

SEM Analysis

Francesco Aletta^{a,*}, Andrew Mitchell^a

^aUniversity College London, Institute for Environmental Design and Engineering, Central House, 14 Upper Woburn Place, London, WC1H 0NN

```
library(tidyverse) # For data manipulation
library(dplyr)      # For data manipulation
library(ggplot2)    # For data visualization
library(CircE)      # For circumplex SEM statistics
library(here)       # For file paths

source(here(project_dir, "sem_funcs.R"))

library(RCurl)      # For downloading data from Zenodo
library(readxl)     # For reading Excel files

temp.file <- paste0(tempfile(), ".xlsx")
download.file(
  "https://zenodo.org/records/10159673/files/SATP%20Dataset%20v1.4.xlsx",
  temp.file,
  mode="wb")
satp <- read_excel(temp.file,
  na = c("", "N/A"),
  col_types = c(
    "text", "text", "text", # Lan, Rec, Part
    "numeric", # Age
    "text", # Gender
    "numeric", "numeric", "numeric", "numeric",
    "numeric", "numeric", "numeric", "numeric", # PAQs
    "numeric", # loud
    "text", # Inst
    "numeric" # sequence
  )
)

# Prep variables for the circumplex analysis
scales <- c("PAQ1", "PAQ2", "PAQ3", "PAQ4",
  "PAQ5", "PAQ6", "PAQ7", "PAQ8"
) # Names of the scales for circumplex analysis
# Ideal angles for circumplex analysis
eq.angles <- c(0, 45, 90, 135, 180, 225, 270, 315)
```

*Corresponding author

Email addresses: f.aletta@ucl.ac.uk (Francesco Aletta), andrew.mitchell.18@ucl.ac.uk (Andrew Mitchell)

```

# Ipsatize the data
# For each participant, we subtract the mean of their response to all scales
# across all recordings from their response to each scale for each recording
# This is done at the suggestion of JM Girard/R Circumplex

satp |>
  group_by(Participant) |>
  mutate(Mean = mean(c_across(all_of(scales)), na.rm = TRUE)) -> parts_means

satp[scales] <- satp |>
  select(all_of(scales)) |>
  mutate(across(all_of(scales), ~ .x - parts_means$Mean))

# Load the acoustic analysis data
lvls <- read_excel(here(project_dir, "LLAN.xlsx"),
  skip = 1,
  na = c("", "N/A"),
  col_types = c("text", "text", "numeric", "numeric", "numeric", "numeric"),
  col_names = c("Recording", "Channel", "Leq", "LAeq", "N", "LA90")
)

# Extract the max for each recording to reduce from two channels to one
result <- lvls |>
  group_by(Recording) |>
  mutate(max_N = ifelse(row_number() == 1, max(N), NA_integer_)) |>
  select(Recording, max_N) |>
  na.omit()

# Join to SATP dataset
satp <- left_join(satp, result, by=join_by(Recording == Recording))

```

0.1. Step One: Confirmatory Factor Analysis

Run all four models for a language and compile into a single table:

0.1.1. Fit Indices

Fit Index	Threshold
Chi-squared (χ^2)	$p < 0.05$
Comparative Fit Index (CFI)	0.90
Goodness of Fit Index (GFI)	0.90
Standardized Root Mean Square Residual (SRMR)	0.08
Root Mean Squared Error of Approximation (RMSEA)	0.13
Minimum Common Score Correlations (MCSC)	-0.8
Gap difference test (GDIFF)	20

Fit indices and thresholds, including the reference from which the threshold is derived. {#tbl-indices}

Each model is assessed against a suite of SEM fit indices, summarised in **?@tbl-indices**. These indices include the χ^2 test, Comparative Fit Index (CFI), Goodness of Fit Index (GFI), Standardized Root Mean Square Residual (SRMR), and Root Mean Squared Error of Approximation (RMSEA). Two additional

indices are calculated which are not considered ‘typical’ fit indices for SEMs, but which have proved to be crucial for the analysis of the circumplex. The first is the Minimum Common Score Correlations (MCSC). The second is the Gap Difference test (GDIFF), which cont. The thresholds for each index are derived from the literature, and are summarised in [?@tbl-indices](#).

0.1.2. Run Step one

```
# Run the models for English
satp_eng <- satp[satp$Language == "eng", ]
circe_satp_eng <- run_all_models(satp_eng, "SATP", "eng", m=3)

# languages <- unique(satp$Language) # Get a list of all the languages

# full_table <- circe_satp_eng$res_table # Start with the English results
# for (lang in languages) {
#   if (lang == "eng" | lang == "zsm") { # TODO: add try catch block
#     next # Skip English, we've already done it
#   }
#   print("=====")
#   print(lang)
#   print("=====")
#   lang_data <- satp[satp$Language == lang, ] # Filter to just the language we want
#   lang_res <- run_all_models(lang_data, "SATP", lang, m=3) # Run the models
#   full_table <- rbind(full_table, lang_res$res_table) # Add the results to the table
# }

# knitr::kable(full_table)
```

References