

R と RStudio の使い方

芳賀敏郎 (2014) 医薬品開発のための統計解析 第2部 実験計画法
3 乱塊法実験
3.1 質的因子の乱塊法

テキストと利用上の注意

●テキスト

芳賀敏郎（2011）医薬品開発のための統計解析

第2部 実験計画法 改訂版、サイエンティスト社、p.294

（サイトへアップすることに対して、サイエンティスト社の了解を得ています）

●Rによる解析事例を紹介

R スクリプトの出力結果を紹介します（tidyverse 系には次期バージョンで対応します）

R スクリプト（文字コードUTF-8に設定）を、このサイトから[ダウンロード](#)できます

R スクリプトを [Compile Report] することにより、Word または HTML で見ることができます

R と RStudio の設定と基本的な使い方は「[R と RStudio の使い方](#)」を参照してください

R の出力結果の見方は、テキストとそれを解説した [PDF ファイル](#) を参照してください

グラフ表示は、解析手段として、必要最小限の表現に止めています

●自己責任で利用

上記のことを理解した上で、自己責任により利用してください

第2部 実験計画法

- 1 因子実験・・・質的因子
 - 1.1 繰り返し数が等しい場合、1.2 繰り返し数が異なる場合
 - 1.3 多重比較、1.4 ばらつきを特性値とする実験
 - 1.5 ノンパラメトリック検定
- 量的因子
 - 2.1 直線関係の場合、2.2 非直線関係の場合
 - 2.3 ダミー変数による質的因子の効果の推定
- 乱塊法**・・・**3.1 質的因子の乱塊法**、3.2 量的因子の乱塊法、3.3 欠測値のある場合
- 共分散分析・・・4.1 共分散分析の目的、4.2 解析手順、4.3 医薬品開発における共分散分析の例
- 2 因子実験・・・5.1 2 因子実験の基礎、5.2 質的因子×質的因子、5.3 質的因子×量的因子
 - 5.4 質的因子×量的因子（変形）、5.5 量的因子×量的因子
- 多因子実験・・・6.1 多因子実験の基礎、6.2 スクリーニング計画、6.3 応答局面計画
- 変量モデルほか・・・7.1 1 因子実験、7.2 枝分れ実験、7.3 乱塊法の拡張、7.4 経時データ、7.5 交差試験



第2部 実験計画法

- 1 因子実験・・・質的因子
 - 1.1 繰り返し数が等しい場合、1.2 繰り返し数が異なる場合
 - 1.3 多重比較、1.4 ばらつきを特性値とする実験
 - 1.5 ノンパラメトリック検定
- 量的因子
 - 2.1 直線関係の場合、2.2 非直線関係の場合
 - 2.3 ダミー変数による質的因子の効果の推定

乱塊法実験では欠測値が発生することがある

乱塊法・・・3.1 質的因子の乱塊法、3.2 量的因子の乱塊法、3.3 欠測値のある場合

- 共分散分析・・・4.1 共分散分析
- 2 因子実験・・・5.1 2 因子実験

欠測値が発生しても対応できることを前提としてRの関数を選択
Rの結果とJMPの結果を比較できるように、同じ条件で解析

- 5.4 質的因子×量的因子（変形）、5.5 量的因子×量的因子
- 多因子実験・・・6.1 多因子実験の基礎、6.2 スクリーニング計画、6.3 応答局面計画
- 変量モデルほか・・・7.1 1 因子実験、7.2 枝分れ実験、7.3 乱塊法の拡張、7.4 経時データ、7.5 交差試験



lm 関数による乱塊法データの解析

● 表示3.17 表示3.1.9 [モデルのあてはめ]の出力

スクリプトファイル：Green2-3-1a.R

利用した関数：lm、summary、car::Anova

方法：制御因子（group）とブロック因子（block）をプラスで結び、
これと目的変数（y）とをチルダ（~）で結ぶ

JMP は「水準効果の和が0という制約」で解析しているため、
contrasts 引数で下のよう指定する（[§2.3](#)参照）

（平方和 Type III を使うための前提でもある）

car パッケージの Anova 関数（平方和 Type III に対応）を用い、

仮に欠測値が発生しても解析できるように備える

（欠測値がない場合、anova 関数（Type I）と同じ結果になる）

```
# データ (乱塊法実験)
df
##      group block    y
## 1      A1     B1 10.8
## 2      A1     B2  9.9
## 3      A1     B3  9.7
## 4      A1     B4 10.4
## 5      A1     B5 10.7
## 6      A2     B1 10.7
## 7      A2     B2 10.6
## 8      A2     B3 11.0
## 9      A2     B4 10.8
##          . . . . .
```

```
lm_out <- lm(y ~ group + block, data = df,
            contrasts = list(group = "contr.sum", block = "contr.sum"))
summary(lm_out)
car::Anova(lm_out, type = 3) # 欠測値がない場合は Type I と同じ結果になる
```



