Supplementary Material (B) - Code for Steps 1 & 2, Browne's Circular Stochastic Process Model

Andrew Mitchell[a,*], Francesco Aletta[a]

[a] *University College London, Institute for Environmental Design and Engineering, Central House, 14 Upper Woburn Place, London, WC1H 0NN*

Abstract

Code accompanying the paper: "Soundscape descriptors in eighteen languages: translation and validation through listening experiments

1. Testing the quasi-circumplex structure (Steps 1 and 2)

```
library(devtools)
library(ggplot2)
library(tidyverse)
library(dplyr)
library(readxl)
library(here)
library(knitr)
library(CircE)
library(RCurl)

source(here("utils/sem_funcs.R")) # Load our own functions

# Create a folder for the outputs
dir.create(here("outputs", Sys.Date()), showWarnings = FALSE)
output_dir <- here("outputs", Sys.Date())

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

# Load in the SATP dataset from Zenodo
temp.file <- paste0(tempfile(), ".xlsx")
download.file(
    "https://zenodo.org/records/10159673/files/SATP%20Dataset%20v1.4.xlsx",
    temp.file,
    mode="wb")
```

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

```r
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
    )
  )
```

1.0.1. Ipsatization

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.

```r
# Ipsatize the data

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

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

1.1. Step One: Tracey's Circular Order Model

```r
library(RTHORR)
source(here("utils/RTHORR_funcs.R")) # Load our own functions

# Run the RTHORR analysis
matrices <- list(na.omit(satp[scales]))
names(matrices) <- "SATP"
for (lang in unique(satp$Language)) {
    lang_data <- na.omit(satp[satp$Language == lang, ][scales])
    matrices[[lang]] <- lang_data
}
randall_df_output <- my_randall_from_df(matrices, names(matrices), ord="circular8")
knitr::kable(randall_df_output, digits=3, align = "c")
```

Table 1: Results of the Circular Order analysis of the SATP dataset.

mat	pred	met	tie	CI	p	description
1	288	283	0	0.965	0.000	SATP
2	288	272	0	0.889	0.000	arb
3	288	262	0	0.819	0.000	cmn
4	288	284	0	0.972	0.000	deu
5	288	276	0	0.917	0.000	ell

Table 1: Results of the Circular Order analysis of the SATP dataset.

mat	pred	met	tie	CI	p	description
6	288	286	0	0.986	0.000	eng
7	288	278	0	0.931	0.000	fra
8	288	268	0	0.861	0.000	hrv
9	288	255	0	0.771	0.000	ind
10	288	275	0	0.910	0.000	ita
11	288	264	0	0.833	0.000	jpn
12	288	262	0	0.819	0.000	kor
13	288	261	0	0.812	0.000	nld
14	288	254	0	0.764	0.000	por
15	288	283	0	0.965	0.000	spa
16	288	284	0	0.972	0.000	swe
17	288	261	0	0.812	0.000	tur
18	288	244	0	0.694	0.002	vie
19	288	241	0	0.674	0.002	zsm

```r
print("Pass: ")
```

```
[1] "Pass: "
```

```r
print(randall_df_output[randall_df_output$CI > 0.7, "description"])
```

```
 [1] "SATP" "arb"  "cmn"  "deu"  "ell"  "eng"  "fra"  "hrv"  "ind"  "ita"
[11] "jpn"  "kor"  "nld"  "por"  "spa"  "swe"  "tur"
```

```r
print("Fail: ")
```

```
[1] "Fail: "
```

```r
print(randall_df_output[randall_df_output$CI < 0.7, "description"])
```

```
[1] "vie" "zsm"
```

```r
pass <- randall_df_output[randall_df_output$CI > 0.7, "description"][-1]
satp <- satp[satp$Language %in% pass, ] # Filter to just the languages that pass
```

1.2. Structural Equation Modelling using Browne's Stochastic Circumplex Model

1.2.1. Run `CircE` Analysis

The bulk of the code for this process has been pulled out into a separate `sem_funcs.R` file, which is loaded at the beginning of the analysis. This file contains the functions used to run the circumplex analysis and compile the results into a single table.

`step_one_test(data, model_type, scales = c("PAQ1", "PAQ2", "PAQ3", "PAQ4", "PAQ5", "PAQ6", "PAQ7", "PAQ8"), m = 3) )` is the function used to run the circumplex analysis for a single model for a single language. It takes the data for that language, the model_type (one of `Circumplex`, `Equal comm.`, `Equal ang.`, or `Unconstrained`), the names of the scales, and the number of betas for the fourier series correlation function (we're using `m=3` by default). It then runs the analysis and returns a list of the results, including a list of the desired results (`res_list`) and the model object (`res_model`).

`run_all_models(data, datasource, language, m)` is the function used to run the circumplex analysis for all four models for a single language. It takes the data for that language, the name of the data source (e.g. `SATP`), the language code, and the number of betas for the fourier series correlation function (m). It then runs the analysis for each of the four models and returns a list of the results, including a list of the four results and a table combining the results from all four models.

First, we run the circumplex analysis for the English data. This is done separately from the other languages to set up the results data table.

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

Then, we run the circumplex analysis for each of the other languages. This is done in a loop, with each language being run separately. The results for each language are then added to the results table.

Within each loop, we check for any errors in execution and append these to a list of errors to inspect later.

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

full_table <- circe_satp_eng$res_table # Start with the English results
for (lang in languages) {
    if (lang == "eng") {
        next # Skip English, we've already done it
    }
    print("======================================")
    print(lang)
    print("======================================")
    lang_data <- satp[satp$Language == lang, ] # Filter to just the language we want

    pass_on_error <- FALSE
    errors <- list()
    tryCatch(
      lang_res <- run_all_models(lang_data, "SATP", lang, m = 3),
      error = function(e) {
        pass_on_error <<- TRUE
        errors[lang] <<- e
      }
    )
    if (pass_on_error) {
      next
    }
    # 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
}
```

```
# Catching any errors

for (name in names(errors)) {
  print("==========  Error in:  =============")
  print(errors[name])
}
```

