mci_style_neu_rating_analysis.R

2024-02-22

```r
## RATING DATA ANALYSES ##
# Script reads-in context story rating data, provides descriptive data and
# computes statistical analyses.
# For quality control, 32 participants rated the context stories for
# "fairytaleness" on a 5-point Likert style. We expected fairy-tale
# context stories to receive higher ratings than the unmarked versions.

## Setup ## ------------------------------------------------
# Encoding
Sys.setlocale("LC_ALL", "de_DE.UTF-8")
```

```
## [1] "de_DE.UTF-8/de_DE.UTF-8/de_DE.UTF-8/C/de_DE.UTF-8/en_US.UTF-8"
```

```r
# Load packages
library(MASS)         # Version 7.3-51.6
library(lme4)         # Version 1.1-23
```

```
## Loading required package: Matrix
```

```r
library(lmerTest)     # Version 3.1-2
```

```
##
## Attaching package: 'lmerTest'

## The following object is masked from 'package:lme4':
##
##     lmer
## The following object is masked from 'package:stats':
##
##     step
```

```r
library(afex)         # Version 0.27-2
```

```
## ************
## Welcome to afex. For support visit: http://afex.singmann.science/

## - Functions for ANOVAs: aov_car(), aov_ez(), and aov_4()
## - Methods for calculating p-values with mixed(): 'S', 'KR', 'LRT', and 'PB'
## - 'afex_aov' and 'mixed' objects can be passed to emmeans() for follow-up tests
## - Get and set global package options with: afex_options()
## - Set sum-to-zero contrasts globally: set_sum_contrasts()
## - For example analyses see: browseVignettes("afex")
## ************

##
## Attaching package: 'afex'

## The following object is masked from 'package:lme4':
##
##     lmer
```

```r
library(emmeans)      # Version 1.4.8
library(tidyverse)    # Version 1.3.0
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x dplyr::select() masks MASS::select()
## x tidyr::unpack() masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
## Read-in raw files ## -------------------------------------
# Read in file names
files_full <- list.files(list.dirs(list.dirs(
  path=here::here("ratings", "raw", "rating_data"), recursive=T),
  recursive=F), full.names=T)

# Bind all files into one
df <- data.frame()
for(i in 1:length(files_full)){
  if(endsWith(files_full[i], ".csv")){
    temp <- read.csv(files_full[i])
    df <- rbind(df, temp)
  }
}

## Language check ## ------------------------------------------
# Check whether all participants indicated German as their mother tongue
df %>% group_by(language) %>% count()
```

```
## # A tibble: 2 x 2
## # Groups:   language [2]
##   language                  n
##   <chr>                 <int>
## 1 Deutsch                1450
## 2 unter anderem Deutsch   250
```

```
## Quality check ## -------------------------------------------
# Make sure none of the participants had zero variance in the ratings
df %>% group_by(subject) %>%
  summarise(mean = mean(rating),
            sd = sd(rating),
            n = sum(!is.na(rating))) %>%
  filter(sd == 0)
```

```
## # A tibble: 0 x 4
## # i 4 variables: subject <int>, mean <dbl>, sd <dbl>, n <int>
```

```
## Subset data to final set ## -------------------------------
# Check whether 50% of the participants saw version A and B
df %>% group_by(rating_version) %>% count() %>% mutate(n=n/50)
```

```
## # A tibble: 2 x 2
## # Groups:   rating_version [2]
##   rating_version     n
##   <chr>          <dbl>
## 1 versionA          16
## 2 versionB          18
```

```
# We planned 16 ratings per story each: Select the first 16 data
# sets in version A and version B
df %>% filter(rating_version=="versionA") %>%
  count(subject) %>% select(subject) %>% arrange(subject) -> a
a <- a[1:16,]
df %>% filter(rating_version=="versionB") %>%
  count(subject) %>% select(subject) %>% arrange(subject) -> b
b <- b[1:16,]
df %>% filter((subject %in% a | subject %in% b) & !is.na(subject)) -> df
df %>% group_by(rating_version) %>% summarise(length(unique(subject)))
```

```
## # A tibble: 2 x 2
##   rating_version ‘length(unique(subject))‘
##   <chr>                               <int>
## 1 versionA                               16
## 2 versionB                               16
```

```
