Pipeline Statistical Analysis

Getting the Brain into Gear – an online healthy ageing study

Contents

Data wrangling

Control tasks

- Filename: CleanControlTasks.R
- Packages: dplyr','tidyr','purrr', 'rio', 'tibble', 'hablar', 'ggplot2', 'psych', 'tidyverse', 'readr', 'plyr', 'multicon', 'DescTools'
- Stroop Task (executive functioning; inhibition):
  1) Select columns/variables of interest
  2) Filter out rows without information for analysis (e.g. a row that says START TASK)
  3) Filter out participants that were excluded based on screening, missing data, etc.
  4) Remove column with the public ID of the participants (to ensure anonymity)
  5) Export as *Stroop_tidier*
  6) Filter out incorrect and practice trials
  7) Summarise → create mean reaction time per participant for compatible and incompatible trials separately
  8) Create a stimulus-response compatibility variable (*Stroop.SRC*) by subtracting the reaction time of incompatible from the reaction time of compatible trials (incompatible – compatible)
  9) Convert data frame into wide format (i.e. no repeated measures)
  10) Standardise Stroop.SRC by creating z-scores
  11) Winsorize extreme outliers to -2.5 and 2.5 (there were no missing values).

12) Because the data is reaction time data, lower z-scores mean higher inhibitory control (i.e., faster reaction times. Reverse Stroop SRC z scores by multiplying the z-scores per participant by -1, so that higher values mean a higher score.
13) Save file as *Stroop_scores* for inclusion data analysis.

- Simple and Choice Reaction Time task (speed of processing):
  1) Select columns/variables of interest
  2) Filter out rows without information for analysis (e.g. a row that says START TASK)
  3) Filter out participants that were excluded based on screening, missing data, etc.
  4) Remove column with the public ID of the participants (to ensure anonymity)
  5) Export as *SoP_tidier*
  6) Filter out incorrect and practice trials
  7) Summarise → create mean reaction time for Simple Reaction Time (SRT) and Choice Reaction Time (CRT) separately (1 NA for SRT and 1 NA for CRT but different participants).
  8) Convert data into wide format (i.e. no repeated measures)
  9) Standardise SRT and CRT scores by creating z-scores.
  10) Winsorize extreme outliers to -2.5 and 2.5.
  11) Create a composite variable by averaging the z-scores of SRT and CRT per participant (no NAs after this).
  12) Because the data is reaction time data, lower z-scores mean faster processing speed. Reverse SoP composite score by multiplying the z-scores per participant by -1, so that higher values mean a higher score.
  13) Save file as *SoP_scores* for inclusion data analysis
- Digit Reordering task (working memory):
  1) Select columns/variables of interest
  2) Filter out rows without information for analysis (e.g. a row that says START TASK)
  3) Filter out participants that were excluded based on screening, missing data, etc.
  4) Remove column with the public ID of the participants (to ensure anonymity)
  5) Export as *WM_tidier*
  6) Filter out practice trials
  7) Create average Working Memory scores per participant by taking the sum of the number of digits that had to be remembered multiplied by 0 (for incorrect trials) or 1 (for correct trials). For example, the number of digits that had to be reordered was 4, then the score for that trial would be 4 for correct trials and 0 for incorrect trials. This way, the more digits the participant had to remember, the higher the score would be if the answer was correct, and hence, weighted scores.
  8) 3 participants had a score of 0 as WM score (2 older adults and 1 middle-aged adult), so we treated these as missing data (as it's very unlikely the participant would have scored 0 correct). We used single imputation by imputing the mean of the age group the participant belonged to.
  9) Standardise the WM score per participant by using z-scores.
  10) Winsorize extreme outliers to -2.5 and 2.5
  11) Save file as *WM_scores* for inclusion data analysis.

