

# IRT troubleshooting

Sara

10/17/2019

## Original code

```
set.seed(052319)

# load packages
packages = c("tidyverse", "janitor", "psych", "devtools", "PAutilities", "measurements", "here")
lapply(packages, library, character.only = TRUE)
rm(packages)

#keys and variable names
source(here("scripts/personality_scales.R"))

all_names = gsub("Extra", "B5: Extra", all_names)
all_names = gsub("Agree", "B5: Agree", all_names)
all_names = gsub("Neur", "B5: Neur", all_names)
all_names = gsub("Open", "B5: Open", all_names)
all_names = gsub("Consc", "B5: Consc", all_names)

#read in data
load(here("../data/SAPA/collaboration/SAPAdat07feb2017thru22jul2019forSara2.rdata"))
sapa = SAPAdat07feb2017thru22jul2019x

keys = read.csv("data/superKey.csv", header = TRUE, row.names = 1)

# -----
# score 5 personality factors (sum scores) #
# -----

# select just the rows that correspond to variables in the current SAPA dataset
keys = keys[names(sapa), ]

# select just the Big 5 scales that are scored using the SPI_135 form
bfkeys = keys %>%
  select(contains("SPI_135")) %>%
  select(1:5)

# score the items (this contains item and scale statistics too!)
scored = scoreItems(bfkeys, sapa)

# add scores to SAPA
sapa = cbind(sapa, scored$scores)

# -----
# score 27 personality factors (IRT scores) #
# -----

load("../data/SAPA/IRTinfoSPI27.rdata")
```

This section of coded was added on 10-18-19.

```
#reverse score items that should be negatively keyed
spi_keys = keys %>%
  select(contains("SPI_135"))%>%
  select(6:32) %>%
  mutate(item = rownames(.)) %>%
  gather("scale", "key", -item) %>%
  filter(key != 0)

#confirm each item is only in dataset once
length(unique(spi_keys$item)) == nrow(spi_keys)

reverse = spi_keys %>%
  filter(key == -1)
reverse = reverse[, "item"]
reverse_df = apply(sapa[, reverse], 2, function(x) max(x, na.rm=T) + 1 - x)
sapa[, reverse] = reverse_df
```

Return to original code.

```
# IRT score
dataSet <- subset(sapa, select = c(orderForItems))

SPIirtScores <- matrix(nrow=dim(dataSet)[1], ncol=27)

for (i in 1:length(IRToutputSPI27)) {
  data <- subset(dataSet, select = c(rownames(IRToutputSPI27[[i]]$irt$difficulty[[1]])))
  calibrations <- IRToutputSPI27[[i]]
  scored <- scoreIrt(calibrations, data, keys = NULL, cut = 0)
  SPIirtScores[,i] <- scored$theta1
}

SPIirtScores <- as.data.frame(SPIirtScores)
colnames(SPIirtScores) <- gsub("SPI27_", "SPI_135_27_5_", scaleNames)

#add to sapa dataset
sapa = cbind(sapa, SPIirtScores)
```

Results in the following associations.

```
R_sapa = sapa %>%
  select(matches("SPI")) %>%
  cor(use="pairwise")

rownames(R_sapa) = colnames(R_sapa) = all_names[rownames(R_sapa)]

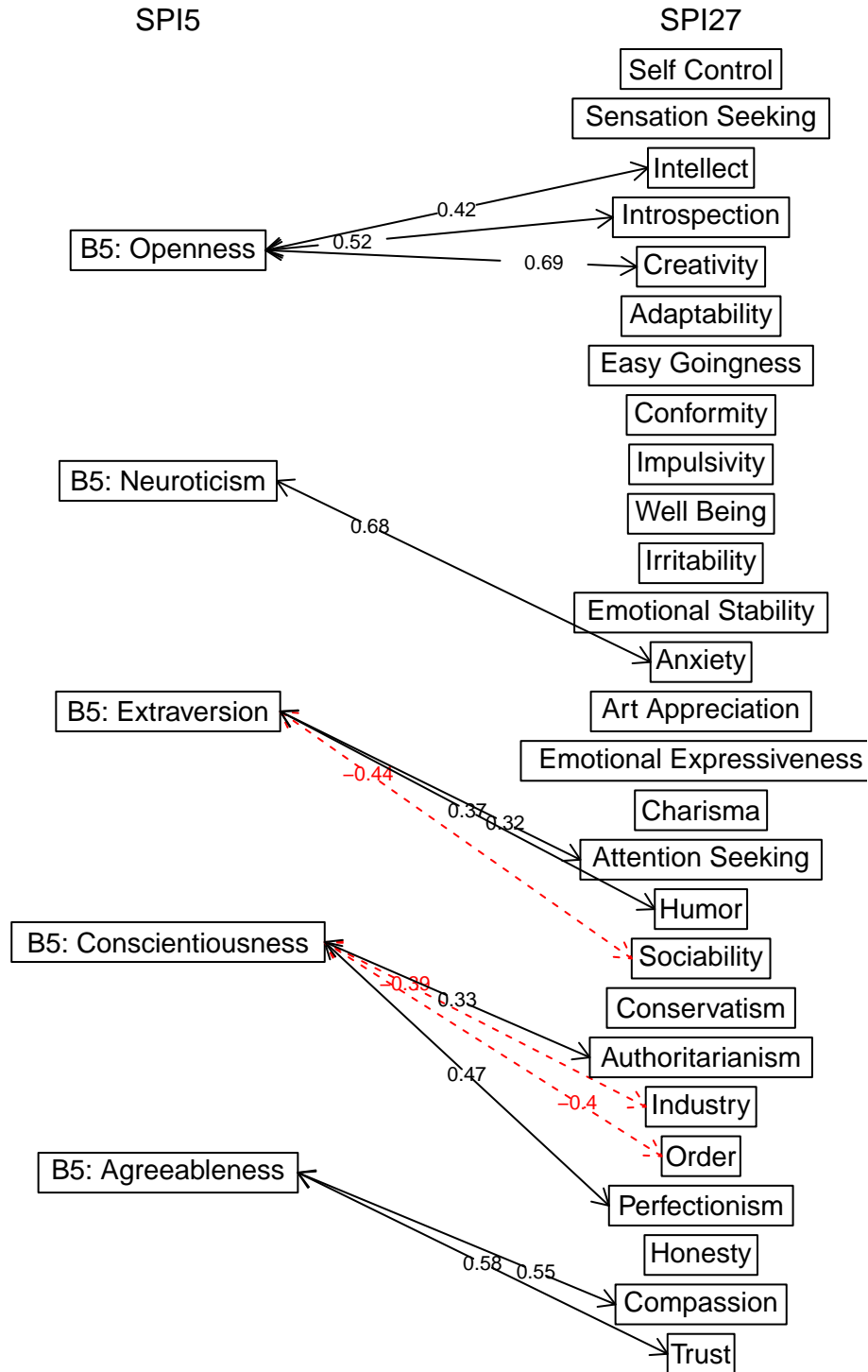
spi5 = which(grepl("B5", rownames(R_sapa)))
spi27 = which(!grepl("B5", rownames(R_sapa)))
new = rownames(psych::fa.sort(R_sapa[spi27, spi5]))

spi27 = sapply(new, function(x) which(rownames(R_sapa) == x))

psych::het.diagram(R_sapa,
  levels = list(SPI5 = spi5, SPI27 = spi27),
  cut=.3, digits=2, main = "Heterachy diagram (full sample)",
```

