

# R と RStudio の使い方

芳賀敏郎 (2016) 医薬品開発のための統計解析 第3部 非線形モデル  
3 計数値の解析  
3.3 割合の差の推定・検定と区間推定

# テキストと利用上の注意

---

## ●テキスト

芳賀敏郎（2016）医薬品開発のための統計解析

第3部 非線形モデル 改訂版、サイエンティスト社、p.288

（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した（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

本PDF ファイルをダウンロードし、Adobe Acrobat Reader DC で表示してください

## ●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3部 非線形モデル

---

## 1. 非線形最小2乗法（基礎）

- 1.1 線形と非線形、1.2 非線形最小2乗法の基本的な考え方、1.3 指数曲線のあてはめ、  
1.4 Emaxモデルとロジスティック曲線

## 2. 非線形最小2乗法（応用）

- 2.1 誤差を考慮した解析、2.2 効力比、2.3 併用効果（相乗・拮抗交換）、  
2.4 モデルの探索（複数の曲線の同時あてはめ）、2.5 薬物動態の解析

## 3. 計数値の解析

- 3.1 2項分布、3.2 割合の推定・検定と区間推定、**3.3 割合の差の推定・検定と区間推定**、  
3.4 多項分布（名義尺度）、3.5 多項分布（順序尺度）、3.6 要因が複数の場合

## 4. ロジスティック回帰分析

- 4.1 復習、4.2 ロジスティック回帰分析（基本）、4.3 ロジスティック回帰分析（応用）



## 3.3 割合の差の推定・検定と区間推定

### 2 × 2 分割表

分割表とそのグラフ

割合、オッズ、対数オッズ（ロジット）

割合の差、割合の比、オッズ比

割合の差の検定と区間推定

対応がある二値データ（補足）

glm 関数の使い方

名義ロジスティックのあてはめ（2つの割合）

### a × 2 分割表

名義ロジスティックのあてはめ（3つ以上の割合）

割合の差の検定（補足）

割合の差の検定の多重比較（補足）

2 × 2 分割表（2つの割合）

##	event			
##	group	yes	no	
##	A0	4	11	4/15
##	A1	8	5	8/13

a × 2 分割表（3つ以上の割合）

##	event			
##	group	yes	no	
##	A0	4	11	4/15
##	A1	8	5	8/13
##	A2	12	8	12/20
##	A3	10	6	10/16

## 3.3 割合の差の推定・検定と区間推定

### 2 × 2 分割表

分割表とそのグラフ

割合、オッズ、対数オッズ（ロジット）

割合の差、割合の比、オッズ比

割合の差の検定と区間推定

対応がある二値データ（補足）

glm 関数の使い方

名義ロジスティックのあてはめ（2つの割合）

### a × 2 分割表

名義ロジスティックのあてはめ（3つ以上の割合）

割合の差の検定（補足）

割合の差の検定の多重比較（補足）

2 × 2 分割表（2つの割合）

##	event			
##	group	yes	no	
##	A0	4	11	4/15
##	A1	8	5	8/13

a × 2 分割表（3つ以上の割合）

##	event			
##	group	yes	no	
##	A0	4	11	4/15
##	A1	8	5	8/13
##	A2	12	8	12/20
##	A3	10	6	10/16

# 2 × 2 分割表

分割表とそのグラフ

割合、オッズ、対数オッズ（ロジット）

割合の差、割合の比、オッズ比

## 2 × 2 分割表

- 表示3.3.1 2つの割合のデータ

表示3.3.4 JMP[二変量の関係]による解析結果

スクリプトファイル：Green3-3-3a.R

利用した関数

matrix、rownames、colnames  
dimnames (後述)

方法

matrix 関数で、ベクトルから  
マトリックスを作成

ベクトルの成分数は、通常

nrow × ncol に一致 (4 = 2 × 2)

生データからクロス集計して

分割表を作成 ([§3.4](#))

データ構造：マトリックスではなくテーブル

分割表と  
同じように  
配置

TRUEを指定  
行ごとに  
並べる

```
mx1 <- matrix(c( 4, 11,
                 7,  3),
              nrow = 2, ncol = 2, byrow = TRUE)
rownames(mx1) <- c("old", "new")
colnames(mx1) <- c("plus", "minus")
mx1
##      plus minus
## old    4     11
## new    7      3
```

分割表

	有効	無効	合計
旧処方	4	11	15
新処方	7	3	10



# 2 × 2 分割表

## ● 表示3.3.1 2つの割合のデータ

### 表示3.3.4 JMP[二変量の関係]による解析結果

スクリプトファイル：Green3-3-3a.R

利用した関数

matrix、rownames、colnames  
dimnames (後述)

方法

matrix 関数で、ベクトルから  
マトリックスを作成

ベクトルの成分数は、通常

nrow × ncol に一致 (4 = 2 × 2)

生データからクロス集計して

分割表を作成 ([§3.4](#))

データ構造：マトリックスではなくテーブル

分割表と同じように配置

TRUEを指定行ごとに並べる

```
mx1 <- matrix(c( 4, 11,
                 7,  3),
              nrow = 2, ncol = 2, byrow = TRUE)
rownames(mx1) <- c("old", "new")
colnames(mx1) <- c("plus", "minus")
mx1
```

```
##      plus minus
## old    4     11
## new    7      3
```

分割表

	有効	無効	合計
旧処方	4	11	15
新処方	7	3	10



# 2 × 2 分割表

## ● 表示3.3.1 2つの割合のデータ

表示3.3.4 JMP[二変量の関係]による解析結果

スクリプトファイル：Green3-3-3a.R

利用した関数

addmargins

方法

addmargins 関数を使って周辺合計を計算

第2引数により、3種類の計算を切替

テキストの事例では、中段の合計に意味がある

	有効	無効	合計
旧処方	4	11	15
新処方	7	3	10

```
## 周辺合計
addmargins(mx1) # 行と列の周辺合計

##      plus minus Sum
## old   4     11  15
## new   7     3   10
## Sum  11    14  25

addmargins(mx1, 2) # 行の周辺合計

##      plus minus Sum
## old   4     11  15
## new   7     3   10

addmargins(mx1, 1) # 列の周辺合計

##      plus minus
## old   4     11
## new   7     3
## Sum  11    14
```

第2引数  
省略

第2引数  
2:行

第2引数  
1:列

# 2 × 2 分割表

## ● 表示3.3.1 2つの割合のデータ

表示3.3.4 JMP[二変量の関係]による解析結果

スクリプトファイル：Green3-3-3a.R

利用した関数

proportions

方法

proportions 関数を使って割合を計算

第2引数により、3種類の計算を切替

テキストの事例では、中段が正しい割合

	有効	無効	合計
旧処方	4	11	15
新処方	7	3	10

## 割合

proportions(mx1)

# 総合計に対する割合

## plus minus

## old 0.16 0.44

## new 0.28 0.12

第2引数  
省略

第2引数  
1:列

proportions(mx1, 1) # 行の割合

## plus minus

## old 0.2666667 0.7333333

## new 0.7000000 0.3000000

proportions(mx1, 2) # 列の割合

## plus minus

## old 0.3636364 0.7857143

## new 0.6363636 0.2142857

第2引数  
2:行



# 2 × 2 分割表

## ● 表示3.3.1 2つの割合のデータ

### 表示3.3.4 JMP[二変量の関係]による解析結果

スクリプトファイル：Green3-3-3a.R

#### 方法

マトリックスの一部を表す場合は、  
オブジェクト名 [行の番号、列の番号] とする

mx1の列の部分を c(2, 1) としてmx2 に付値すると  
mx1の2行目が mx2の1行目、  
mx1の1行目が mx2の2行目になる

行の場合も同様にして入れ替えできる

	有効	無効	合計
旧処方	4	11	15
新処方	7	3	10

```
mx1
##      plus minus
## old   4    11
## new   7     3

# 列の入れ替え
mx2 <- mx1[, c(2, 1)]
mx2
##      minus plus
## old   11    4
## new    3    7

# 行の入れ替え
mx3 <- mx1[c(2, 1), ]
mx3
##      plus minus
## new   7     3
## old   4    11
```

列番号

行番号

# 2 × 2 分割表：グラフ化

## ●表示3.3.1 2つの割合のデータ

### 表示3.3.4 JMP[二変量の関係]による解析結果

スクリプトファイル：Green3-3-3a.R

利用した関数

barplot, t

方法

マトリックスの位置とグラフの関係

barplot 関数での legend 引数

t 関数でマトリックスを転置 (行と列を入替)

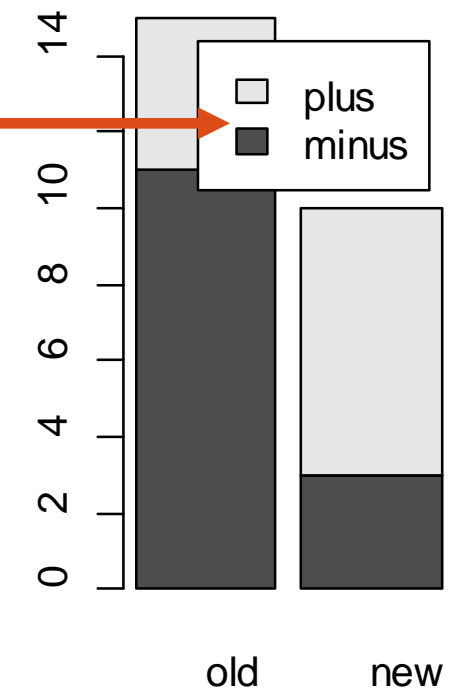
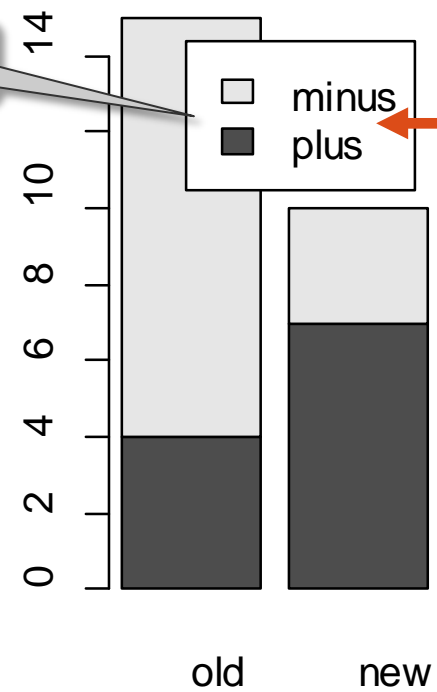
転置