Cognitive Reserve Questionnaire

- Filename: CRqCleaning.R
- Packages: 'dplyr','tidyr','purrr', 'rio', 'tibble', 'hablar', 'DescTools', 'multicon'

- Steps:

1) Select columns/variables of interest
2) Rename the variables/columns to meaningful names (e.g. Q19 becomes *Age.Category*)
3) Recode answers of General Activities Questionnaire for quantification (e.g. never gets a score of 0)
4) Recode answers of physical activities questionnaire so that “no” gets a score of 0 and “yes” a score of 1. This is necessary for later calculation of the physical activity score.
5) Recode NA answers as 0. This is only for answers that participants hadn’t seen because they previously selected “no”.
6) Convert variables with class *character* to class *numeric.*
7) Filter out participants with a Beck Depression Inventory score of 20 or higher, with an IQCODR score of 3.6 or higher, participants that were excluded due to e.g. missing data, and participants that were not right-handed.
8) Export file as *CRq_tidier*
9) In Excel: obtain SOC 2020 occupation scores using Cascot: Computer Assisted Structured COding Tool to calculate the Standard Occupation Classification 2020 scores. Then in R:
10) Replace missing data per column/variable by mean the mean of that column/variable, based on age category (i.e. the means for young, middle-aged, and older adults differed). We chose to impute missing values by the mean as to avoid missing data in later analysis as this variable is an important predictor. Missing values would lead to exclusion of all repeated measures of that participant in the model, and hence, a significant reduction in sample size.
11) Create mean scores per subscale of the General Activities Questionnaire (i.e. for cognitively stimulating, social, and productive activities separately)
12) Create a composite score for the Physical Activities Questionnaire. Scores were calculated by summing up the scores for vigorous and moderate activity related to work and to sports, and physical activity due to traveling. Each score was calculated by: 0 (no) or 1 (yes) * number of days in a week * total minutes on a day * 4 (for moderate activity or travel activity) or 8 (vigorous activity).
    a. E.g. someone who participated in vigorous activity for 60 minutes for 5 days in a week would receive a score of 1*5*60*8 = 2400. Someone who didn’t participate in vigorous work activity would receive a score of 0 (0*0*0*8)
13) Select columns/variables of interest that will be included in the complete dataset with the behavioural data.
14) Impute missing data for the physical composite score by the mean of the age category.
15) Create a General Activity Score by summing up the CR cognitive, social, and productive score.
16) Save as *CRq_tidiest*
17) Standardise the General Activities CR score, Physical Activity composite score, and years of education using z-scores, per age group.
18) Winsorize extreme outliers for General Activities CR, Physical Activity, and education z-scores to -2.5 and 2.5 to avoid missing data for the data analysis. By winsorizing data, extreme data would still be on the extreme sides but would not lead to loss of data/decreased sample size.
19) Create a CR composite score by averaging the z-scores of General Activities, Physical activity, occupation code, and years of education per participant.
20) Save as *zCRq_tidiest*

Picture Naming – Wrangling

- Filename: CleanPictureNaming.R
- Packages: 'dplyr','tidyr','purrr', 'rio', 'tibble', 'hablar', 'ggplot2', 'psych', 'tidyverse'

This was done for object and action naming separately but the same steps were taken.

1) Select rows of interest (both practice and real trials), while excluding rows without information for data analysis. We also excluded the first trial as it might contain measurement error due to first microphone use and stimulus loading.
2) Select columns/variables of interest.
3) Unite columns that contain the same information but represented in different columns due to belonging to different branches in the Gorilla Experiment Builder.
4) Filter out participants that were excluded
5) Save as *PNobjects_tidier* and *PNactions_tidier*
6) Add the General Cognitive Processing scores by participants that were obtained via CleanControlTasks.R
7) Save as *PNobjects_tidiest* and *PNactions_tidiest.*

Picture Naming – Complete Dataset

- Filename: CompleteDatasets.R
- Packages: 'dplyr','tidyr','purrr', 'rio', 'tibble', 'hablar', 'readxl', 'ggplot2'
- Steps:

