest(kappa_centrality1, .drop = T) %>%
  mutate(fit = "studied", type = "Contemporaneous")

# let's reshape and merge the results across models
centrality <- centrality_fit0_PDC_long %>%
  full_join(centrality_fit1_PDC_long) %>%
  full_join(centrality_fit0_PCC_long) %>%
  full_join(centrality_fit1_PCC_long) %>%
  gather(key = measure, value = value, Betweenness:OutStrength,
         Strength, InDegree, OutDegree, degree) %>%
  filter(!is.na(value)) %>%
  group_by(fit, wave, type, SID, measure) %>%
  mutate(z = as.numeric(scale(value))) %>%
  ungroup()

# all the centrality ratings with rankings + outcomes
centrality5 <- centrality %>%
  left_join(target.ratings) %>%
  left_join(w1_composites) %>%
  filter(!is.na(fit)) %>%
  mutate()

subsw1 <- (gVAR_fit %>% filter(!is.na(gVAR_fit0) & !is.na(beta_centrality0) &
                               !is.na(kappa_centrality0) & wave == "1"))$SID
centrality_plot_fun <- function(sub, Type){
  print(sub)
centrality %>%
  filter(wave == "1" & fit == "personality" & SID == sub & type == Type & grepl("egree", measure) == T) %>%
    ggplot(aes(x = var, y = z, group = measure)) +
      geom_point() +
      geom_line() +
      geom_hline(aes(yintercept = 0), linetype = "dashed") +
      scale_y_continuous(limits = c(-3,3), breaks = seq(-3,3,1)) +
      labs(x = NULL, title = sprintf("Subject %s", sub)) +
      coord_flip() +
      facet_grid(.~measure) +
      theme_bw() +
      theme(axis.text = element_text(face = "bold", size = rel(1.2)),
            axis.title = element_text(face = "bold", size = rel(1.2)),
            strip.text = element_text(face = "bold", size = rel(1.3)),
            plot.title = element_text(face = "bold", size = rel(1.3),
                                      hjust = .5))
}

library(animation)
#set up function to loop through the draw.a.plot() function #
# temporal network centrality #
loop.animate <- function() {
    lapply(subsw1[1:50], function(i) {
        print(centrality_plot_fun(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(subsw1[1:50], function(i) {
        print(centrality_plot_fun(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/")
```

5.2 Regressions

5.2.1 Centrality Indices

```
centrality_model <- function(df){
    mod0 <- lm(value2 ~ ESM_Neuroticism, data = df)
    mod1 <- lm(value2 ~ value, data = df)
    modC <- lm(value2 ~ value + ESM_Conscientiousness, data = df)
    modN <- lm(value2 ~ value + ESM_Neuroticism, data = df)
    modT <- lm(value2 ~ value + ESM_Conscientiousness + ESM_Neuroticism, data = df)
    results <- list(mod0, mod1, modC, modN, modT)
    return(results)
}

centrality_long <- centrality5 %>%
  select(SID:measure, value, z, GPA, procrastinate, life_sat,
         ESM_Neuroticism, ESM_Conscientiousness) %>%
  gather(key = outcome, value = value2, GPA:life_sat) %>%
  mutate(Edge_Trait1 = ifelse(grepl("C_",var) == T, "C",
                       ifelse(grepl("N_", var) == T, "N", NA))) %>%
  filter(Edge_Trait1 == "C" | Edge_Trait1 == "N") %>%
  mutate(model_type = ifelse(outcome == "rel_stat", "glm", "lm"))

centrality_fits <- centrality_long %>%
  group_by(outcome, wave, fit, var, measure, type) %>%
  nest() %>%
  mutate(model = map(data, possibly(centrality_model, NA_real_)),
         tidy = map(model, possibly(PDC_tidy, NA_real_))) %>%
  filter(!is.na(model))