1.2.2. SEM Analysis Results

Below is the table of results for the circumplex analysis of the soundscape survey translations. The table includes the results for each of the four models for each language. The results are presented in the order of the models, with the unconstrained model first, followed by the equal spacing model, the equal communality model, and the circumplex model. The results for each model include the χ^2 test, CFI, GFI, SRMR, RMSEA, MCSC, and GDIFF. These results are saved to a CSV file for later use.

Importantly, this table also reports the derived angles for each scale for the unconstrained and Equal comm. models. These angles will be carried over and used in the next stage of the analysis, where we will validate the survey instrument by correlating the survey responses with the acoustic indices using the Structural Summary Method (SSM).

```r
write.csv(full_table, here(output_dir, "sem-fit-ipsatized.csv"))
if (is_html) {
    kable(full_table, digits = 5, align = "c") %>%
        kableExtra::kable_styling(
          bootstrap_options = c(
            "striped", "hover", "condensed", "responsive", "bordered"
            )
          )
} else {
    kableExtra::kbl(
      full_table[, ! colnames(full_table) %in% c(
        'Dataset', 'Model Type', 'RMSEA.L', 'RMSEA.U', 'GDIFF', scales
        )],
      format = "latex",
      row.names = FALSE,
      booktabs = T,
      digits = 5,
      align = "c",
      longtable = TRUE,
      linesep = c("", "", "", "\\addlinespace")
    ) |>
    kableExtra::kable_classic_2()
}
```

Language	n	m	ChiSq	df	p	CFI	GFI	AGFI	SRMR	MCSC	RM
eng	864	3	75.855189055029	10	3.24229532111531e-12	0.988	0.981	0.932	0.037	-0.936	0
eng	864	3	370.055528785066	17	0	0.934	0.907	0.803	0.052	-0.898	0
eng	864	3	534.252188046965	17	0	0.903	0.869	0.723	0.099	-0.919	0
eng	864	3	830.648940930941	24	0	0.848	0.811	0.717	0.111	-0.92	0
arb	809	3	44.0444222102112	10	3.23173722305281e-06	0.99	0.99	0.964	0.018	-0.825	0
arb	809	3	119.261928856334	17	0	0.971	0.969	0.934	0.044	-0.849	0
arb	809	3	527.624555746401	17	0	0.857	0.863	0.71	0.179	-0.85	0
arb	809	3	649.442216172646	24	0	0.825	0.837	0.755	0.18	-0.834	(
cmn	1832	3	172.220795767266	10	0	0.982	0.978	0.921	0.023	-0.989	0
cmn	1832	3	366.933848618136	17	0	0.96	0.954	0.903	0.044	-0.988	0
cmn	1832	3	1542.17621055718	17	0	0.827	0.828	0.636	0.288	-0.956	0
cmn	1832	3	1716.87022466114	24	0	0.808	0.813	0.719	0.284	-0.959	0
deu	810	3	23.373842792726	10	0.00944767599528051	0.997	0.996	0.986	0.009	-1	0
deu	810	3	316.718936400525	17	0	0.943	0.915	0.82	0.059	-0.998	0
deu	810	3	403.258868716265	17	0	0.926	0.893	0.773	0.133	-0.983	0
deu	810	3	766.612033890909	24	0	0.859	0.813	0.719	0.137	-0.97	0
ell	810	3	71.4809335919041	10	2.29338770196819e-11	0.981	0.981	0.932	0.026	-0.996	0

