Nested ANOVA and MANOVA

Rui Zhou

目录

1 这章在讲什么

普通单因素 ANOVA 假设组与组之间独立、组内观测也独立。但心理学实验里数据经常**有层级结构**：每个学校抽几个班，每个班抽几个学生——「学

生」嵌套在「班」里，「班」嵌套在「学校」里。如果不显式建模这种嵌套，就会低估误差、把 p 值算小、得出虚高的显著性。这一章讲两件事：**嵌套 ANOVA**（一个因变量，但变量有嵌套层级）和 **MANOVA**（多个因变量同时考察组间差异）。

两个核心区别

Nested ANOVA：误差项要拆——「组间」的检验要用「子组间」的方差作分母，而不是「子组内」。R 里 `aov(DV ~ group + Error(group/subgroup))` 这种 `Error()` 项就是干这件事。

MANOVA：把多个 DV 看成一个向量 $\mathbf{Y}_i$，检验「组别」是否对 $\mathbf{Y}$ 的多维分布有影响。统计量从 F 换成 Pillai's trace、Wilks' Λ、Hotelling-Lawley、Roy's largest root——这四个统计量各有侧重，Pillai 最稳健（对协方差齐性违反不敏感），是默认选择。MANOVA 显著之后再做 univariate ANOVA 看是哪个 DV 在驱动差异。

2 Nested ANOVA

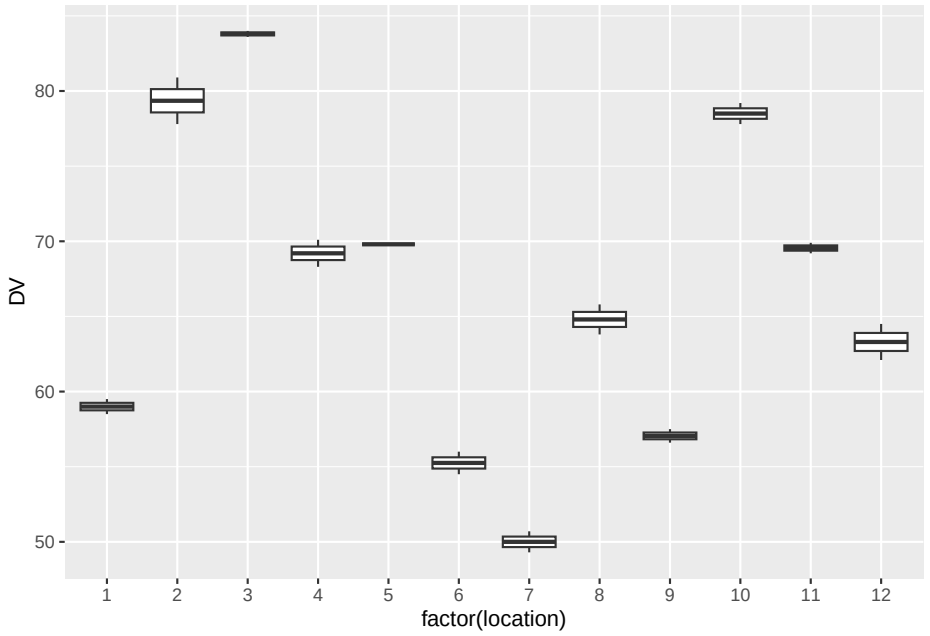
2.1 0

```
library(tidyverse)
library(car)

nestedData = readxl::read_excel("nested anova.xlsx")
```

2.2 visualization

```
# we should eyeball the data first to make sense of the test.
library(ggplot2)
# create boxplots to visualize the data
ggplot(nestedData, aes(x=factor(location), y=DV)) +
  geom_boxplot()
```



2.3 nested ANOVA

Theoretically, the order of computing the variance sum of squares matters, thus appointing the `summation = 'I'`.

However, here the summation type does not matter.

