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
```

```
## 载入需要的程辑包：coda
```

```
## 载入需要的程辑包：Matrix
```

```
## ************
## Welcome to BayesFactor 0.9.12-4.3. If you have questions, please contact Richard Mor
##
## Type BFManual() to open the manual.
```

```
## ************
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- 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 tidyr::unpack() masks Matrix::unpack()
```

```
# library(here)
```

2 导入数据

```
df <- read_csv("/Users/zhengyuanrui/Desktop/SBFA/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   0.9093042   0.8655600   0.7014032   1.1991076   1.1149060
##  [7]   2.0060562   2.6873412   4.2852944  11.2008425   3.8648072   9.7589807
## [13]  13.9544256  18.0472772  27.7213725  33.4998854  40.2743944  95.1521970
## [19] 131.4854521 400.2762456
```

```
BFs_valence
```

```
##  [1]          NA   0.5367660   0.7057221   2.3770512   2.0285497   5.0915627
##  [7]   6.5876690   9.5020914  20.3369769  12.7270155  48.8438452  56.7354946
## [13]  26.1812938  29.6711418  38.2317769  82.3386615 147.4454763 118.9068572
## [19] 247.2357395 122.3666839
```

```
BFs_int
```

```
##  [1]           NA 1.543211e+00 3.694320e+00 5.074814e+00 6.928469e+00
##  [6] 4.006296e+01 7.350215e+01 1.660965e+02 5.892989e+02 2.479563e+02
## [11] 5.720678e+02 1.912574e+03 1.835334e+03 5.449767e+03 1.491577e+04
## [16] 7.541347e+04 7.334727e+04 1.115412e+05 2.082175e+05 6.840184e+04
```