ell	810	3	246.732675808593	17	0	0.928	0.934	0.86	0.079	-1	0
ell	810	3	445.382177926038	17	0	0.866	0.884	0.754	0.126	-0.947	0
ell	810	3	595.981810321126	24	0	0.82	0.849	0.773	0.143	-0.931	0
fra	891	3	41.4589598217322	10	9.3567026955288e-06	0.993	0.991	0.968	0.017	-0.954	0
fra	891	3	357.314599675967	17	0	0.919	0.913	0.816	0.098	-0.965	0
fra	891	3	267.552998715169	17	0	0.94	0.934	0.86	0.124	-0.939	0
fra	891	3	625.064878799758	24	0	0.857	0.855	0.782	0.138	-0.925	0
hrv	864	3	58.920907564549	10	5.79625847318965e-09	0.991	0.986	0.95	0.014	-0.914	0
hrv	864	3	290.700218409579	17	0	0.949	0.926	0.843	0.065	-0.895	0
hrv	864	3	1394.2010574866	17	0	0.743	0.715	0.396	0.201	-0.877	0
hrv	864	3	1688.56505488852	24	0	0.689	0.675	0.513	0.214	-0.873	0
ind	891	3	39.9618968523582	10	1.72085370703234e-05	0.993	0.992	0.971	0.022	-0.959	0
ind	891	3	315.187566703313	17	0	0.933	0.923	0.837	0.078	-0.905	0
ind	891	3	732.266073879797	17	0	0.84	0.833	0.646	0.21	-0.928	0
ind	891	3	1177.14093335817	24	0	0.742	0.756	0.634	0.258	-0.923	0
ita	810	3	58.381584144373	10	7.32599136910039e-09	0.989	0.985	0.946	0.03	-0.958	0
ita	810	3	251.280293245251	17	0	0.944	0.932	0.856	0.069	-0.952	0
ita	810	3	660.36653177006	17	0	0.847	0.835	0.651	0.155	-0.903	0
ita	810	3	933.149067661957	24	0	0.784	0.781	0.671	0.168	-0.902	0
jpn	917	3	26.8028563303843	10	0.00279799807460734	0.996	0.996	0.986	0.015	-0.944	0
jpn	917	3	440.259195572614	17	0	0.892	0.896	0.78	0.087	-0.994	0
jpn	917	3	760.838278422792	17	0	0.809	0.831	0.642	0.184	-0.895	0
jpn	917	3	1200.25366437214	24	0	0.699	0.757	0.635	0.196	-0.906	0
kor	810	3	42.9454591678388	10	5.085936594873e-06	0.992	0.99	0.964	0.012	-0.996	0
kor	810	3	220.342708701949	17	0	0.952	0.941	0.875	0.084	-0.994	0
kor	810	3	605.83286625951	17	0	0.86	0.846	0.674	0.224	-0.999	0
kor	810	3	763.881376570395	24	0	0.824	0.814	0.721	0.228	-0.999	0
nld	864	3	31.4364127000124	10	0.00049684137665007	0.997	0.994	0.978	0.014	-0.945	0
nld	864	3	225.41010308231	17	0	0.967	0.943	0.879	0.056	-0.917	0
nld	864	3	1069.73153202843	17	0	0.833	0.766	0.504	0.247	-0.901	0
nld	864	3	1505.26755819124	24	0	0.766	0.7	0.55	0.254	-0.88	0
por	1890	3	98.279990399476	10	1.11022302462516e-16	0.99	0.989	0.96	0.018	-0.898	0
por	1890	3	703.273165383338	17	0	0.925	0.917	0.824	0.092	-0.869	0
por	1890	3	1279.35636110687	17	0	0.862	0.857	0.697	0.217	-0.835	0
por	1890	3	2116.46532429089	24	0	0.771	0.783	0.675	0.232	-0.833	0
spa	1647	3	117.438882600955	10	0	0.987	0.984	0.942	0.035	-0.984	0
spa	1647	3	671.283195056481	17	0	0.92	0.91	0.809	0.063	-0.974	0
spa	1647	3	489.9185395306	17	0	0.942	0.933	0.858	0.109	-0.983	0
spa	1647	3	1049.9852694728	24	0	0.875	0.865	0.797	0.132	-0.991	0
swe	945	3	63.5433178668487	10	7.69214247853256e-10	0.99	0.986	0.95	0.016	-0.94	0
swe	945	3	326.094447246618	17	0	0.944	0.924	0.839	0.053	-0.946	0
swe	945	3	683.641662587339	17	0	0.878	0.85	0.682	0.134	-0.919	0
swe	945	3	893.001335390687	24	0	0.841	0.813	0.719	0.142	-0.928	0
tur	918	3	107.188172969441	10	0	0.979	0.974	0.906	0.03	-0.905	0
tur	918	3	358.886881929338	17	0	0.927	0.915	0.82	0.079	-0.915	0
tur	918	3	744.570406538834	17	0	0.844	0.835	0.651	0.228	-0.867	0

tur	918	3	1057.39307595611	24	0	0.778	0.78	0.67	0.23	-0.834	0

References