凡例の有無

```
barplot(t(mx1), legend = TRUE)  
barplot(t(mx2))
```

mx1			mx2		
##	plus	minus	##	minus	plus
## old	4	11	## old	11	4
## new	7	3	## new	3	7

凡例





# 2 × 2 分割表：グラフ化

## ●表示3.3.1 2つの割合のデータ

### 表示3.3.4 JMP[二変量の関係]による解析結果

スクリプトファイル：Green3-3-3a.R

利用した関数

barplot, t

方法

マトリックスの位置とグラフの関係

barplot 関数での beside 引数

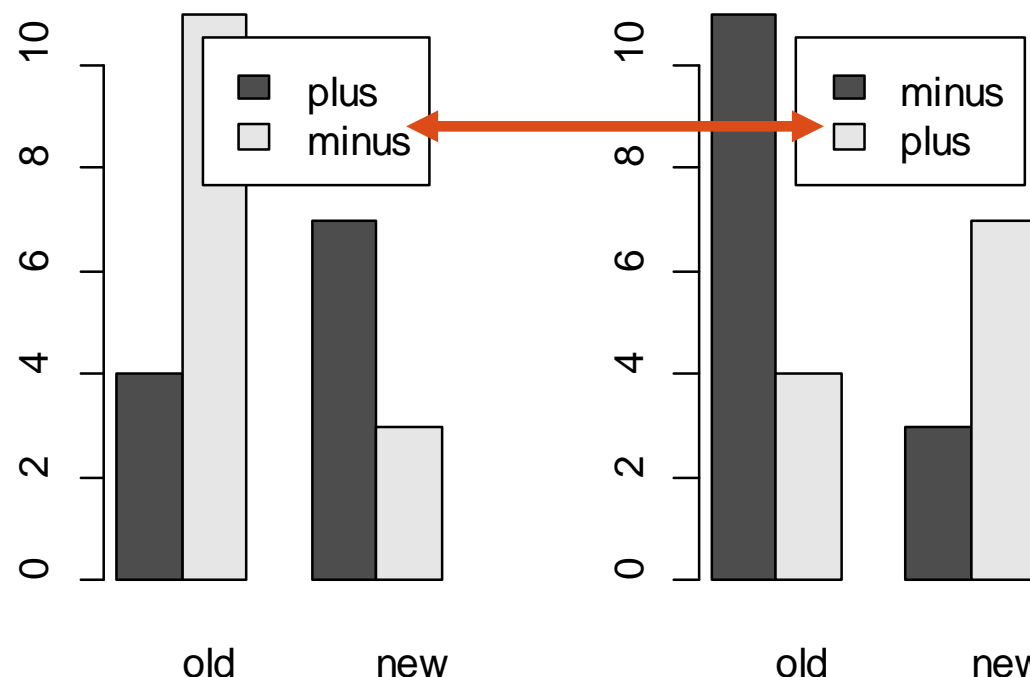
t 関数でマトリックスを転置 (行と列を)

転置

凡例の有無

```
barplot(t(mx1), beside = TRUE, legend = TRUE)
barplot(t(mx2), beside = TRUE)
```

mx1			mx2		
##		plus minus	##		minus plus
## old	4	11	## old	11	4
## new	7	3	## new	3	7



# 2 × 2 分割表：グラフ化

## ● 表示3.3.1 2つの割合のデータ

### 表示3.3.4 JMP[二変量の関係]による解析結果

スクリプトファイル：Green3-3-3a.R

利用した関数

`mosaicplot, t`

方法

マトリックスの位置とグラフの関係

```
mosaicplot(mx1, color = TRUE, cex.axis = 1)  
mosaicplot(mx2, color = TRUE, cex.axis = 1)
```

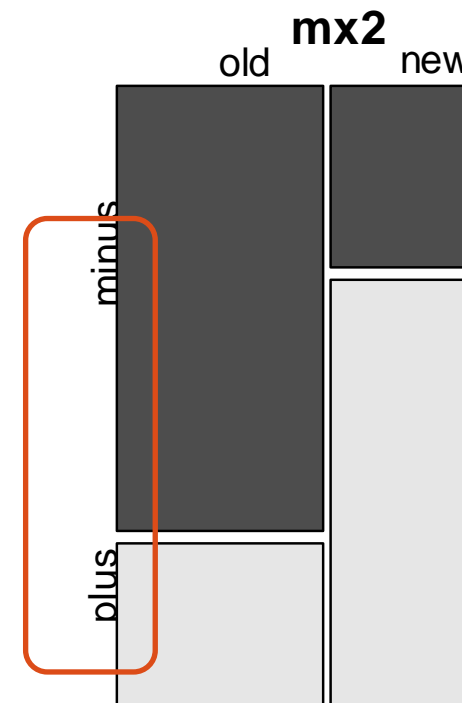
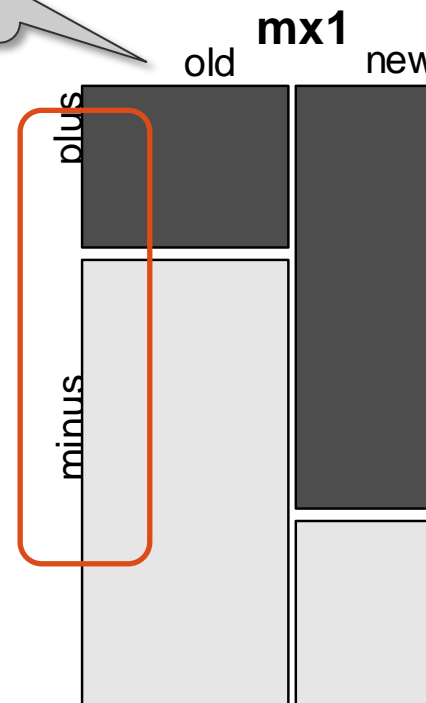
着色の有無

軸目盛の  
フォントの大きさ

mx1	plus	minus
## old	4	11
## new	7	3

mx2	minus	plus
## old	11	4
## new	3	7

軸目盛の  
フォントの大きさ



# 割合、オッズ、対数オッズ（ロジット）

- 表示3.3.2 pとオッズあるいはその対数との関係

スクリプトファイル：Green3-3-3b.R

利用した関数

plot、abline、seq

方法

ベクトル演算で 50 個の数値を

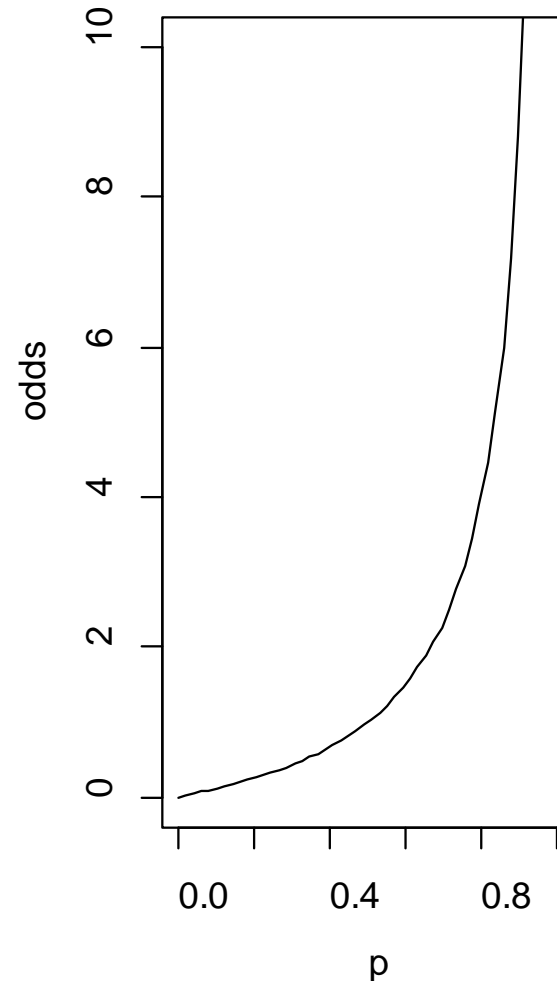
1 行のスクリプトで計算

0~1 から  
等間隔で  
50個生成

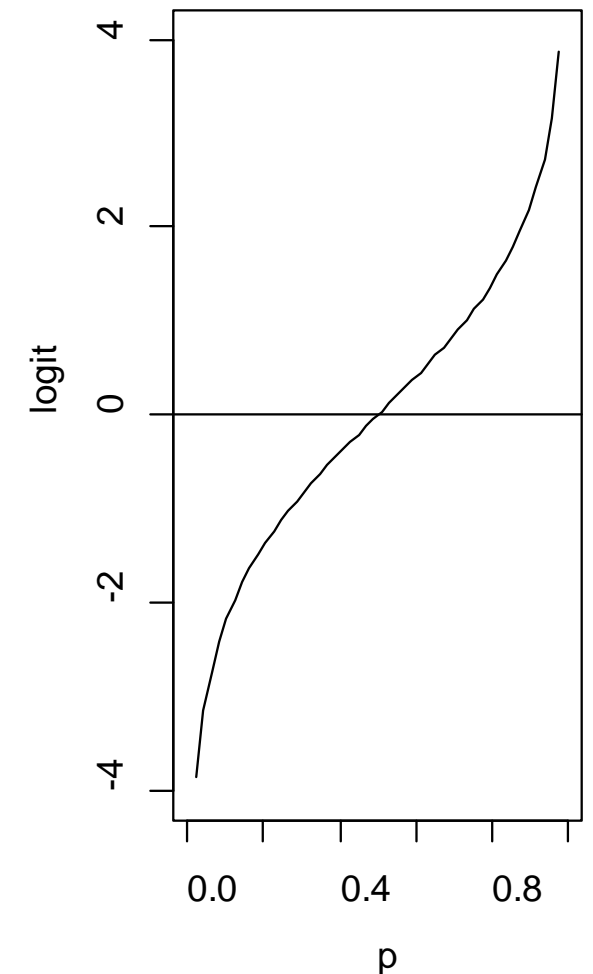
```
p <- seq(0, 1, length = 50) # 割合  
odds <- p / (1 - p) # オッズ  
logit <- log(odds) # ロジット
```