cex=.9, l.cex = .7)

## Heterarchy diagram (full sample)



Which is clearly wrong.

## Irritability (aka, the other title for this documentation)

I investigated one factor that seemed especially problematic, Irritability, which has a correlation of -0.253 with Neuroticism.

According to the files posted on sapa-project.org, the items for Irritability should be keyed in the following way:

- Positively keyed – q\_952, q\_1357
- Negatively keyed – q\_1683, q\_1585, q\_176

In theory, the negatively keyed items should have been dealt with per the code added on 10-18-19 which reverse scores any negatively keyed items. We can check to see if the items are all negatively correlated with neuroticism

```
cor(sapa$SPI_135_27_5_Neuro, sapa[,rownames(IRToutputSPI27$SPI27_Irritability$irt$difficulty[[1]])], us=)
```

```
##           q_952    q_1357    q_1683    q_1585    q_176
## [1,] 0.1863874 0.1668976 -0.1582735 -0.08418652 -0.07882458
```

And indeed they are.

What I checked next was the output stored in the IRT results loaded in, and found the following:

```
IRToutputSPI27$SPI27_Irritability$fa$loadings
```

```
##
## Loadings:
##      MR1
## q_952 -0.884
## q_1357 -0.818
## q_1683 0.734
## q_1585 0.738
## q_176 0.711
##
##      MR1
## SS loadings 3.041
## Proportion Var 0.608
```

According to the IRT output, the items that will be “reverse keyed” in the scoring function are the first two items, which should be positively keyed. Meaning, all items are scored opposite the direction they should be, explaining why this correlation is negative.

## A note

Not all 27 factor traits are scored opposite of what they should be. Some of the correlations in the heterarchy diagram make sense. The task is therefore to write code to check, for each trait, whether the key direction matches what it should according to the SPI documentation online and, if it doesn't match, perform an operation to correct this.

## The solution

We remove the code that reverse scores items before the IRT scoring procedure.

Within the IRT scoring loop, we add the following code:

```

loadings = calibrations$fa$loadings[,1]
loadings = ifelse(loadings < 0, -1, 1)
loadings = data.frame(item = names(loadings), loadings = loadings)
keys_direction = spi_keys %>%
  filter(grepl(scaleNames[i], scale)) %>%
  full_join(loadings)
same = sum(keys_direction$key == keys_direction$loadings)
if(same == 0) data[,1:ncol(data)] = apply(data[,1:ncol(data)], 2, function(x) max(x, na.rm=T) + 1 - x)
if (same > 0 & same < 5) print("Error in loadings")

```

This code pulls the direction of the loadings from the factor analysis matrix inside the IRT output and compares the direction of those loadings to the loadings scored in the keys matrix. If they are all the same, nothing changes. However, if they are all different, meaning the factor analysis will score the items in the opposite way of the key, we first reverse score all variables in this scale. If something between 0 and 5 matches occurs, the loop should report an error. This did not occur.

Again, we check the heterarchy:

```

R_sapa = sapa %>%
  select(matches("SPI")) %>%
  cor(use="pairwise")

rownames(R_sapa) = colnames(R_sapa) = all_names[rownames(R_sapa)]

spi5 = which(grepl("B5", rownames(R_sapa)))
spi27 = which(!grepl("B5", rownames(R_sapa)))
new = rownames(psych::fa.sort(R_sapa[spi27, spi5]))

spi27 = sapply(new, function(x) which(rownames(R_sapa) == x))

psych::het.diagram(R_sapa,
  levels = list(SPI5 = spi5, SPI27 = spi27),
  cut=.3, digits=2, main = "Heterarchy diagram (full sample)",
  cex=.9, l.cex = .7)

```

## Heterarchy diagram (full sample)