1) Combine all Picture Naming answer files for action and object naming separately. In context: each participant had a separate excel file where their anonymous ID, answer, reaction time, and accuracy were transcribed/recorded. These files had to be merged.
2) Drop stimuli/trials that had to be excluded due to low accuracy or low name agreement across the participant sample.
3) Merge the combined Picture Naming answer files with the data file obtained through Gorilla and with the CRq questionnaire data.
4) Convert the Accuracy with code “2” to code “1”. Code 2 means that participants correctly names the picture but used an alternative answer to the target name.
5) Exclude practice trials
6) Save as *PNactions_complete* and *PNobjects_complete.*

Verbal Fluency – Wrangling

- Filename: CleanVerbalFluency.R
- Packages: 'dplyr','tidyr','purrr', 'rio', 'tibble', 'hablar', 'readxl', ‘fs’, ‘stringr’, ‘tidyverse’
- Steps:

1) Combine all Verbal Fluency answer files for semantic, letter, and action fluency separately. In context: each participant had a separate excel file where their anonymous ID, answer, reaction time, and accuracy were transcribed/recorded. These files had to be merged.
2) Select variables of interest for data analysis
3) Filter out participants that were excluded
4) Add the General Cognitive Processing scores by participants that were obtained via CleanControlTasks.R
5) Save as *VFcat_tidiest_final.csv*, *VFact_tidiest_final.csv,* and *VFlet_tidiest_final.csv.*

Verbal Fluency – Complete dataset

- Filename: CompleteDatasets.R
- Packages: 'dplyr','tidyr','purrr', 'rio', 'tibble', 'hablar', 'readxl', 'ggplot2'
- Steps:

1) Add participant private ID to the datasets so they can be merged later with the other datafiles.
2) Only include the measures for Number of Correctly Produced Words and Average Frequency as these are the measures of interest for data analysis.
3) Merge the verbal fluency data frames with the CRq questionnaire data.
4) For each verbal fluency prompt, standardise the scores (z-scores)
5) Filter out any outliers (i.e., above or below +/-2 standard deviations) for Number of Correctly Produced Words and Average Frequency separately (using group_by).
6) Create verbal fluency composite scores per participants by averaging the z-scores per verbal fluency task. This resulted in 3 composite scores: one for semantic fluency, one for letter fluency, and one for action fluency.
7) Save as *VFcat_complete_final.csv, VFlet_complete_final.csv,* and *VFact_complete_final.csv.*

Data Analysis – Picture Naming

Reaction Time data

- Filename: R Code Full Analysis Picture Naming Reaction Times.Rmd
- Packages: 'dplyr','tidyr','purrr', 'rio', 'tibble', 'hablar', 'fs', 'ggplot2', 'e1071', ' stringr', 'tidyverse', 'lme4', 'lmerTest', 'performance', 'lattice', 'broom', 'car', 'sjPlot', 'knitr'
- Steps:

1) Combined the object and action naming data frames.
2) Filter out inaccurate trials as only reaction times of accurate trials will be investigated.
3) Standardise reaction times per age group and type (i.e., object vs. action) and remove outliers. Outliers are any reaction times above or below +/-2.5 SD.
4) Create visualisations to check the data.

For action and object naming separately (filter out one type, i.e., either actions or objects, to run the analysis).

5) Create Helmert contrasts for the variable Age Category, with middle-aged adults as first factor level, so that the first contrast in the model output refers to middle-aged vs. younger adults, and the second contrast will refer to older adults vs. the average between younger and middle-aged adults.
6) Check skewness
7) Run base model with no predictors and no random effects, and base models with one, two, or both random effects to argue for inclusion of random effects (using AIC).
8) Run full model with predictors, covariates and random effects.
9) Check assumptions (linearity, homoscedasticity, normally distributed residuals.
10) Log transform reaction times to try obtain better normality of residuals.
11) Recheck model fit and outliers
12) Calculate effect sizes (marginal and conditional R2)
13) Run model comparisons for the CR measure preceding and coinciding with the covid-19 pandemic using anova's

Accuracy data

- Filename: R Code Full Analysis Picture Naming Reaction Times.Rmd
- Packages: 'dplyr','tidyr','purrr', 'rio', 'tibble', 'hablar', 'fs', 'ggplot2', 'e1071', ' stringr', 'tidyverse', 'lme4', 'lmerTest', 'performance', 'lattice', 'broom', 'car', 'knitr', 'DHARMa', 'MuMIn', 'Hmisc'
- Steps:

1) Combined the object and action naming data frames.
2) Create visualisations to check the data.