ベクトル演算

割合 p とオッズの関係



割合 p とロジットの関係



# 割合の差、割合の比、オッズ比

- 表示3.3.3 割合の差、割合の比とオッズ比の等高線

スクリプトファイル

Green3-3-3b.R

利用した関数

function、plot、curve、abline、text

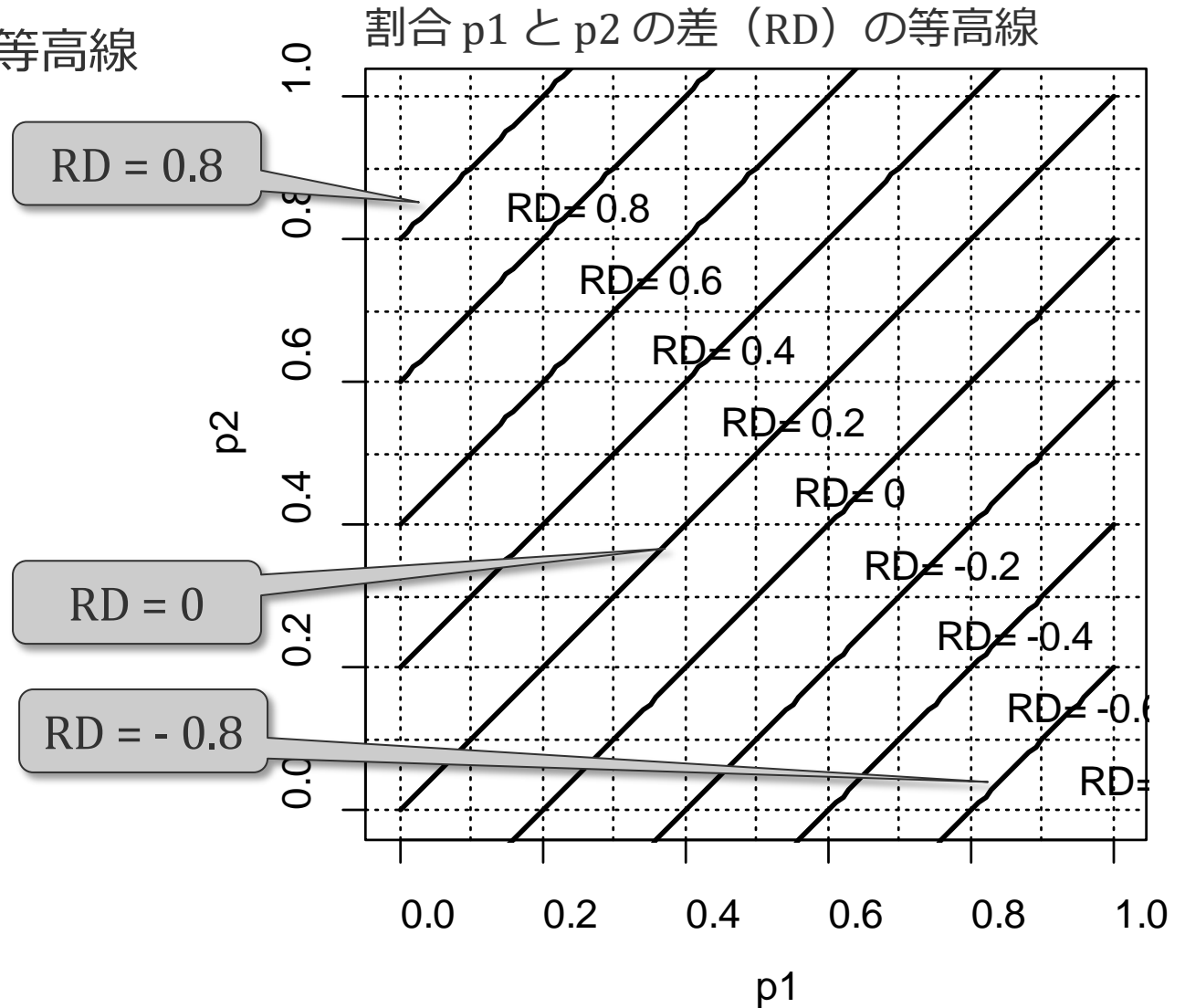
方法

割合の差 (RD) の等高線を表す式を

function 関数で定義

curve 関数で等高線を描画

$$RD = p2 - p1$$





# 割合の差、割合の比、オッズ比

- 表示3.3.3 割合の差、割合の比とオッズ比の等高線

スクリプトファイル

Green3-3-3b.R

利用した関数

function、plot、curve、abline、text

方法

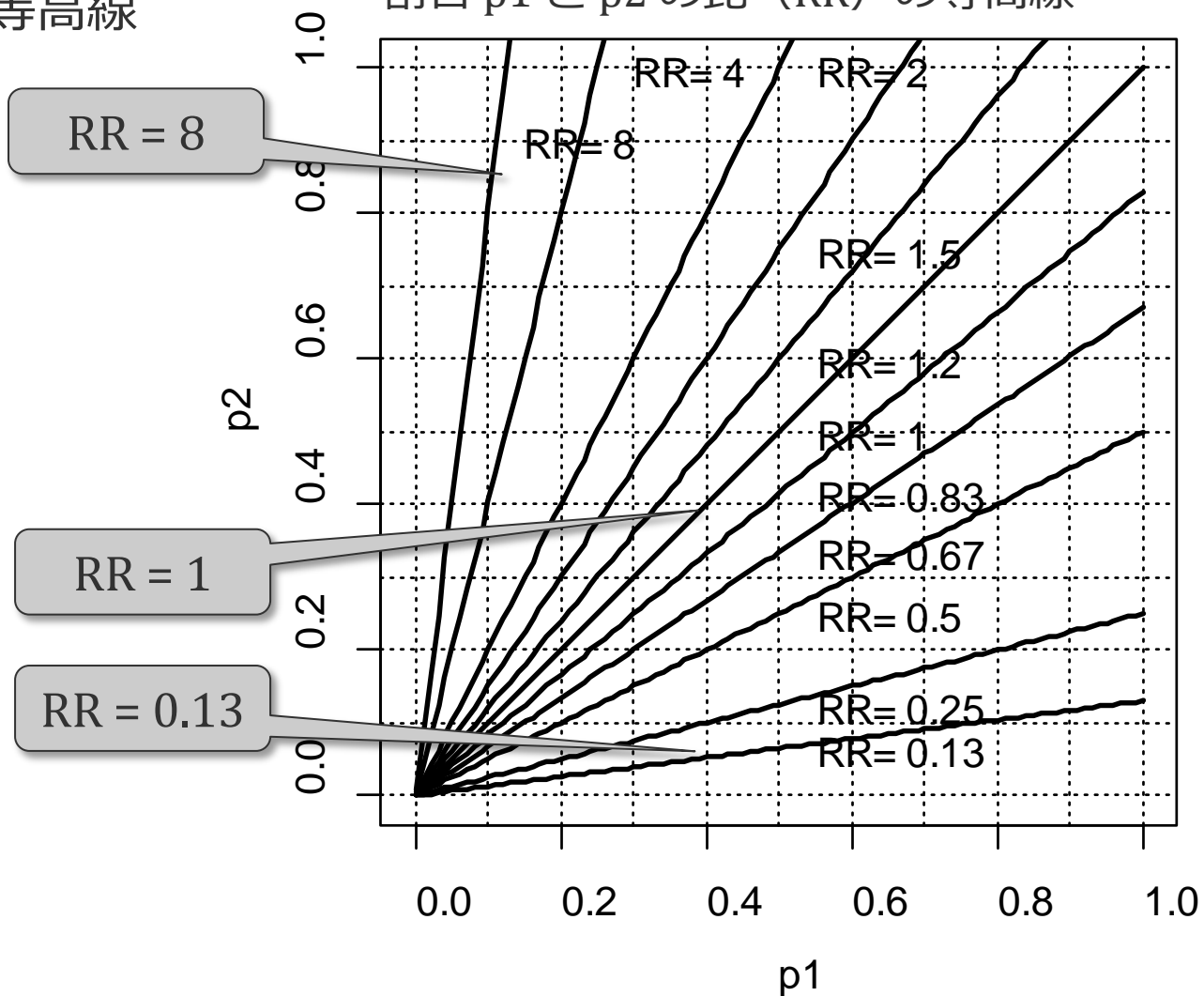
割合の比 (RR) の等高線を表す式を

function 関数で定義

curve 関数で等高線を描画

$$RR = p2 / p1$$

割合 p1 と p2 の比 (RR) の等高線



# 割合の差、割合の比、オッズ比

- 表示3.3.3 割合の差、割合の比とオッズ比の等高線

スクリプトファイル

Green3-3-3b.R

利用した関数

function、plot、curve、abline、text

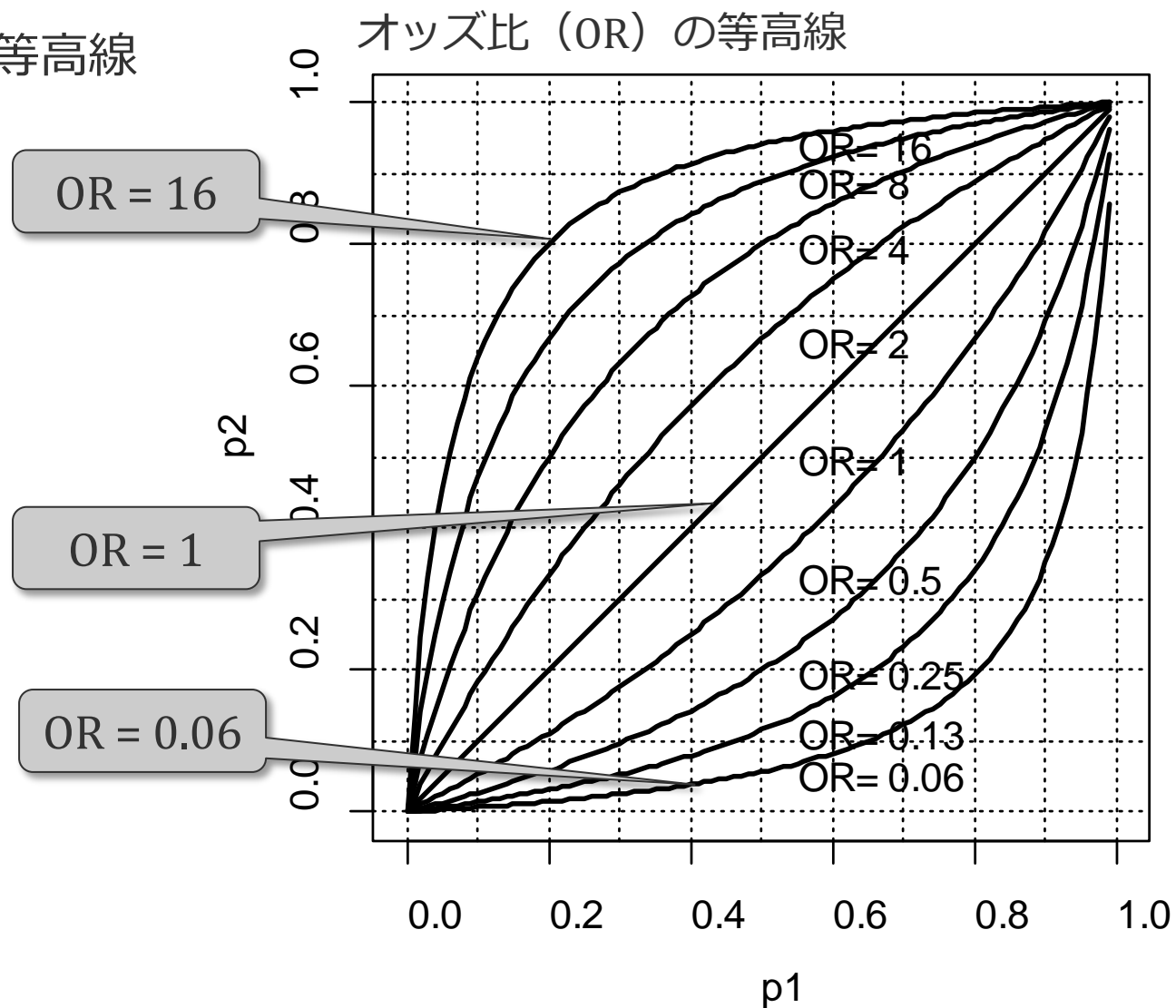
方法

オッズ比 (OR) の等高線を表す式を

function 関数で定義

curve 関数で等高線を描画

$$OR = p2(1 - p1) / p1(1 - p2)$$





# 2 × 2 分割表

割合の差の検定と区間推定  
対応がある二値データ（補足）



# 2 × 2 分割表：割合の差の検定と区間推定

- 表示3.3.5 JMP[二変量の関係]による解析結果

スクリプトファイル

Green3-3-3a.R

利用した関数

vcd::aassocstats

方法

尤度比検定（G検定）、Pearsonのカイ2乗検定

```
mx1
##      plus minus
## old    4     11
## new    7      3
```

尤度比検定（G検定）  
Pearson のカイ2乗検定

ファイ係数  
コンテンジエンシイ係数（分割係数）  
クラメールのV

```
assocstats(mx1)
##              X^2 df P(> X^2)
## Likelihood Ratio 4.6817  1 0.030485
## Pearson          4.5725  1 0.032489
##
## Phi-Coefficient   : 0.428
## Contingency Coeff.: 0.393
## Cramer's V       : 0.428
```



# 2 × 2 分割表：割合の差の検定と区間推定

p.170

- 表示3.3.5 JMP[二変量の関係]による解析結果

表示3.3.6 Excel による尤度比検定

スクリプトファイル

Green3-3-3a.R

利用した関数

DescTools::GTest

方法

尤度比検定 (G検定)

```
mx1
##      plus minus
## old    4     11
## new    7      3
```

```
GTest(mx1, correct = "none")
##
## Log likelihood ratio (G-test) test of
## independence without correction
##
## data:  mx1
## G = 4.6817, X-squared df = 1, p-value = 0.03048
```



# 2 × 2 分割表：割合の差の検定と区間推定

p.170

- 表示3.3.5 JMP[二変量の関係]による解析結果

表示3.3.7 Pearson と 2 × 2 分割表

スクリプトファイル

Green3-3-3a.R

利用した関数

chisq.test、round

方法

Pearson のカイ 2 乗検定

期待度数

相対的外れ

```
chi_out <- chisq.test(mx1, correct = FALSE)
chi_out
## Pearson's Chi-squared test
## data: mx1
## X-squared = 4.5725, df = 1, p-value = 0.03249
chi_out$expected
##      plus minus
## old  6.6    8.4
## new  4.4    5.6
round(chi_out$residuals, digits = 3)
##      plus  minus
## old -1.012  0.897
## new  1.240 -1.099
```

小数点以下 3 桁  
に四捨五入

# 2 × 2 分割表：割合の差の検定と区間推定

p.170

- 表示3.3.5 JMP[二変量の関係]による解析結果

表示3.3.8 Fisher の正確検定

スクリプトファイル

Green3-3-3a.R

利用した関数

fisher.test、exact2x2::fisher.exact

方法

Fisher の正確検定

条件付きの最尤推定量  
(conditional MLE)  
によるオッズ比と区間推定

両側検定

```
fisher.test(mx1,  
            alternative = "two.sided",  