Note that the result of the main effect of the group is wrong. We can manually make it straight from the SS of the group and the subgroup.

```
# note that nested anova can use error type I and III.
# aov() has a default type I to process factors in an sequential order
# here type I and III don't differ, since effectively we have no orders
aov_model = aov(DV ~ factor(method)/factor(location),
                data = nestedData,summation="I")
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'summation' will be disregarded
```

```
summary(aov_model)
```

```
##                                 Df Sum Sq Mean Sq F value   Pr(>F)
## factor(method)                   2  665.7   332.8   255.7 1.45e-10 ***
## factor(method):factor(location)  9 1720.7   191.2   146.9 6.98e-11 ***
## Residuals                       12   15.6     1.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
aov_model = aov(DV ~ factor(method)/factor(location),
                data = nestedData,summation="III")
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'summation' will be disregarded
```

```
summary(aov_model)
```

```
##                                 Df Sum Sq Mean Sq F value   Pr(>F)
## factor(method)                   2  665.7   332.8   255.7 1.45e-10 ***
## factor(method):factor(location)  9 1720.7   191.2   146.9 6.98e-11 ***
## Residuals                       12   15.6     1.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.3.1 error adjustment

An alternatDVe refinement is to adjust the error term.

```
aov.result2 = aov(DV ~ factor(method) +
                    Error(factor(method):factor(location)),
                  data = nestedData)
```

```
## Warning in aov(DV ~ factor(method) + Error(factor(method):factor(location)), :
## Error() model is singular
```

```
summary(aov.result2)
```

```
##
## Error: factor(method):factor(location)
##                 Df Sum Sq Mean Sq F value Pr(>F)
## factor(method)  2  665.7   332.8   1.741   0.23
## Residuals       9 1720.7   191.2
##
## Error: Within
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 12  15.62   1.302
```

2.3.2 comparison with one-way

If we wrongly ignore the nested factor and use one-way ANOVA, the results would be different

Note that there's only one group factor, therefore, the type of summation does not matter.

```
aov_model = aov(DV ~ factor(method),
                data = nestedData,summation="III")
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'summation' will be disregarded
```

```
summary(aov_model)
```

```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## factor(method)  2  665.7   332.8   4.026 0.0331 *
## Residuals      21 1736.3    82.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
aov_model = aov(DV ~ factor(method),
                data = nestedData)
```

```r
summary(aov_model)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## factor(method)  2  665.7   332.8   4.026 0.0331 *
## Residuals      21 1736.3    82.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3 MANOVA

3.1 0

```r
library("readxl")
library("rstatix")
```

```
##
## Attaching package: 'rstatix'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```r
library("MVTests")
```

```
##
## Attaching package: 'MVTests'
```

```
## The following object is masked from 'package:datasets':
##
##     iris
```

```r
library("heplots")
```

```
## Loading required package: broom
```

```r
# salary data example
salaryData = readxl::read_xlsx("example.xlsx")
DV = cbind(salaryData$salary, salaryData$salbegin)
jobcat = as.factor(salaryData$jobcat)
gender = as.factor(salaryData$gender)
```

3.2 visualization

Note that in order to generate the boxplot according to the "categorical" data, "factor" type should be designated in the mapping parameter.