lm 関数による乱塊法データの解析

●表示3.17 表示3.1.9 [モデルのあてはめ]の出力

スクリプトファイル：Green2-3-1a.R

利用した関数：lm、summary、car::Anova

方法：制御因子（group）とブロック因子（block）をプラスで結び、
これと目的変数（y）とをチルダ（~）で結ぶ

JMP は「水準効果の和が0という制約」で解析しているため、
contasts 引数で下のよう指定する（[§2.3](#)参照）

（平方和 Type III を使うための前提でもある）

car パッケージの Anova 関数（平方和 Type III に対応）を用い、
仮に欠測値が発生しても解析できるように備える
（欠測値がない場合、anova 関数（Type I）と同じ結果になる）

```
# データ (乱塊法実験)
df
##      group block    y
## 1      A1     B1 10.8
## 2      A1     B2  9.9
## 3      A1     B3  9.7
## 4      A1     B4 10.4
## 5      A1     B5 10.7
## 6      A2     B1 10.7
## 7      A2     B2 10.6
## 8      A2     B3 11.0
## 9      A2     B4 10.8
##      . . . . .
```

car::Anova 関数のヘルプの記述

The designations "type-II" and "type-III" are borrowed from SAS,
but the definitions used here do not correspond precisely to those employed by SAS.



lm 関数による乱塊法データの解析

- 表示3.17 表示3.1.9 [モデルのあてはめ]の出力

スクリプトファイル：Green2-3-1a.R

利用した関数

lm、summary

$$\text{group4} = -(\text{group1} + \text{group2} + \text{gropu3})$$

$$\text{block5} = -(\text{block1} + \text{block2} + \text{block3} + \text{block4})$$

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.090e+01  6.325e-02 172.344 < 2e-16 ***
## group1      -6.000e-01  1.095e-01  -5.477 0.000141 ***
## group2     -1.000e-01  1.095e-01  -0.913 0.379286
## group3      3.000e-01  1.095e-01   2.739 0.017977 *
## block1      3.000e-01  1.265e-01   2.372 0.035292 *
## block2     -3.000e-01  1.265e-01  -2.372 0.035292 *
## block3     -2.500e-01  1.265e-01  -1.976 0.071550 .
## block4      1.167e-15  1.265e-01   0.000 1.000000
## ---
## Residual standard error: 0.2828 on 12 degrees of freedom
## Multiple R-squared:  0.8182, Adjusted R-squared:  0.7121
## F-statistic: 7.714 on 7 and 12 DF,  p-value: 0.001181
```

全水準の推定値

あてはめの要約

- 表示3.17 表示3.1.9 [モデルのあてはめ]の出力

スクリプトファイル：Green2-3-1a.R

利用した関数

lm、car::Anova、anova

欠測値がないため
Type I と同じ結果

car::Anova の結果

```
## Anova Table (Type III tests)
##
## Response: y
##              Sum Sq Df    F value    Pr(>F)
## (Intercept) 2376.20  1 29702.5000 < 2.2e-16 ***
## group         3.10   3   12.9167 0.0004576 ***
## block         1.22   4    3.8125 0.0317801 *
## Residuals    0.96  12
```

anova の結果