            conf.int = TRUE,  
            conf.level = 0.95 )
```

```
##  
## Fisher's Exact Test for Count Data  
##  
## data: mx1  
## p-value = 0.04862  
## alternative hypothesis: true odds  
## ratio is not equal to 1  
## 95 percent confidence interval:  
## 0.01811618 1.19549228  
## sample estimates:  
## odds ratio  
## 0.1698609
```

# 2 × 2 分割表：割合の差の検定と区間推定

p.177

- 表示3.3.9 JMP[二変量の関係]による解析結果
- 表示3.3.10 オッズとオッズ比の信頼区間

スクリプトファイル

Green3-3-3a.R

利用した関数：Epi::twoby2

方法

割合の差、割合の比、  
オッズ比とその区間推定

1 行目の水準 / 2 行目の水準  
(new / old)

1 行目の水準 - 2 行目の水準  
(new - old)

```
twoby2(mx3, alpha = 0.05)
## 2 by 2 table analysis:
## -----
## Outcome      : plus
## Comparing    : new vs. old
##
##           plus minus      P(plus) 95% conf. interval
## new        7      3      0.7000  0.3763  0.9002
## old        4     11      0.2667  0.1038  0.5331
##
##                                     95% conf. interval
##           Relative Risk: 2.6250      1.0335  6.6674
##           Sample Odds Ratio: 6.4167  1.0911 37.7347
##           Conditional MLE Odds Ratio: 5.8872 0.8365 55.1993
##           Probability difference: 0.4333  0.0385  0.6820
##
##           Exact P-value: 0.0486
##           Asymptotic P-value: 0.0397
```

mx3

	plus	minus
new	7	3
old	4	11

fisher.testの結果と一致





# 2 × 2 分割表：割合の差の検定と区間推定

- 表示3.3.9 JMP[二変量の関係]による解析結果

表示3.3.10 オッズとオッズ比の信頼区間

スクリプトファイル

Green3-3-3a.R

利用した関数

DestTools::OddsRatio

方法

オッズ比の区間推定

計算方法を指定  
デフォルトは"wald"

```
OddsRatio(mx2, method = "mle", conf.level = 0.95)
## odds ratio      lwr.ci      upr.ci
## 5.8871691      0.8364755 55.1992700

OddsRatio(mx2, method = "midp", conf.level = 0.95)
## odds ratio      lwr.ci      upr.ci
## 5.755319        1.026918 41.533631

OddsRatio(mx2, method = "wald", conf.level = 0.95)
## odds ratio      lwr.ci      upr.ci
## 6.416667        1.091134 37.734696
```



# 2 × 2 分割表：割合の差の検定と区間推定

p.177

## ● 表示3.3.9 JMP[二変量の関係]による解析結果

表示3.3.11 割合の差の区間推定

スクリプトファイル

Green3-3-3a.R

利用した関数

prop.test

方法

割合の差の区間推定

Wald法（正規近似）による

（表示3.3.11 上段）

```
prop.test(mx2, alternative = "two.sided",
          conf.level = 0.95, correct = FALSE)
## Warning in prop.test(mx2, alternative =
## "two.sided", conf.level = 0.95, :
## Chi-squared approximation may be incorrect
##
## 2-sample test for equality of proportions
## without continuity correction
##
## X-squared = 4.5725, df = 1, p-value = 0.03249
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.07173704 0.79492962
## sample estimates:
##      prop 1      prop 2
## 0.7333333 0.3000000
```