centrality_fits %>%
  filter(!is.na(tidy)) %>%
  unnest(tidy) %>%
  filter(fit == "personality" & (measure %in% c("degree", "InDegree", "OutDegree")) &
           term == "value" & covar == "EW" & type == "Contemporaneous" & wave == "1" &
           !(outcome %in% c("sr_health", "procrastinate"))) %>%
  mutate(outcome = recode(outcome, `sr_health` = "Self-Rated\nHealth",
                          `friendship_sat` = "Friendship\nSatisfaction",
                          `life_sat` = "Life\nSatisfaction")) %>%
  group_by(wave, outcome) %>%
  mutate(ymin = min(`2.5 %`, na.rm = T),
         ymax = max(`97.5 %`, na.rm = T),
         ymin2 = ifelse(abs(ymin) > abs(ymax), ymin, -1*ymax),
         ymax2 = ifelse(abs(ymax) > abs(ymin), ymax, -1*ymin)) %>%
  unite(comb, measure, wave, sep = ": ", remove = F) %>%
  ggplot(aes(x = var, y = estimate)) +
    geom_errorbar(aes(ymin = `2.5 %`, ymax = `97.5 %`), width = .2, position = "dodge") +
    geom_hline(aes(yintercept = 0)) +
    geom_point(aes(color = var), size = 5) +
    geom_blank(aes(y = ymin2)) + geom_blank(aes(y = ymax2)) +
    labs(y = "Estimated Centrality Coefficient", x = NULL) +
    coord_flip() +
    facet_grid(measure~outcome, scales = "free") +
    theme_classic() +
    theme(legend.position = "none",
          axis.text = element_text(face = "bold", size = rel(1.3)),
          axis.title = element_text(face = "bold", size = rel(1.3)),
          strip.text = element_text(face = "bold", size = rel(1.2)))
```




```r
ggsave(file = "~/Box Sync/network/PAIRS/outcomes/graphs/contemp_cent_bs.png", width = 6, height = 6)
```

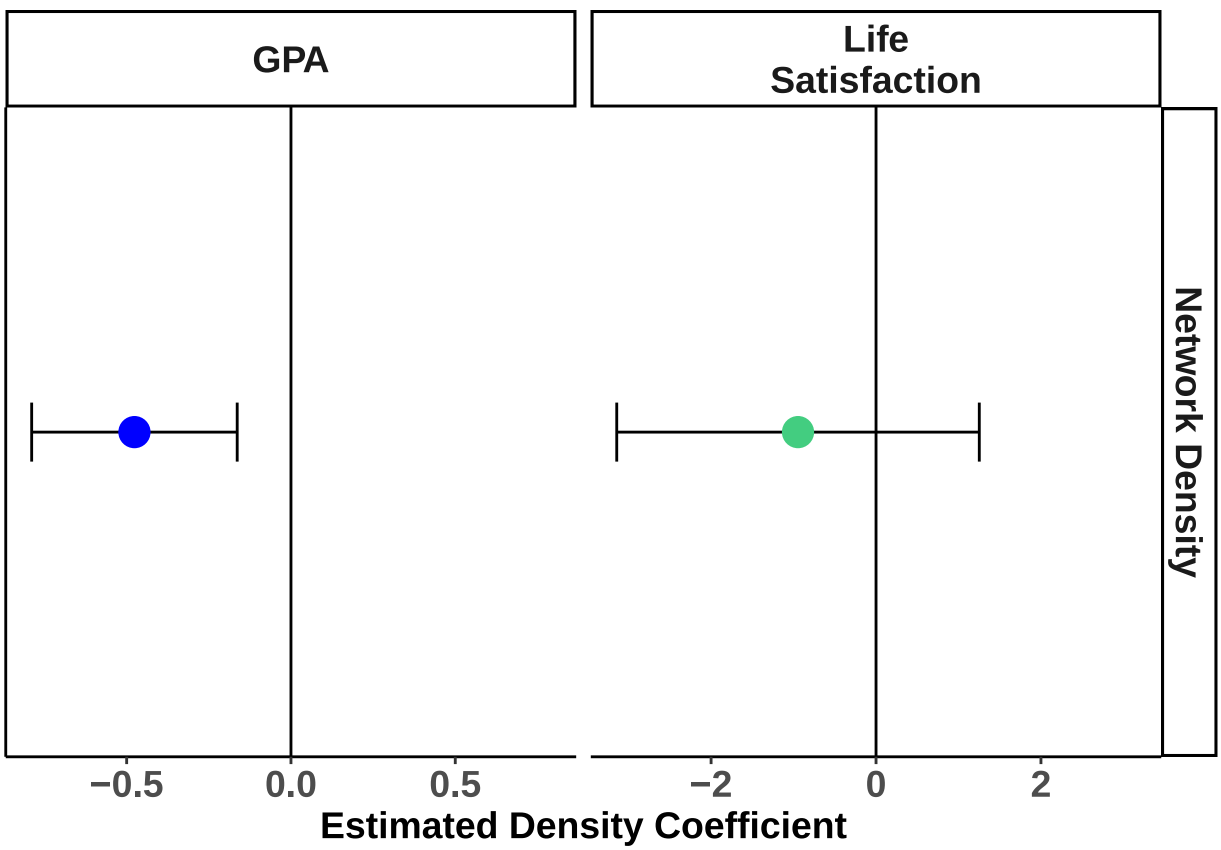
6 Question 3: Does density predict outcomes