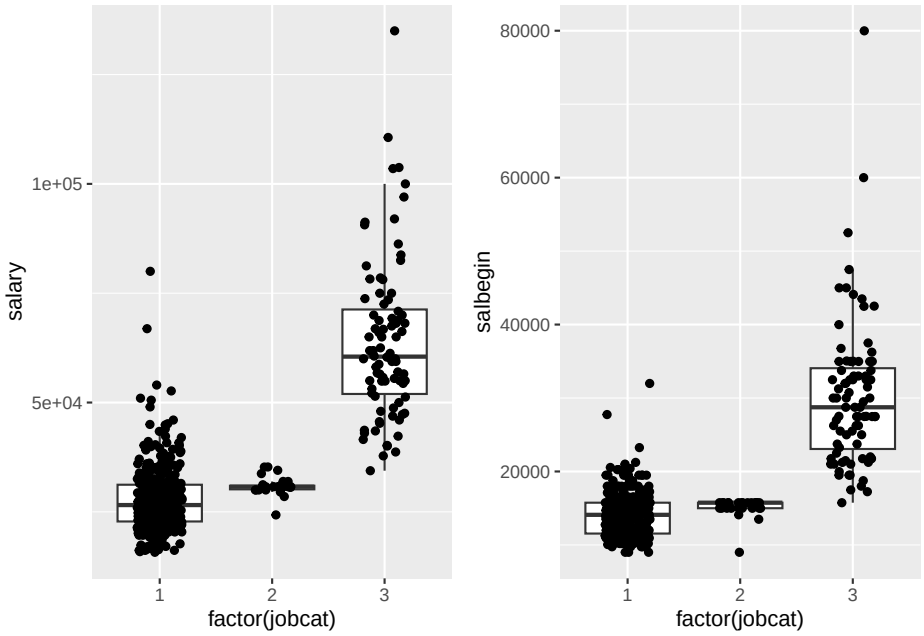
If not "factored", the scatter plot works fine with grouped effect, but the boxplot fails.

```r
# eyeball the data
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

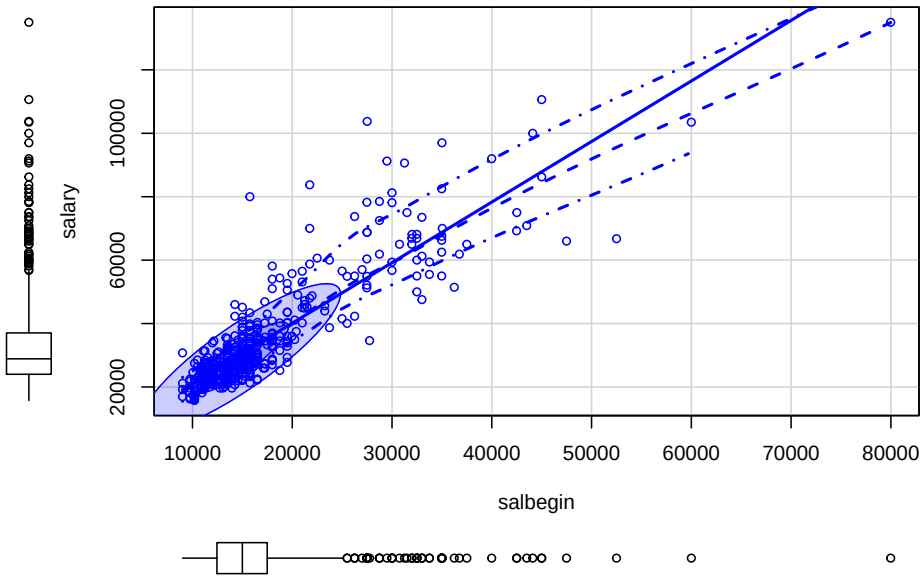
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
p1 = ggplot(salaryData, aes(x = factor(jobcat), y = salary)) +
  geom_boxplot(outlier.shape = NA) + geom_jitter(width = 0.2) +
  theme(legend.position="top")
p2 = ggplot(salaryData, aes(x = factor(jobcat), y = salbegin)) +
  geom_boxplot(outlier.shape = NA) + geom_jitter(width = 0.2) +
  theme(legend.position="top")
grid.arrange(p1, p2, ncol=2)
```



We can also eyeball the DVs.

```
library(car)
scatterplot(salary ~ salbegin, data=salaryData, ellipse=TRUE,
  smooth=list(style="lines"))
```



3.3 assumptions

3.3.1 normality

3.3.1.1 pipe-friendly (Preferred) Preferred!

