贝叶斯因子序列分析在 R 语言中的实现

郑元瑞　　胡传鹏

目录

1 下载和安装需要的 R 语言程序包

```
# install.packages(c("tidyverse", "BayesFactor", "here"))
library(BayesFactor)# 计算 T 检验和方差分析的贝叶斯因子
library(tidyverse)
library(here)
here()
```

```
## [1] "/Users/zhengyuanrui/SFBA_tutorial"
```

2 导入数据

```
df <- read_csv(here("2_Data", "df.js.sum_jasp.csv"))
```

```
## Rows: 20 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): subj_idx
## dbl (9): Match_Bad, Match_Good, Match_Neutral, Mismatch_Bad, Mismatch_Good, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#T 检验使用数据
```

2.1 数据长宽数据转换

```
df_anova <- df %>%
  select(subj_idx:Mismatch_Good) %>%
  pivot_longer(
    cols = Match_Bad:Mismatch_Good,
    names_to = c("Matchness", "Valence"),
    names_sep = "_", values_to = "rt"
  )#anova 使用数据
```

3 查看数据的被试信息

```
subj_num <- unique(df$subj_idx) # 每个被试的编号
n <- length(unique(df$subj_idx)) # 一共有 20 个被试
```

#（配对样本）T 检验的 R 语言实现 ## good_match 条件与 bad_match

条件的对比先建立一个空的列表，用来储存后续的贝叶斯因子。列表长度为目前数据的样本量

```
bf_output <- rep(NA, length(subj_num)) ### 先建立一个列表
```

```
for (i in seq_along(subj_num)) {#i 遍历 subj_num
  if (i == 1) {
    next
    # 由于一个被试不能正确计算贝叶斯因子，所以当 i 等于 1 时，跳过
  }
  # 将 df 数据框中的 subj_idx 列转换为字符串型
  df$subj_idx <- as.character(df$subj_idx)
  # 提取出遍历到的被试编号
  id <- unique(df$subj_idx)[1:i]
  # 从愿数据中筛选被试
  df.selected <- df %>% filter(subj_idx %in% id)
  # 转换为因子型
  df.selected$subj_idx <- as.factor(df.selected$subj_idx)
  bayesfactors <- ttestBF(
    x = df.selected$Match_Bad, y = df.selected$Match_Good,
    paired = TRUE
  )# 计算贝叶斯因子
  bf_output[i] <- bayesfactors[1]
}
```

```
bf_output
```

```
##  [1]          NA    1.079341    2.095059    3.339921    6.652698   19.097998
##  [7]   31.855000   49.752936  130.961355   58.188628   93.428773  239.350270
## [13]  127.216734  250.586118  571.531998 1522.706898 2032.527877 2787.948423
## [19] 5664.722345 3113.214090
```

4 （重复测量）方差分析的 R 语言实现

4.1 数据的基本信息

```
subj_num <- unique(df_anova$subj_idx) # 每个被试的编号
n <- length(unique(df_anova$subj_idx)) # 一共有 20 个被试
```

生成三个列表用来储存两个主效应和交互项

```
BFs_match <- rep(NA, length(subj_num))

BFs_valence <- rep(NA, length(subj_num))

BFs_int <- rep(NA, length(subj_num))
```

```
for (i in seq_along(subj_num)) {
  if (i == 1) {
    next
  }
  df_anova$subj_idx <- as.character(df_anova$subj_idx)
  id <- unique(df_anova$subj_idx)[1:i]
  df.selected <- df_anova %>% filter(subj_idx %in% id)
  df.selected$subj_idx <- as.factor(df.selected$subj_idx)
  df.selected$Matchness <- as.factor(df.selected$Matchness)
  df.selected$Valence <- as.factor(df.selected$Valence)
  bayesfactors <- bf <- anovaBF(rt ~ Valence*Matchness + subj_idx,
    data = data.frame(df.selected),
    whichRandom = "subj_idx"
  )
  BFs_match[i] <- bayesfactors[1]
  BFs_valence[i] <- bayesfactors[2]
  BFs_int[i] <- bayesfactors[4] / bayesfactors[3]
}
```

```
BFs_match
```

```
##  [1]        NA  0.8789661  0.8443724  0.6834420  1.2285291  1.1077484
##  [7]   2.0256113   2.6630755   4.3115568  11.2786746   3.8717980   9.6159303
## [13]  13.6730882  16.3692466  27.4418157  33.2735564  40.2709205  95.8151040
## [19] 129.9870582 413.5490633
```

```
BFs_valence
```

```
##  [1]          NA   0.5299612   0.7117524   2.3793472   2.0289827   5.0632267
##  [7]   6.4834859   9.4154313  20.4873504  12.1327746  47.6669823  56.3116610
## [13]  26.5288909  29.9348935  37.4425330  81.6525643 113.9471261 118.5246519
## [19] 244.0687375 120.9679650
```

```
BFs_int
```

```
##  [1]           NA 1.485518e+00 3.341815e+00 4.647112e+00 7.531493e+00
##  [6] 3.621552e+01 7.544977e+01 1.725721e+02 5.686858e+02 2.541384e+02
## [11] 5.151070e+02 1.783258e+03 1.759747e+03 5.258070e+03 1.519619e+04
## [16] 6.720935e+04 6.922442e+04 1.040784e+05 2.281637e+05 7.793509e+04
```