## Participant descriptives ## ------------------------------
# gender
```

```r
df %>% filter(presentation_order==1) %>%
  group_by(gender) %>%
  count()
```

```
## # A tibble: 3 x 2
## # Groups:   gender [3]
##   gender       n
##   <chr>    <int>
## 1 divers       1
## 2 männlich    10
## 3 weiblich    21
```

```r
# age
df %>% filter(presentation_order==1) %>%
  #group_by(age) %>%
  summarise(
    mean=mean(age),
    sd=sd(age),
    min=min(age),
    max=max(age))
```

```
##      mean       sd min max
## 1 32.4375 16.28563  17  84
```

```r
# language
(df %>% filter(presentation_order==1) %>%
  group_by(language) %>%
  count() -> x)
```

```
## # A tibble: 2 x 2
## # Groups:   language [2]
##   language                  n
##   <chr>                 <int>
## 1 Deutsch                  27
## 2 unter anderem Deutsch     5
```

```r
(no_of_German_speakers <- x$n[x$language=="Deutsch"] +
  x$n[x$language=="unter anderem Deutsch"])
```

```
## [1] 32
```

```r
## Clean data frame and export # -------------------------------
# replace jatos ID by subject numbers
df$subject <- as.numeric(factor(df$subject, levels=unique(df$subject)))
# select only relevant columns
df <- df %>% select(subject, presentation_order, rating_version,
                    presentation_order, context_no, version, rating) %>%
  arrange(subject, presentation_order)
# save as csv
write.csv(df, here::here("ratings", "raw", "cleaned", "rating_results.csv"),
          col.names = F)
```

```
## Warning in write.csv(df, here::here("ratings", "raw", "cleaned",
## "rating_results.csv"), : attempt to set ’col.names’ ignored
```

```r
## DESCRIPTIVES OF RATING DATA
# Descriptives: All context stories
df <- read.csv2(here::here("ratings", "raw", "cleaned", "rating_results.csv"),
                sep=",")
df %>% group_by(version) %>%
  summarise(mean=mean(rating),
            sd=sd(rating),
            min=min(rating),
            max=max(rating))
```

```
## # A tibble: 2 x 5
##   version    mean    sd   min   max
##   <chr>     <dbl> <dbl> <int> <int>
## 1 fairytale  4.10 0.999     1     5
## 2 unmarked   1.85 0.997     1     5
```

```r
# Descriptives: Separately for all context stories
df %>% group_by(context_no, version) %>%
  summarise(mean=mean(rating),
            sd=sd(rating),
            min=min(rating),
            max=max(rating))
```

```
## ‘summarise()‘ has grouped output by ’context_no’. You can override using the
## ‘.groups‘ argument.
```

```
## # A tibble: 100 x 6
## # Groups:   context_no [50]
##    context_no version    mean    sd   min   max
##         <int> <chr>     <dbl> <dbl> <int> <int>
##  1          1 fairytale  4.25 0.931     2     5
##  2          1 unmarked   1.44 0.629     1     3
##  3          2 fairytale  4.75 0.447     4     5
##  4          2 unmarked   1.38 0.719     1     3
##  5          3 fairytale  3.56 1.21      1     5
##  6          3 unmarked   2.25 0.931     1     4
##  7          4 fairytale  4.56 0.814     2     5
##  8          4 unmarked   1.69 0.793     1     3
##  9          5 fairytale  4.19 0.911     2     5
## 10          5 unmarked   1.56 0.629     1     3
## # i 90 more rows
```

```r
## MIXED MODEL ANALYSES
# Preparation
df$style <- factor(df$version, levels=c("unmarked", "fairytale"))
t(contrasts.style <- t(cbind(c("unmarked" = -1, "fairytale" = 1))))
```

```
##           [,1]
## unmarked    -1
## fairytale    1
```

```
contrasts(df$style) <- ginv(contrasts.style)
df$rating <- scale(df$rating, center=T, scale=F)
df$subject <- factor(df$subject)
df$context_no <- factor(df$context_no)
control_params <- lmerControl(calc.derivs = FALSE,
                              optimizer = "bobyqa",
                              optCtrl = list(maxfun = 2e5))

# LMM for fairytaleness-rating
mod_rating <- lmer_alt(
  rating ~ style + (style||subject) + (style||context_no),
  data = df, control = control_params)
#summary(mod_rating)
anova(mod_rating)
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##       Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## style 122.37  122.37     1 51.461  227.65 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```