Note that `mshapiro_test()` is from the package `rstatix`, one step further from `tidyverse`.

```
salaryData |> select(salary,salbegin) |> mshapiro_test()
```

```
## # A tibble: 1 x 2
##   statistic  p.value
##       <dbl>    <dbl>
## 1     0.707 6.13e-28
```

```
salaryData |> group_by(jobcat) |>
  shapiro_test(salary,salbegin)
```

```
## # A tibble: 6 x 4
##   jobcat variable statistic        p
##    <dbl> <chr>        <dbl>    <dbl>
## 1      1 salary       0.882 4.57e-16
## 2      1 salbegin     0.931 6.10e-12
## 3      2 salary       0.818 2.86e- 4
## 4      2 salbegin     0.499 1.64e- 8
## 5      3 salary       0.929 1.72e- 4
## 6      3 salbegin     0.860 2.14e- 7
```

3.3.1.2 classical (Not Preferred) Not preferred. It's not efficient to do this way.

The specific function `mshapiro.test()` requires to library the package `mvnormtest`, which surprisingly has a single function! Moreover, it requires the data to be transposed before being processed.

The most annoying part is that the data has to be packed purposefully.

```
mvnormtest::mshapiro.test(t(DV))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.70689, p-value < 2.2e-16
```

3.3.2 homogeniety of variance & covariance

Using the Box's M-test and Levene's Test.

3.3.2.1 pipe-friendly (Preferred) Using `box_m()` from the package `rstatix` to test the homogeniety of covariance matrices, in which the first parameter is the DVs needed to be analyzed (No group variable! This can be achieved through `select()`), and the second is the `group` parameter specifying the group variable.

Moreover, the function `heplots::boxM()` is helpful. Instead of taking the formula to specify the parameters, using the "traditional" way to write `group = ...` works pretty well combined with pipe-friendly environment!

The drawback of the functions is that they could not search column variables automatically, and that should be specified manually. And more annoying one is that, when encountering with NAs, an error will be raised by `rstatix::box_m()`, while that's fine with `heplots::boxM()`.

```
salaryData |> select(salary,salbegin) |>
  box_m(group = jobcat)
```

```
## # A tibble: 1 x 4
##   statistic   p.value parameter method
##       <dbl>     <dbl>     <dbl> <chr>
## 1      509. 1.14e-106         6 Box's M-test for Homogeneity of Covariance Matr~
```

```
salaryData |> select(salary,salbegin) |>
  heplots::boxM(group = jobcat)
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  select(salaryData, salary, salbegin)
## Chi-Sq (approx.) = 508.68, df = 6, p-value < 2.2e-16
```

Do not forget that the assumption of homogeniety of variance for each DV is needed either! However, the traditional `levene_test()` (or its variants) can deal with one DV at a time, which is tedious when we have many DVs to check.

Hopefully, we have `heplots::leveneTests(data,group)` to deal with that issue. Combined with pipe-friendly procedures, that's fabulous!

```
salaryData |>  select(salbegin,salary) |>
  heplots::leveneTests(group = jobcat)
```

```
## Levene's Tests for Homogeneity of Variance (center = median)
##
##          df1 df2 F value    Pr(>F)
## salbegin   2 471  68.862 < 2.2e-16 ***
## salary     2 471  51.189 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.3.2.2 classical The most annoying part is that the data has to be packed purposefully.

An alternative way to use `heplots::boxM()` has been mentioned above. We can renovate the efficiency to carry this test by replacing the "formula" specifying-pattern with the "group" one.

```
heplots::boxM(cbind(salary, salbegin) ~ factor(jobcat),
              data = salaryData)
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  Y
## Chi-Sq (approx.) = 508.68, df = 6, p-value < 2.2e-16
```

3.4 MANOVA

Use the funtion `manova()`, which is from the baseR, no special need to library packages.

Similarly, the most annoying part is that the DVs and IVs has to be packed previously.

```
fit = manova(DV ~ jobcat)
```

As before, simply use `summary()` of the fitted model to see the results.