# 2 × 2 分割表：割合の差の検定と区間推定

p.177

## ● 表示3.3.9 JMP[二変量の関係]による解析結果

### 表示3.3.11 割合の差の区間推定

スクリプトファイル

Green3-3-3a.R

利用した関数

prop.test

方法

割合の差の区間推定

各々の成分に 1 を加えてから、

prop.test を実行

Agresti-Caffo による方法

(JMP が用いている手法、

表示3.3.11 下段)

```
mx3 <- mx2 + 1
prop.test(mx3, alternative = "two.sided",
          conf.level = 0.95, correct = FALSE)
##
## 2-sample test for equality of proportions
## without continuity correction
##
## data:  mx3
## X-squared = 3.9476, df = 1, p-value =
## 0.04694
## alternative hypothesis: two.sided
## 95 percent confidence interval:
## 0.02896216 0.71613588
```

各々の成分に  
1 を加える



# 2 × 2 分割表：割合の差の検定と区間推定

## ● 表示3.3.9 JMP[二変量の関係]による解析結果

### 表示3.3.11 割合の差の区間推定

スクリプトファイル

Green3-3-3a.R

利用した関数

DestTools::BinomdiffCI

方法

割合の差の区間推定

Wald 法、Agresti-Caffo 法、  
Score 法など多くの手法の  
区間推定を表示

```
BinomDiffCI(x1 = 7, n1 = 10, x2 = 4, n2 = 15,  
            conf.level = 0.95,  
            sides = "two.sided", method = "wald")
```

```
## 方法を自動的に選択して表示  
##  
##          lwr.ci upr.ci  
## 1. Wald, no CC      0.0717 0.7949  
## 2. Wald, CC        -0.0116 0.8783  
## 3. Haldane          0.0336 0.7153  
## 4. Jeffreys-Perks  0.0337 0.7155  
## 5. Mee              0.0359 0.7128  
## 6. Miettinen-Nurminen 0.0277 0.7169  
## 10. Score, no CC    0.0385 0.6820  
## 11. Score, CC      -0.0152 0.7153  
## 12. Hauck-Andersen 0.0048 0.8619  
## 13. Agresti-Caffo  0.0290 0.7161  
## 16. Brown-Li       0.0330 0.7681
```

方法を指定

# 2 × 2 分割表：対応がある二値データ（補足）

## ● McNemar 検定

スクリプトファイル：Green3-3-3e.R

利用した関数：matrix、

dimnames、addmargins

proportions、round

### 事例

655人に薬を投与し、

投与前(b)と投与後(a)に

ある反応の有無 (yes/no) を調査

```
addmargins(mx)
```

第2引数  
省略

```
##          after
## before  a_yes a_no Sum
##  b_yes   93  147 240
##  b_no  100  315 415
##  Sum    193  462 655
```

縦計、横計  
共に意味がある

```
mx <- matrix(c( 93, 147, (藤井,2010)
               100, 315),
             nrow = 2, ncol = 2, byrow = TRUE)

dimnames(mx) <- list(before = c("b_yes", "b_no"),
                    after = c("a_yes", "a_no"))

mx

##          after
## before  a_yes a_no
##  b_yes   93  147
##  b_no  100  315
```

beforeとafterに  
対応がある  
(独立ではない)

655の内訳

```
round(proportions(mx),
      digits = 3)
```

```
##          after
## before  a_yes a_no
##  b_yes 0.142 0.224
##  b_no 0.153 0.481
```

655人に  
対する割合  
合計は1

## 2 × 2 分割表：対応がある二値データ（補足）

### ● McNemar 検定

スクリプトファイル

Green3-3-3e.R

利用した関数

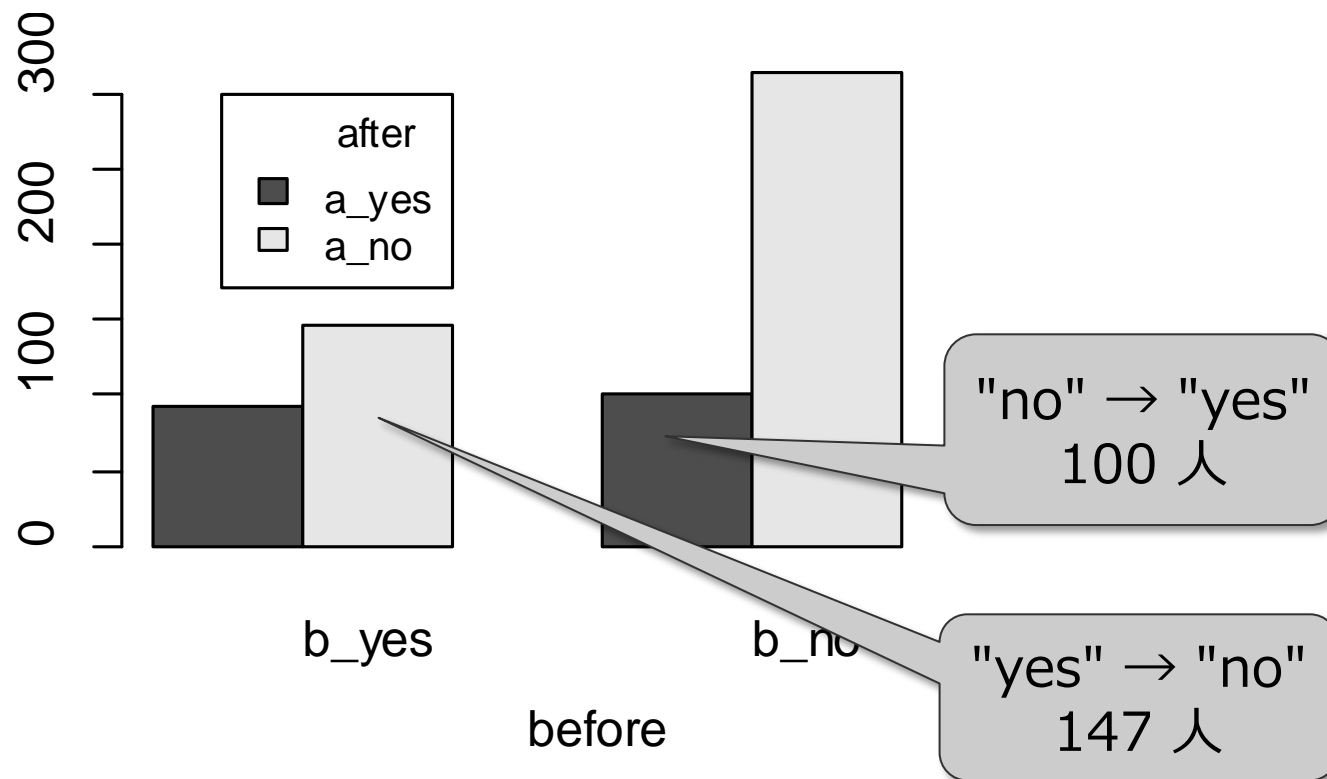
barplot

方法

655人に薬を投与し、  
投与前(b)と投与後(a)に  
反応の有無 (yes/no) を調査

投薬の効果がなければ、  
"no" → "yes"の人数と  
"yes" → "no"の人数は同数になる  
ことが期待される → 2項検定

```
barplot(t(mx), beside = TRUE, legend = TRUE,  
        ylim = c(0, 300), xlab = "before",  
        args.legend = list(x = 3, y = 300, cex = 0.8,  
                            title = "after"))
```



## 2 × 2 分割表：対応がある二値データ（補足）

- McNemar 検定

スクリプトファイル

Green3-3-3e.R

利用した関数

binom.test

方法

2 項検定

帰無仮説 ( $\pi = 0.5$ ) の下で、

247人 (100 + 147) が 100人と 147人以上に偏る確率を計算 (両側検定)

```
mx
##          after
## before  a_yes a_no
##  b_yes   93  147
##  b_no  100  315
```

```
pbinom(q = 100, size = 247, prob = 0.5, lower.tail = TRUE) * 2
## [1] 0.003341124
```

両側検定



## 2 × 2 分割表：対応がある二値データ（補足）

- McNemar 検定

スクリプトファイル：Green3-3-3e.R

利用した関数：mcnemar.test、exact2x2::macnemar.exact

方法：McNemar 検定

```
mx
##           after
## before  a_yes a_no
##  b_yes    93  147
##  b_no   100  315
```

```
mcnemar.exact(mx, conf.level = 0.95)
##
## Exact McNemar test (with central confidence intervals)
##
## data:  mx
## b = 147, c = 100, p-value = 0.003341
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  1.132481 1.914678
## sample estimates:
## odds ratio
##          1.47
```





# 2 × 2 分割表

glm 関数の使い方  
名義ロジスティックのあてはめ (2つの割合)



# glm 関数の使い方 (補足)

- glm による名義ロジスティック回帰の入力 (スクリプトファイル: Green3-3-3g.R)

事例: 旧処方(old) 15例、新処方(new) 10例で、結果が有効 (yes) か無効 (no) かを観察した  
この二値データを順番に加工、いずれのオブジェクトでも glm 関数により解析が可能

(1) df1

group	event
new	yes
old	no
new	no
new	yes
...	...
old	no
old	no
old	no

(1) 二値の生データ (度数なし、ランダム化された実験)

(2) 生データを集計 (度数あり)

(3) 分割表

(4) 分割表から yes の割合を算出 (Sum : yes と no の合計、p : yesの割合)

(2) df2

group	event	Freq
old	no	11
new	no	3
old	yes	4
new	yes	7

(3) tb1

group	event	
	yes	no
a_old	4	11
b_new	7	3

(4) df3

yes	no	Sum	p	group
4	11	15	0.267	a_old
7	3	10	0.700	b_new

# glm 関数の使い方 (補足)

- glm による名義ロジスティック回帰の入力 (スクリプトファイル: Green3-3-3g.R)

(1) 二値の生データ (度数なし)

```
df <- read_excel("Green3-3.xlsx", sheet = "33-wariai")
df$group <- factor(df$group, levels = c("old", "new"))
df$event <- factor(df$event, levels = c("no", "yew"))
df1 <- df
```