```
## Analysis of Variance Table
##
## Response: y
##              Df Sum Sq Mean Sq F value    Pr(>F)
## group         3    3.10  1.0333 12.9167 0.0004576 ***
## block         4    1.22  0.3050  3.8125 0.0317801 *
## Residuals   12    0.96  0.0800
```

Type I

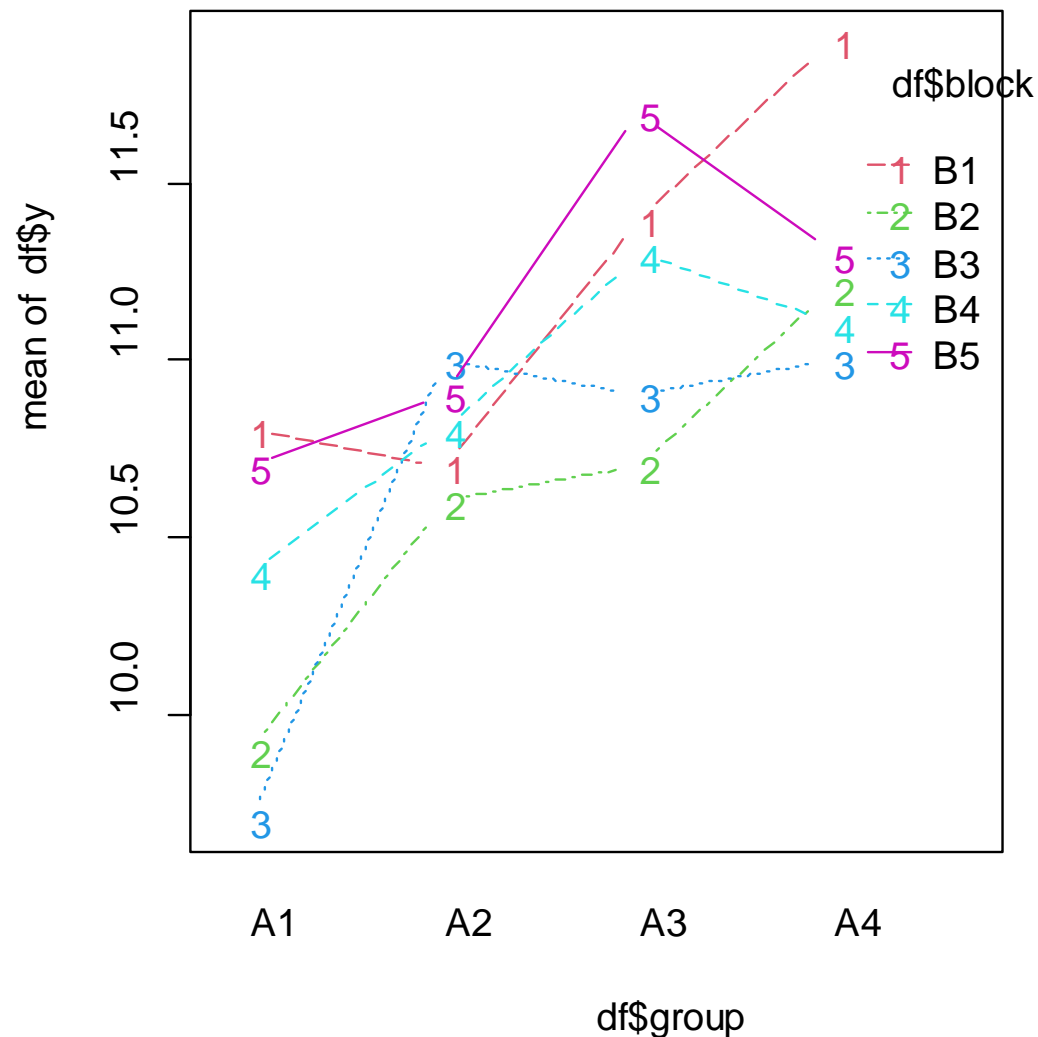
グラフ化

●表示3.1.2 データのグラフ化 (左)

スクリプトファイル：Green2-3-1a.R

利用した関数：interaction.plot

```
interaction.plot(x.factor = df$group,
                trace.factor = df$block,
                response = df$y,
                fixed = TRUE,
                col = 2:6,
                type = "b")
```



- 表示3.1.5 平均値の差の検定

スクリプトファイル

Green2-3-1a.R

利用した関数

lm、emmeans::emmeans、
pairs、multcomp::cld

方法

多重性を考慮しない比較
であることに留意

```
emmeans_out1 <- emmeans(lm_out, specs = "group")  
pairs(emmeans_out1, adjust = "none")
```

```
## contrast estimate SE df t.ratio p.value  
## A1 - A2 -0.5 0.179 12 -2.795 0.0162  
## A1 - A3 -0.9 0.179 12 -5.031 0.0003  
## A1 - A4 -1.0 0.179 12 -5.590 0.0001  
## A2 - A3 -0.4 0.179 12 -2.236 0.0451  
## A2 - A4 -0.5 0.179 12 -2.795 0.0162  
## A3 - A4 -0.1 0.179 12 -0.559 0.5864
```

```
cld(emmeans_out1, adjust = "none",  
sort = TRUE, reversed = TRUE, Letters = letters)
```

```
## group emmean SE df lower.CL upper.CL .group  
## A4 11.3 0.126 12 11.0 11.6 a  
## A3 11.2 0.126 12 10.9 11.5 a  
## A2 10.8 0.126 12 10.5 11.1 b  
## A1 10.3 0.126 12 10.0 10.6 c
```

- 表示3.1.5 平均値の差の検定（補足）

スクリプトファイル

Green2-3-1a.R

利用した関数

lm、emmeans::emmeans、

pairs、multcomp::cld

方法

多重比較（Tukey のHSD 検定）