```
summary(fit)  #for manova
```

```
##            Df  Pillai approx F num Df den Df    Pr(>F)
## jobcat      2 0.67298   119.43      4    942 < 2.2e-16 ***
## Residuals 471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.5 effect size

Checking the effect size is also indispensable.

Use `eta_squared()` from the package `effectsize`, which contains plenty of useful functions computing effect size under various scenarios.

Note that there's also an `eta_squared()` from the package `rstatix()`, but that can only deal with univariate ANOVA, it would fail when encountering MANOVA models.

Also note that the package `effectsize` is incorporated in `bruceR`!

```
effectsize::eta_squared(fit)
```

```
## # Effect Size for ANOVA (Type I)
##
## Parameter | Eta2 (partial) |       95% CI
## -----------------------------------------
## jobcat    |           0.34 | [0.30, 1.00]
##
## - One-sided CIs: upper bound fixed at [1.00].
```

3.6 univariate ANOVA

Needed if MANOVA reports significant results.

`summary.aov()` will report the univariate ANOVA respectively.

```
summary.aov(fit)  #for indDVidual anovas
```

```
##  Response 1 :
##              Df     Sum Sq    Mean Sq F value    Pr(>F)
## jobcat        2 8.9438e+10 4.4719e+10  434.48 < 2.2e-16 ***
## Residuals   471 4.8478e+10 1.0293e+08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##  Response 2 :
##              Df     Sum Sq    Mean Sq F value    Pr(>F)
## jobcat        2 1.7926e+10 8962772266  371.11 < 2.2e-16 ***
## Residuals   471 1.1375e+10   24151508
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.7 post-hoc tests

Note that the post-hoc tests here are essentially for the univariate ANOVA.

Note that the `games_howell_test()` is different from `tukey_hsd()`.

Note that for many DVs to take the post-hoc tests simultaneously, first `gather()` the targeted columns, and then take the **grouped** data for the test.

```
salaryData %>% gather(key = "var",
                      value = "value",
                      salary, salbegin) |>
  group_by(var) |>
  games_howell_test(value~jobcat)
```

```
# A tibble: 6 x 9
  var      .y.   group1 group2 estimate conf.low conf.high    p.adj p.adj.signif
* <chr>    <chr> <chr>  <chr>     <dbl>    <dbl>     <dbl>    <dbl> <chr>
1 salary   value 1      2         3100.    1746.     4455. 1.22e- 6 ****
2 salary   value 1      3        36139.   31302.    40977. 4.25e-10 ****
3 salary   value 2      3        33039.   28196.    37881. 4.21e-10 ****
4 salbegin value 1      2          982.     256.     1707. 6   e- 3 **
5 salbegin value 1      3        16162.   13539.    18784. 2.74e-10 ****
6 salbegin value 2      3        15180.   12514.    17846. 4.77e-10 ****
```

4 容易踩的坑

嵌套与 MANOVA 都容易栽的几件事

1. **嵌套与交叉（crossed）混淆**：当「班」嵌套在「学校」里时，A 校的「3 班」和 B 校的「3 班」不是同一回事，必须写成 `Error(school/class)`。反之如果是真正的两个交叉因子（如「教学法」×「教师」），应该用 `Y ~ method * teacher`。判断标准：每个 level 的子因子是否在所有上级因子里出现？
2. **MANOVA 跳过协方差齐性检查就上结果**：MANOVA 的 p 值对 $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_g$ 的同方差假设敏感。Box's M 检验给出齐性 p；若 $p < 0.001$ 且各组样本量差异大，Pillai 之外的统计量（特别是 Wilks）都不可信。
3. **MANOVA 显著就停手**：MANOVA 只告诉你「至少有一个 DV 上组间有差异」。要知道 ** 是哪个 DV** 必须再跑 univariate ANOVA，最好做 Bonferroni 校正控制族错误率。
4. **事后检验用错**：组间方差齐时用 Tukey HSD；不齐时用 Games-Howell（`rstatix::games_howell_test`）。本笔记上面演示的就是 Games-Howell，因为各职位类别样本量差异很大。