(1) df1

group	event
new	yes
old	no
new	no
new	yes
...	...
old	no
old	no
old	no

(2) df2

group	event	Freq
old	no	11
new	no	3
old	yes	4
new	yes	7

自動的に命名

group	event	yes	no
a_old		4	11
b_new		7	3

(4) df3

yes	no	Sum	p	group
4	11	15	0.267	a_old
7	3	10	0.700	b_new

位置決めは重要  
先頭が基準



# glm 関数の使い方 (補足)

- glm による名義ロジスティック回帰の入力 (スクリプトファイル: Green3-3-3g.R)
  - (2) 生データを集計 (度数あり)

```
df2 <- data.frame(table(group = df1$group, event = df1$event))
```

(1) df1

group	event
new	yes
old	no
new	no
new	yes
...	...
old	no
old	no
old	no



(2) df2

group	event	Freq
old	no	11
new	no	3
old	yes	4
new	yes	7

自動的に命名



(3) tb1

group	event	
	yes	no
a_old	4	11
b_new	7	3



(4) df3

yes	no	Sum	p	group
4	11	15	0.267	a_old
7	3	10	0.700	b_new

# glm 関数の使い方 (補足)

- glm による名義ロジスティック回帰の入力 (スクリプトファイルコロンGreen3-3-3g.R)

## (3) 分割表

```
xtabs(~ group + event, data = df3)  
tb1 のデータ構造はテーブル §3.4参照
```

```
tb1 <- xtabs(Freq ~ group + event, data = df2)
```

```
tb1 <- tb1[, c("yes", "no")]
```

この設定で yesの割合になる

```
rownames(tb1) <- c("a_old", "b_new")
```

基準の行名"old"の頭文字を  
順番の早い文字に変更

### (1) df1

group	event
new	yes
old	no
new	no
new	yes
...	...
old	no
old	no
old	no

### (2) df2

group	event	Freq
old	no	11
new	no	3
old	yes	4
new	yes	7

### (3) tb1

group	event	
	yes	no
a_old	4	11
b_new	7	3

列名

行名

### (4) df3

yes	no	Sum	p	group
4	11	15	0.267	a_old
7	3	10	0.700	b_new

自動的に命名

# glm 関数の使い方 (補足)

- glm による名義ロジスティック回帰の入力 (スクリプトファイル: Green3-3-3g.R)
  - (4) 分割表から yes の割合を算出

```
tb2 <- addmargins(tb1, 2) # Sum 列を生成
mx <- cbind(tb2, p = tb2[, "yes"] / tb2[, "Sum"])
df3 <- data.frame(mx, group = c("a_old", "b_new"))
```

(1) df1

group	event
new	yes
old	no
new	no
new	yes
...	...
old	no
old	no
old	no

(2) df2

group	event	Freq
old	no	11
new	no	3
old	yes	4
new	yes	7

(3) tb1

group	event	
	yes	no
a_old	4	11
b_new	7	3

基準の行名 "old" の頭文字を  
順番の早い文字に変更

(4) df3

yes	no	Sum	p	group
4	11	15	0.267	a_old
7	3	10	0.700	b_new

# glm 関数の使い方 (補足)

- glm による名義ロジスティック回帰の入力 (スクリプトファイル: Green3-3-3g.R)  
glm 関数を実行すると、パラメータ推定値、効果の尤度比検定は同じ結果が得られる

```
(1) glm(event ~ group, data = df1, family = binomial(link = "logit"))
```

```
(2) glm(event ~ group, weights = Freq, data = df2, family = ...)
```

```
(3) glm(tb1 ~ c("a_old", "b_new"), family = ...)
```

```
(4) glm(p ~ group, weights = Sum, data = df3, family = ...)
```

(1) df1

group	event
new	yes
old	no
new	no
new	yes
...	...
old	no
old	no
old	no

(2) df2

group	event	Freq
old	no	11
new	no	3
old	yes	4
new	yes	7

(3) tb1

group	event	
	yes	no
a_old	4	11
b_new	7	3

列名

行名

(4) df3

yes	no	Sum	p	group
4	11	15	0.267	a_old
7	3	10	0.700	b_new

# 名義ロジスティックのあてはめ（2つの割合）

## ●表示3.3.12 JMP[モデルのあてはめ]による解析結果

スクリプトファイル

Green3-3-3c.R

利用した関数

factor

方法

因子型の設定

glm 関数に渡すデータの

カテゴリーの位置の設定

group の基準は"old"

ダミー変数 1 の"0"

(contr.treatment の設定)

event の基準は"no"

noに対する yes のオッズ

```
df <- read_excel("Green3-3.xlsx", sheet = "33-wariai")
df <- data.frame(df)
df$group <- factor(df$group, levels = c("old", "new"))
df$event <- factor(df$event, levels = c("no", "yes"))
```

文字型→因子型

カテゴリーの位置を指定  
先頭に基準となる  
カテゴリーを配置

d : ダミー変数 1  
dd : ダミー変数 2  
(第2部 §2.3参照)

```
df
##      group event d dd
## 1     new   yes  1  1
## 2    old    no   0 -1
## 3     new    no  1  1
## 4     new   yes  1  1
## 5     new   yes  1  1
## 6     old    no  0 -1
.....
## 24    old    no  0 -1
## 25    old    no  0 -1
```



# 名義ロジスティックのあてはめ（2つの割合）

- 表示3.3.12 JMP[モデルのあてはめ]による解析結果、表示3.3.10 オッズとオッズの比の信頼区間  
スクリプトファイル

Green3-3-3c.R

利用した関数

glm、summary

方法

説明変数に質的変数 group

を利用

この解析結果は正しい

しかし、パラメータ推定値は  
表示3.3.12（JMPの結果）と  
一致しない

質的変数

```
glm_out <- glm(event ~ group, , data = df,  
               family = binomial(link = "logit"))  
  
summary(glm_out)  
## Coefficients:  
##           Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -1.0116    0.5839  -1.733  0.0832 .  
## groupnew    1.8589    0.9039   2.056  0.0397 *
```

表示3.3.12  
切片  
-0.082

表示3.3.12  
処方[新処方]  
0.929



# 名義ロジスティックのあてはめ（2つの割合）

- 表示3.3.12 JMP[モデルのあてはめ]による解析結果、表示3.3.10 オッズとオッズの比の信頼区間

スクリプトファイル

Green3-3-3c.R

利用した関数

glm、summary

方法

説明変数にダミー変数1「d」  
  を利用

パラメータ推定値は、  
group の contrasts をデフォルトの  
contr.treatment で使った結果と一致  
→ 質的変数をダミー変数1で  
  処理している

ダミー変数1

```

glm_out1 <- glm(event ~ d, , data = df,
               family = binomial(link = "logit"))

summary(glm_out1)

## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.0116    0.5839  -1.733   0.0832 .
## d            1.8589    0.9039   2.056   0.0397 *

```

表示3.3.12  
切片  
-0.082

表示3.3.12  
処方[新処方]  
0.929



# 名義ロジスティックのあてはめ（2つの割合）

- 表示3.3.12 JMP[モデルのあてはめ]による解析結果、表示3.3.10 オッズとオッズの比の信頼区間

スクリプトファイル

Green3-3-3c.R

利用した関数

glm、summary

方法

質的変数の contrasts を

contr.sum に指定すると

パラメータ推定値は

表示3.3.12 の JMP と一致



質的変数はダミー変数 2 で処理される

```
df$group <- factor(df$group, levels = c("new", "old"))

glm_out3 <- glm(event ~ group, data = df,
                family = binomial(link = "logit"),
                contrasts = list(group = "contr.sum"))

summary(glm_out3)

# Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.08215    0.45197  -0.182   0.8558
## group1      0.92945    0.45197  -2.056   0.0397 *
```

contrastsの設定

表示3.3.12  
切片  
-0.082

表示3.3.12  
処方[新処方]  
0.929

# 名義ロジスティックのあてはめ（2つの割合）

p.180

- 表示3.3.12 JMP[モデルのあてはめ]による解析結果、表示3.3.10 オッズとオッズの比の信頼区間

スクリプトファイル

Green3-3-3c.R

利用した関数

glm、summary

方法

質的変数の contrasts を

contr.sum に指定すると

パラメータ推定値は

表示3.3.12 の JMP と一致

↓

質的変数はダミー変数 2 で処理される

ダミー変数 2 の基準「- 1」は、levels 引数で末尾に置いた水準になる

ここでは、「JMP」の出力と合わせるために、「old」を末尾に変更している

```
df$group <- factor(df$group, levels = c("new", "old"))
glm_out3 <- glm(event ~ group, data = df,
                family = binomial(link = "logit"),
                contrasts = list(group = "contr.sum"))
summary(glm_out3)
# Coefficients:
##          Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.08215    0.45197  -0.182   0.8558
## group1      0.92945    0.45197  -2.056   0.0397 *
```

末尾が基準

contrastsの設定

# 名義ロジスティックのあてはめ（2つの割合）

● 表示3.3.12 JMP[モデルのあてはめ]による解析結果

表示3.3.10 オッズとオッズ比の信頼区間

単位オッズ比：1単位だけ変化した場合

範囲オッズ比：全体で変化した場合

(§4.1 参照)