```r
density_fun <- function(edgelist, type){
  if (type == "Temporal"){
    nvar <- length(unique(c(edgelist$from, edgelist$to)))
    pc <- nvar^2
  } else {
    nvar <- length(unique(c(edgelist$Var1, edgelist$Var2)))
    pc <- (nvar * (nvar - 1)) / 2
  }
  density <- sum(edgelist$value != 0) / pc
}

gVAR_fit <- gVAR_fit %>%
  mutate(temp_density0 = map_dbl(beta0, possibly(~density_fun(., "Temporal"), NA_real_)),
         contemp_density0 = map_dbl(kappa0, possibly(~density_fun(., "Contemporaneous"), NA_real_)))

density <- gVAR_fit %>%
  select(SID, wave, contains("density")) %>%
  gather(key = type, value = density, temp_density0:contemp_density0) %>%
  separate(type, into = c("type", "fit")) %>%
  mutate(type = recode(type, `temp` = "Temporal", `contemp` = "Contemporaneous"))
  #left_join(select(target.ratings.w1, SID, GPA, sr_health, friendship_sat))
```

6.1 Regressions

```r
mod_density <- function(data){lm(value ~ density , data = data)}

tidy_density <- function(model){
  tidy(model) %>%
    bind_cols(tbl_df(confint(model, method = "boot")))
}

density_fits <- density %>%
  left_join(target.ratings) %>%
  left_join(w1_composites) %>%
  select(SID, wave, density, type, fit, GPA, procrastinate, life_sat, ESM_Neuroticism, ESM_Conscientiousness) %>%
  gather(key = outcome, value = value, GPA:life_sat) %>%
  group_by(type, wave, fit, outcome) %>%
  nest() %>%
  mutate(model = map(data, possibly(mod_density, NA_real_)),
         tidy = map(model, possibly(tidy_density, NA_real_)))
```

```r
density_fits.df <- density_fits %>%
  filter(!is.na(tidy)) %>%
  unnest(tidy, .drop = T) %>%
  select(-std.error, -statistic) %>%
  gather(key = term2, value = value, estimate, p.value) %>%
  unite(comb, outcome, term2) %>%
  spread(key = comb, value = value)

density_fits %>%
  filter(!is.na(tidy)) %>%
  unnest(tidy, .drop = T) %>%
  filter(term == "density" & type == "Contemporaneous" & wave == "1" & fit == "density0" &
           !(outcome %in% c("sr_health", "procrastinate"))) %>%
  mutate(outcome = recode(outcome, `sr_health` = "Self-Rated Health",
                           `friendship_sat` = "Friendship Satisfaction",
                           `life_sat` = "Life\nSatisfaction"),
         measure = "Network Density") %>%
  unite(comb, outcome, wave, sep = ": ", remove = F) %>%
  group_by(wave, outcome) %>%
  mutate(ymin = min(`2.5 %`, na.rm = T),
         ymax = max(`97.5 %`, na.rm = T),
         ymin2 = ifelse(abs(ymin) > abs(ymax), ymin, -1*ymax),
         ymax2 = ifelse(abs(ymax) > abs(ymin), ymax, -1*ymin)) %>%
  ggplot(aes(x = 1, y = estimate)) +
    geom_errorbar(aes(ymin = `2.5 %`, ymax = `97.5 %`), width = .2, position = "dodge") +
    geom_hline(aes(yintercept = 0)) +
    geom_point(aes(color = outcome), size = 5) +
    scale_x_continuous(limits = c(0,2)) +
    geom_blank(aes(y = ymin2)) + geom_blank(aes(y = ymax2)) +
    scale_color_manual(values = c("blue", "seagreen3")) +
    #scale_y_continuous(limits = c(-1, 1), breaks = seq(-1, 1,.5)) +
    labs(y = "Estimated Density Coefficient", x = NULL) +
    coord_flip() +
    facet_grid(measure~outcome, scales = "free") +
    theme_classic() +
    theme(legend.position = "none",
          axis.text = element_text(face = "bold", size = rel(1.3)),
          axis.title = element_text(face = "bold", size = rel(1.3)),
          strip.text = element_text(face = "bold", size = rel(1.3)),
          axis.text.y = element_blank(),
          axis.ticks.y = element_blank())
```




```r
ggsave(file = "~/Box Sync/network/PAIRS/outcomes/graphs/contemp_density_bs.png", width = 5, height = 5)
```