```
emmeans_out2 <- emmeans(lm_out, specs = "group")
pairs(emmeans_out2, adjust = "tukey")

## contrast estimate SE df t.ratio p.value
## A1 - A2 -0.5 0.179 12 -2.795 0.0674
## A1 - A3 -0.9 0.179 12 -5.031 0.0014
## A1 - A4 -1.0 0.179 12 -5.590 0.0006
## A2 - A3 -0.4 0.179 12 -2.236 0.1687
## A2 - A4 -0.5 0.179 12 -2.795 0.0674
## A3 - A4 -0.1 0.179 12 -0.559 0.9423
##
cld(emmeans_out2, adjust = "tukey",
    sort = TRUE, reversed = TRUE, Letters = letters)

## group emmean SE df lower.CL upper.CL .group
## A4 11.3 0.126 12 10.93 11.7 a
## A3 11.2 0.126 12 10.83 11.6 a
## A2 10.8 0.126 12 10.43 11.2 ab
## A1 10.3 0.126 12 9.93 10.7 b
```

- 表示3.1.5 平均値の差の検定（補足）

スクリプトファイル

Green2-3-1a.R

利用した関数

lm、emmeans::emmeans、
方法

多重比較（Dunnettの検定）

ヘルプの注

The default multiplicity
adjustment method is
"dunnett", a close
approximation to
the Dunnett adjustment.

対照群との多重比較を
指定

比較対照群を指定
事例で、1は「A1」

```
emmeans(lm_out, specs = trt.vs.ctrl1 ~ "group", ref = 1)
##
## $contrasts
##   contrast estimate      SE df t.ratio p.value
##   A2 - A1         0.5 0.179 12  2.795  0.0426
##   A3 - A1         0.9 0.179 12  5.031  0.0008
##   A4 - A1         1.0 0.179 12  5.590  0.0003
##
## Results are averaged over the levels of: block
## P value adjustment: dunnett method for 3 tests
```

Friesman の検定 (ノンパラメトリック検定)

p.130

● 表示3.4.1 Friedman の検定 (補足)

スクリプトファイル

Green2-3-1b.R

利用した関数

friedman.test、unstack、
as.matrix

```
# データ (乱塊法実験)
df
##      group block    y
## 1      A1     B1 10.8
## 2      A1     B2  9.9
## 3      A1     B3  9.7
## 4      A1     B4 10.4
## 5      A1     B5 10.7
## 6      A2     B1 10.7
## 7      A2     B2 10.6
## 8      A2     B3 11.0
## 9      A2     B4 10.8
## . . . . .
```



データフレームから
マトリックスに変換

```
df1 <- unstack(df, y ~ group)
mx <- as.matrix(df1)
mx
##           A1  A2  A3  A4
## [1,] 10.8 10.7 11.4 11.9
## [2,]  9.9 10.6 10.7 11.2
## [3,]  9.7 11.0 10.9 11.0
## [4,] 10.4 10.8 11.3 11.1
## [5,] 10.7 10.9 11.7 11.3
```

ブロックの対応がある

```
friedman.test(mx)
```

```
## Friedman rank sum test
```

```
## data:  mx
```

```
## Friedman chi-squared = 10.224, df = 3, p-value = 0.01675
```

- 表示3.4.1 Friedman の検定（補足）

スクリプトファイル

Green2-3-1b.R

利用した関数

pairwise.wilcox.test

方法

多重性を考慮しない方法

p.adj 引数に "none"

多重性を考慮した方法

(Holm の方法で調整)

p.adj 引数に "holm"

```
# 多重性を考慮しない Wilcoxon の符合付き順位検定
pairwise.wilcox.test(df$y, df$group, paired = TRUE,
                     p.adj = "none")

## Pairwise comparisons using Wilcoxon signed rank test
##      A1      A2      A3
## A2 0.125 -      -
## A3 0.062 0.136 -
## A4 0.062 0.100 0.588

# Holm の方法で調整した Wilcoxon の符合付き順位検定
pairwise.wilcox.test(df$y, df$group, paired = TRUE,
                     p.adj = "holm")

## Pairwise comparisons using Wilcoxon signed rank test
##      A1      A2      A3
## A2 0.40 -      -
## A3 0.38 0.40 -
## A4 0.38 0.40 0.59
```



- 作成 片瀬雅彦
- 作成時期 2021年6月5日