	効果		計	p	対数オッズ (ロジット)
	有効	無効			
旧処方	4	11	15	0.267	-1.012
新処方	7	3	10	0.700	0.847

old  
new

半分

差 1.859

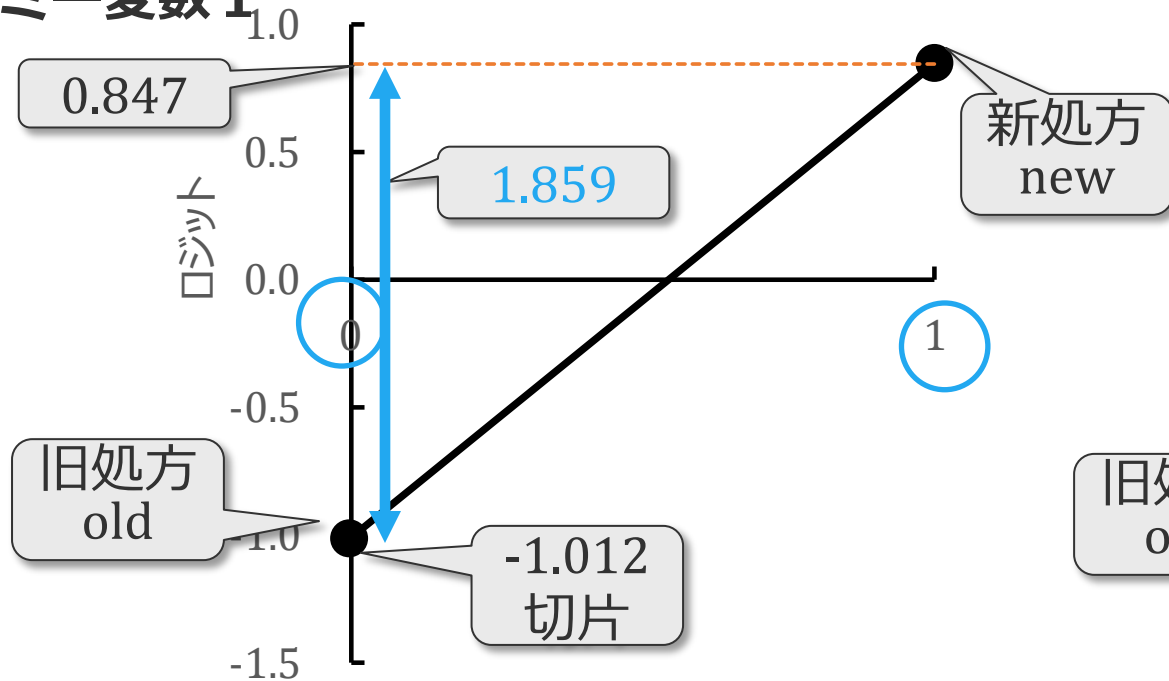
0.929

exp(差) : オッズ比

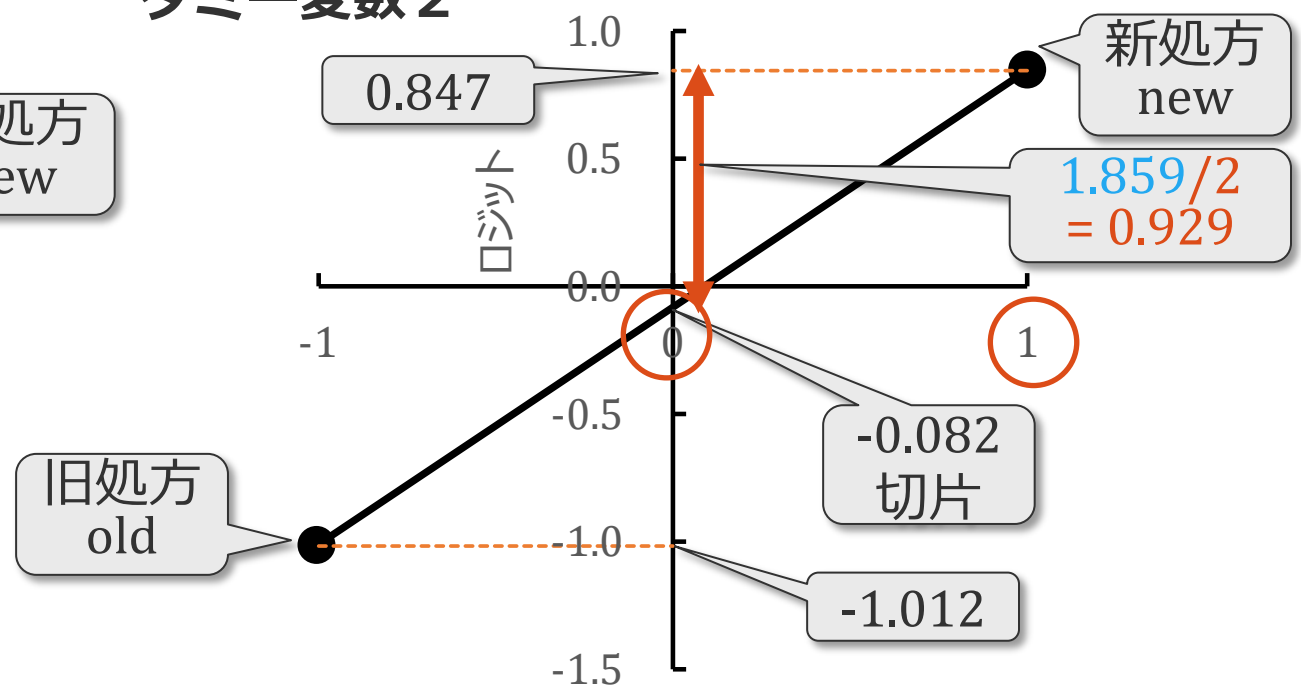
6.417

2.533

ダミー変数 1



ダミー変数 2



# 名義ロジスティックのあてはめ（2つの割合）

- 表示3.3.12 JMP[モデルのあてはめ]による解析結果  
表示3.3.10 オッズとオッズ比の信頼区間

**単位オッズ比**：1単位だけ変化した場合

**範囲オッズ比**：全体で変化した場合

(§4.1 参照)

	効果		計	p	対数オッズ (ロジット)
	有効	無効			
旧処方 old	4	11	15	0.267	-1.012
新処方 new	7	3	10	0.700	0.847