For action and object naming separately (filter out one type, i.e., either actions or objects, to run the analysis).

3) Create Helmert contrasts for the variable Age Category, with middle-aged adults as first factor level, so that the first contrast in the model output refers to middle-aged vs. younger adults, and the second contrast will refer to older adults vs. the average between younger and middle-aged adults.
4) Run base model with no predictors and no random effects, and base models with one, two, or both random effects to argue for inclusion of random effects (using AIC).
5) Run full model with predictors, covariates and random effects. Family = binomial, with a "cloglog" function to account for the type of outcome variable (i.e., binomial with more 1's than 0's).
6) If the model doesn't converge: Check model convergence problems (e.g., singularity) and restart model from previous fit and run with 20000 iterations.
7) Check GLMER assumptions (appropriate link function, no over- or underdispersion).
8) Compute concordance and Somer's D to assess the predictive performance of the model.
9) Compute the effect size (marginal and conditional R2).
10) Run model comparisons for the CR measure preceding and coinciding with the covid-19 pandemic using anova's

Data Analysis – Verbal Fluency

Number of Correctly Produced Words

- <u>Filename:</u> R Code Full Analysis Verbal Fluency Number of Correctly Produced Words.Rmd
- <u>Packages:</u> 'dplyr','tidyr','purrr', 'rio', 'tibble', 'fs', 'ggplot2', 'e1071', ' stringr', 'tidyverse', 'performance', 'lattice', 'broom', 'car', 'olsrr, 'corrplot
- <u>Steps:</u>

  For semantic, letter, and action fluency separately:

1) Filter out rows which reflect the "Average Frequency" measure so that only the "Number of correctly produced words" variable is included.
2) Filter out any outliers (i.e., +/-2.5 SD) from the composite fluency score.
3) Run descriptives and visualise data
4) Create Helmert contrasts for the variable Age Category, with middle-aged adults as first factor level, so that the first contrast in the model output refers to middle-aged vs. younger adults, and the second contrast will refer to older adults vs. the average between younger and middle-aged adults.
5) Run unconditional (i.e., without covariate) and full model.
6) Check assumptions multiple linear regression (linearity, independence of predictor variables, normal distribution of residuals, homoscedasticity).
7) Run model fit diagnostics (Variation Inflation Factor, observed vs. predicted plot)
8) Run model comparisons for the CR measure preceding and coinciding with the covid-19 pandemic using anova's

Average Frequency of Correctly Produced Words

- <u>Filename:</u> R Code Full Analysis Verbal Fluency Average Frequency.Rmd
- <u>Packages:</u> 'dplyr','tidyr','purrr', 'rio', 'tibble', 'fs', 'ggplot2', 'e1071', 'stringr', 'tidyverse', 'performance', 'lattice', 'broom', 'car', 'olsrr, 'corrplot', 'patchwork'
- <u>Steps:</u>

  For semantic, letter, and action fluency separately:

9) Filter out rows which reflect the “Number of correctly produced words” measure so that only the “Average Frequency” variable is included.
10) Check if any outliers are filtered out.
11) Run descriptives and visualise data
12) Create Helmert contrasts for the variable Age Category, with middle-aged adults as first factor level, so that the first contrast in the model output refers to middle-aged vs. younger adults, and the second contrast will refer to older adults vs. the average between younger and middle-aged adults.
13) Run unconditional (i.e., without covariate) and full model.
14) Check assumptions multiple linear regression (linearity, independence of predictor variables, normal distribution of residuals, homoscedasticity).
15) Run model fit diagnostics (Variation Inflation Factor, observed vs. predicted plot)
16) Run model comparisons for the CR measure preceding and coinciding with the covid-19 pandemic using anova’s