Parametric Bootstrap LMM

Anh Nguyen

2023-11-22

Data

```r
music <- read.csv("https://sta279-s22.github.io/labs/music.csv")
```

Library

```r
library(lme4)
library(pbkrtest)
library(tidyverse)
```

Question 1

Test the hypothesis:

- $H_0 : \beta_4 = 0$
- $H_0 : \beta_4 \neq 0$

The reduced model is

$$\text{Anxiety}_{ij} = \beta_0 + \beta_1 \text{ JuriedPerformance}$$

Question 2

```r
# fit full model
full <- lmer(na ~ audience + large + (1|id), data = music)
summary(full)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: na ~ audience + large + (1 | id)
##    Data: music
##
## REML criterion at convergence: 2956.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9124 -0.6560 -0.1830  0.4635  3.9975
```

```
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  6.063   2.462
##  Residual             20.290   4.504
## Number of obs: 497, groups:  id, 37
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)                  14.9784     0.5642  26.549
## audienceJuried Recital        3.7572     0.8086   4.647
## audiencePublic Performance    2.2008     0.6518   3.377
## audienceStudent(s)            3.1937     0.6216   5.138
## large                        -2.2251     0.6501  -3.423
##
## Correlation of Fixed Effects:
##             (Intr) adncJR adncPP adnS()
## adncJrdRctl -0.345
## adncPblcPrf -0.426  0.289
## adncStdnt() -0.429  0.300  0.462
## large       -0.027  0.031 -0.562 -0.121
```

```r
# fit reduced model
reduced <- lmer(na ~ audience + (1|id), data = music)
summary(reduced)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: na ~ audience + (1 | id)
##    Data: music
##
## REML criterion at convergence: 2968.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9962 -0.6657 -0.2006  0.4571  4.2379
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  id       (Intercept)  5.599   2.366
##  Residual             20.852   4.566
## Number of obs: 497, groups:  id, 37
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)                  14.9288     0.5560  26.849
## audienceJuried Recital        3.8268     0.8183   4.677
## audiencePublic Performance    0.9454     0.5452   1.734
## audienceStudent(s)            2.9242     0.6246   4.682
##
## Correlation of Fixed Effects:
##             (Intr) adncJR adncPP
## adncJrdRctl -0.353
## adncPblcPrf -0.547  0.370
## adncStdnt() -0.448  0.306  0.480
```

```r
# extract variance
var_reduced <- as.data.frame(VarCorr(reduced))$vcov
var_reduced
```

```
## [1]  5.598533 20.851675
```

Question 3

```r
# observed F stats
obs_stat <- KRmodcomp(full, reduced)$stats$Fstat
obs_stat
```

```
## [1] 11.63118
```

Question 4

```r
# resample group variance
re_new <- rnorm(n = unique(music$id), mean = 0, sd = var_reduced[1])
```

Question 5

```r
# resample individuals variance
noise_new <- rnorm(n = nrow(music), mean = 0, sd = var_reduced[2])
```

```r
fitted_values <- predict(reduced, re.form=NA)
re_data <- data.frame(id = unique(music$id),
                      re = re_new) %>%
  right_join(dplyr::select(music, id), by = "id")
new_data <- data.frame(id = music$id,
                       audience = music$audience,
                       large = music$large,
                       na = fitted_values + re_data$re + noise_new)
```

Question 6

```r
full_sim <- lmer(na ~ audience + large + (1|id), data = new_data)
reduced_sim <- lmer(na ~ audience + (1|id), data = new_data)
```

Question 7

```r
KRmodcomp(full_sim, reduced_sim)$stats$Fstat
```

```
## [1] 0.3469397
```

Question 8

```
nsim <- 500
f_stats <- rep(NA, nsim)

for(sim in 1:nsim){
  # code from steps 2 and 3 goes here!
  # bootstrap new data
  re_new <- rnorm(n = unique(music$id), mean = 0, sd = var_reduced[1])
  noise_new <- rnorm(n = nrow(music), mean = 0, sd = var_reduced[2])
  fitted_values <- predict(reduced, re.form=NA)
  re_data <- data.frame(id = unique(music$id),
                        re = re_new) %>%
    right_join(dplyr::select(music, id), by = "id")
  new_data <- data.frame(id = music$id,
                         audience = music$audience,
                         large = music$large,
                         na = fitted_values + re_data$re + noise_new)
  # refit model
  full_sim <- lmer(na ~ audience + large + (1|id), data = new_data)
  reduced_sim <- lmer(na ~ audience + (1|id), data = new_data)
  # remember to save the results in f_stats
  f_stats[sim] <- KRmodcomp(full_sim, reduced_sim)$stats$Fstat
}
```

```
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

Question 9

```
mean(f_stats > obs_stat)
```

```
## [1] 0.002
```

The p-value is 0, so we fail to reject the null hypothesis. There is no evidence for a difference in anxiety levels between large and small ensemble performances, after accounting for audience type.

Question 10

```
KRmodcomp(full, reduced)
```

```
## large : na ~ audience + large + (1 | id)
```

```
## small : na ~ audience + (1 | id)
##          stat     ndf     ddf F.scaling  p.value    
## Ftest  11.631   1.000 491.113         1 0.000702 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value for from the chi-squared distribution is only 10^{-3} order away from the p-value from the bootstrap. If we increase the number of simulation, the p-value might be the same.

Question 11

The p-values for the bootstrap can take any values from 0 to somewhere in the 10^{-2} order. This is because we are averaging over 500 simulations. If we increase our simulation, we can get more significant figures for our p-values.