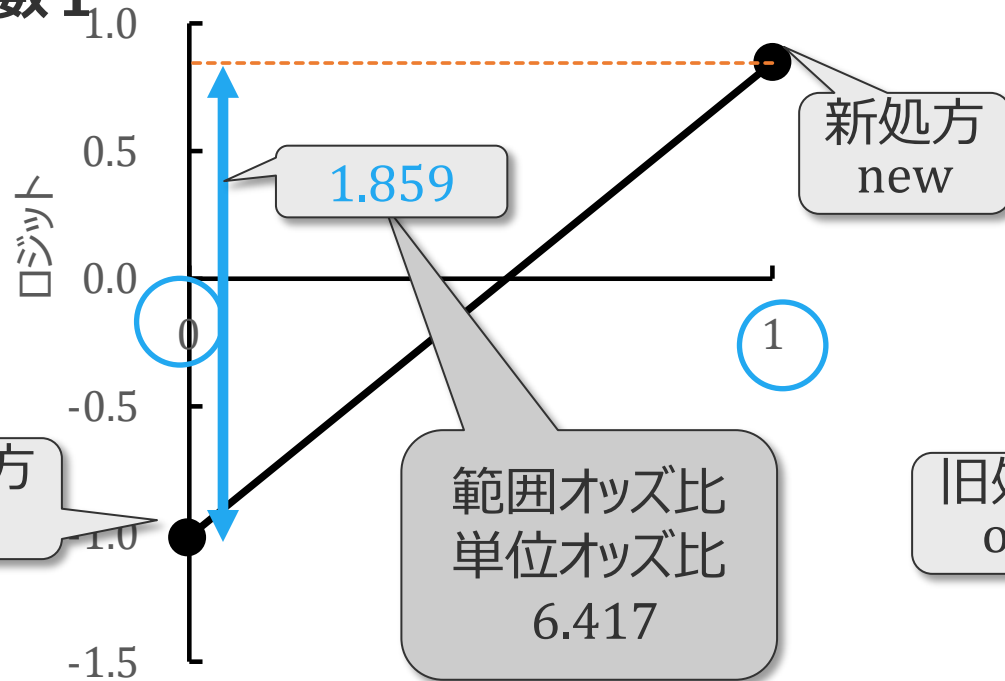
old  
new

半分

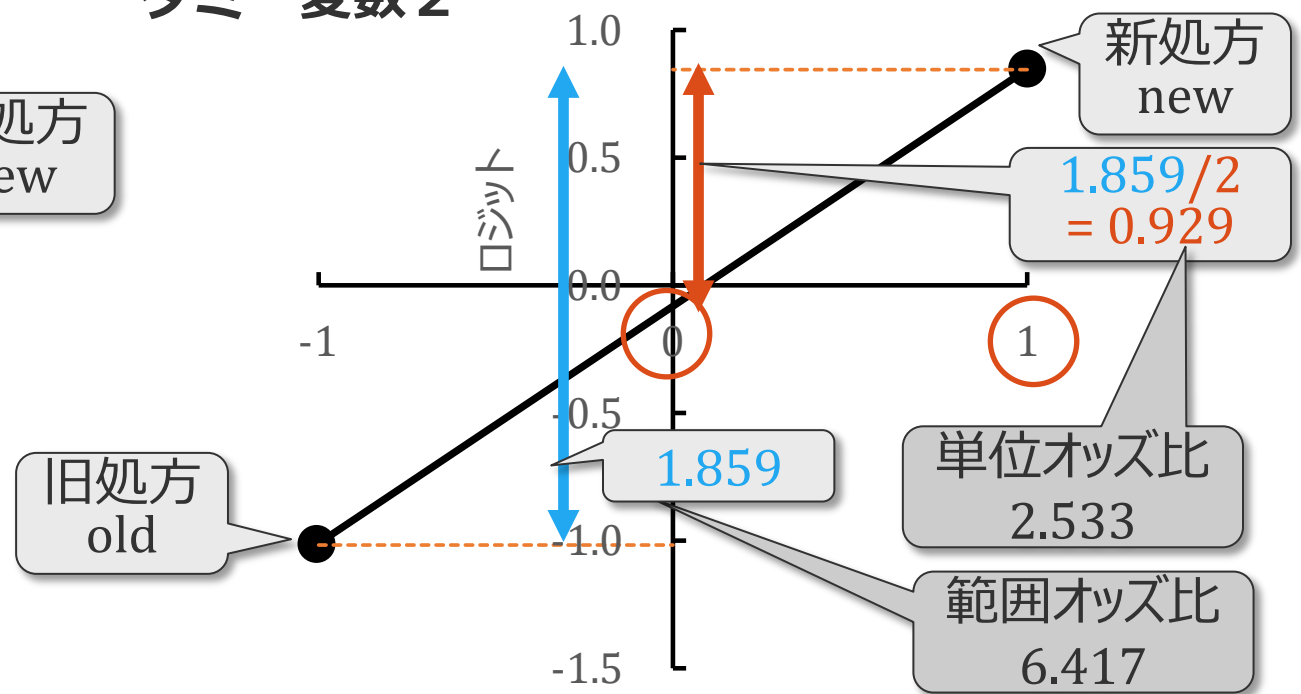
差  
1.859  
6.417

0.929  
2.533

ダミー変数 1



ダミー変数 2



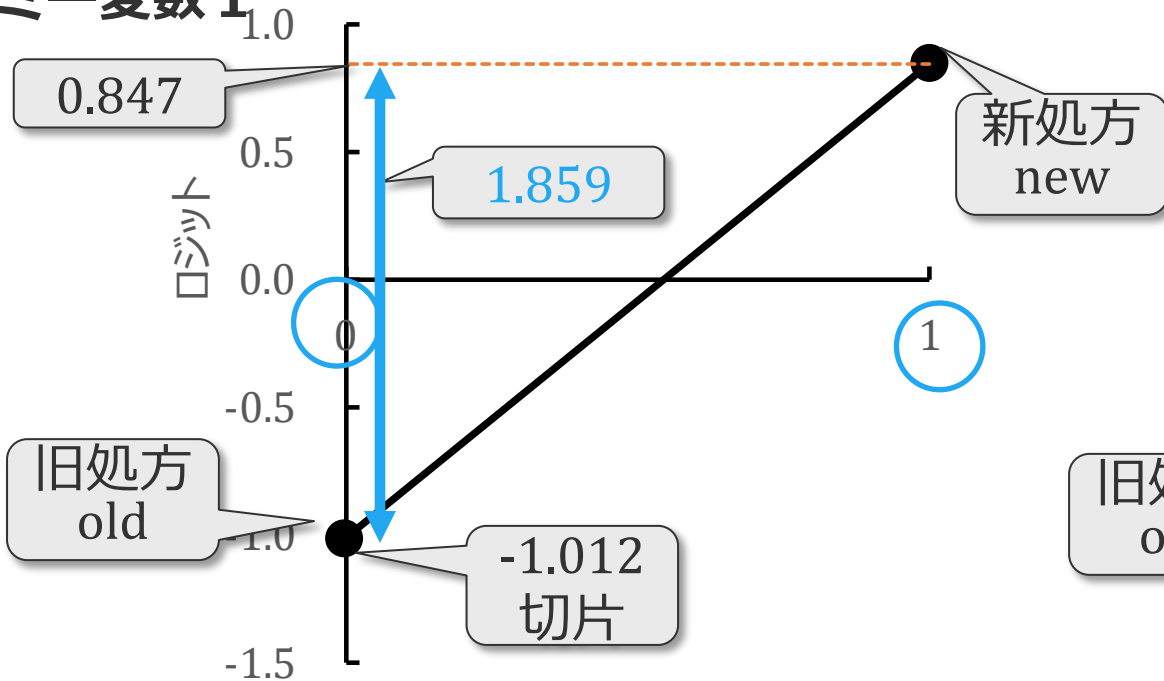
# 名義ロジスティックのあてはめ（2つの割合）

質的変数の contrasts が "contr.treatment" の場合の解

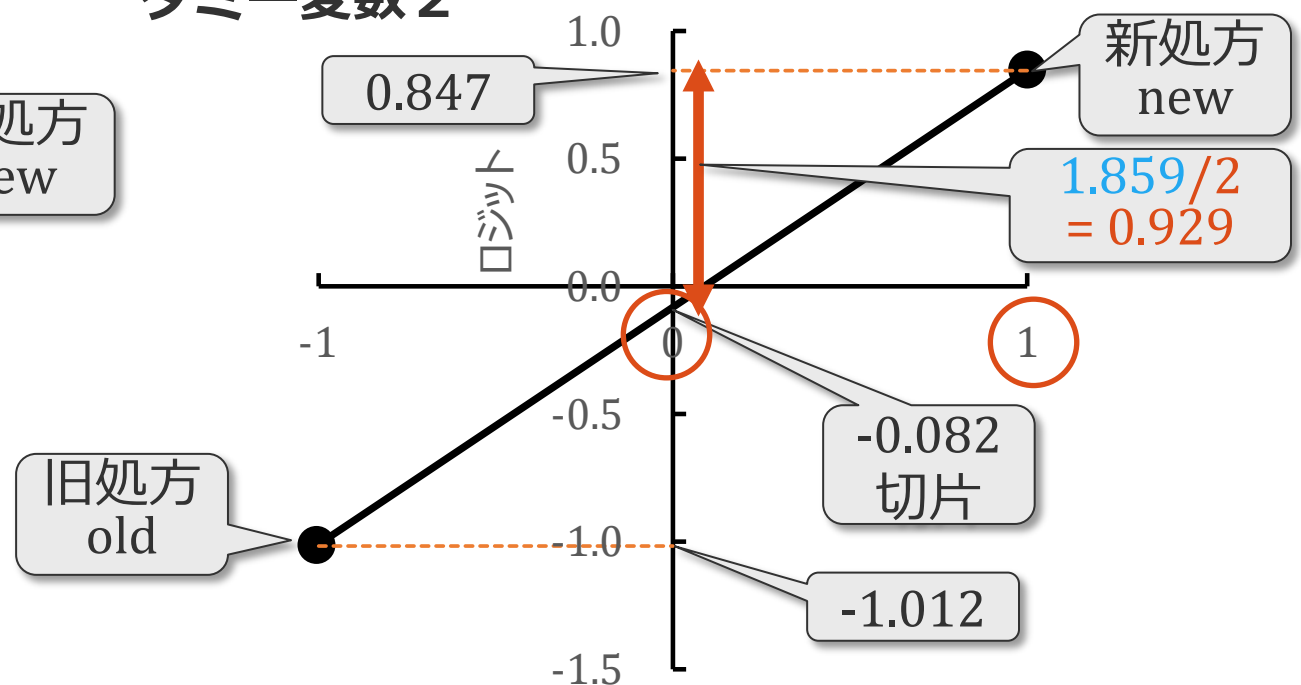
```
## Coefficients: (ダミー変数 1)
## Estimate Std. Error z value Pr
## (Intercept) -1.012 0.5839 -1.733 0.0832
## groupnew 1.859 0.9039 2.056 0.0397
```

	効果		計	p	対数オッズ (ロジット)
	有効	無効			
旧処方	4	11	15	0.267	-1.012
新処方	7	3	10	0.700	0.847
	差				1.859
	exp(差) : オッズ比				6.417
					0.929

ダミー変数 1



ダミー変数 2





# 名義ロジスティックのあてはめ（2つの割合）

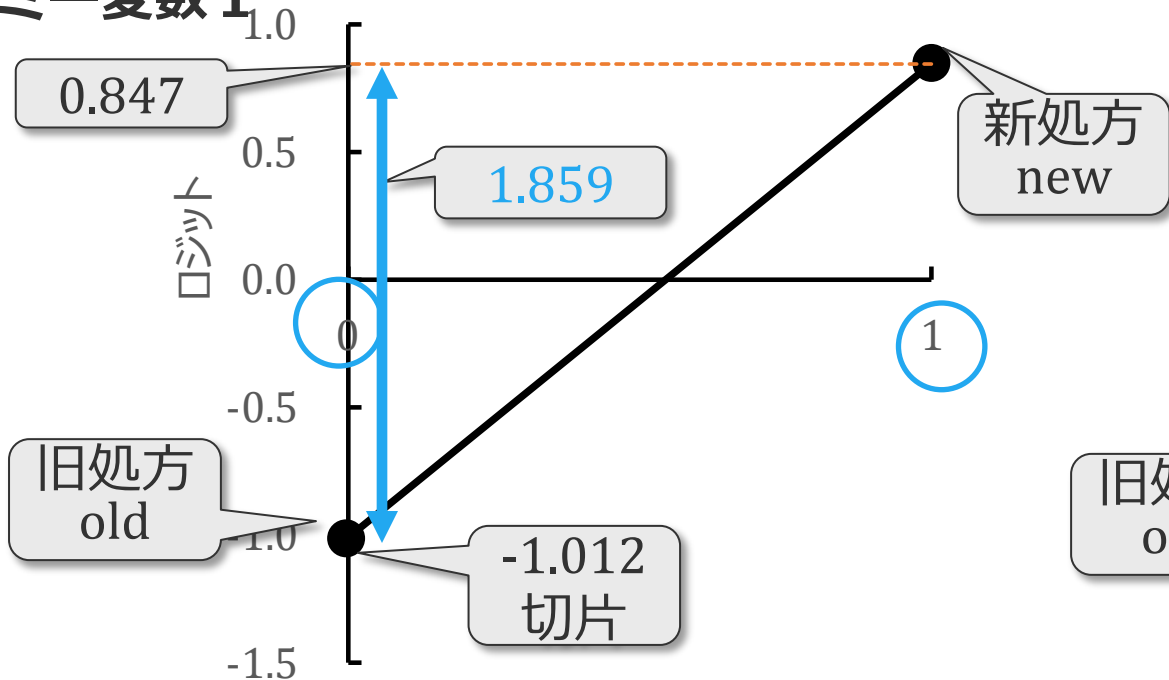
質的変数 group の contrasts が "contr.sum" の場合の解  
 # Coefficients: (ダミー変数 2)

##	Estimate	Std. Error	z value	Pr
## (Intercept)	-0.082	0.45197	-0.182	0.8558
## group1	0.929	0.45197	-2.056	0.0397

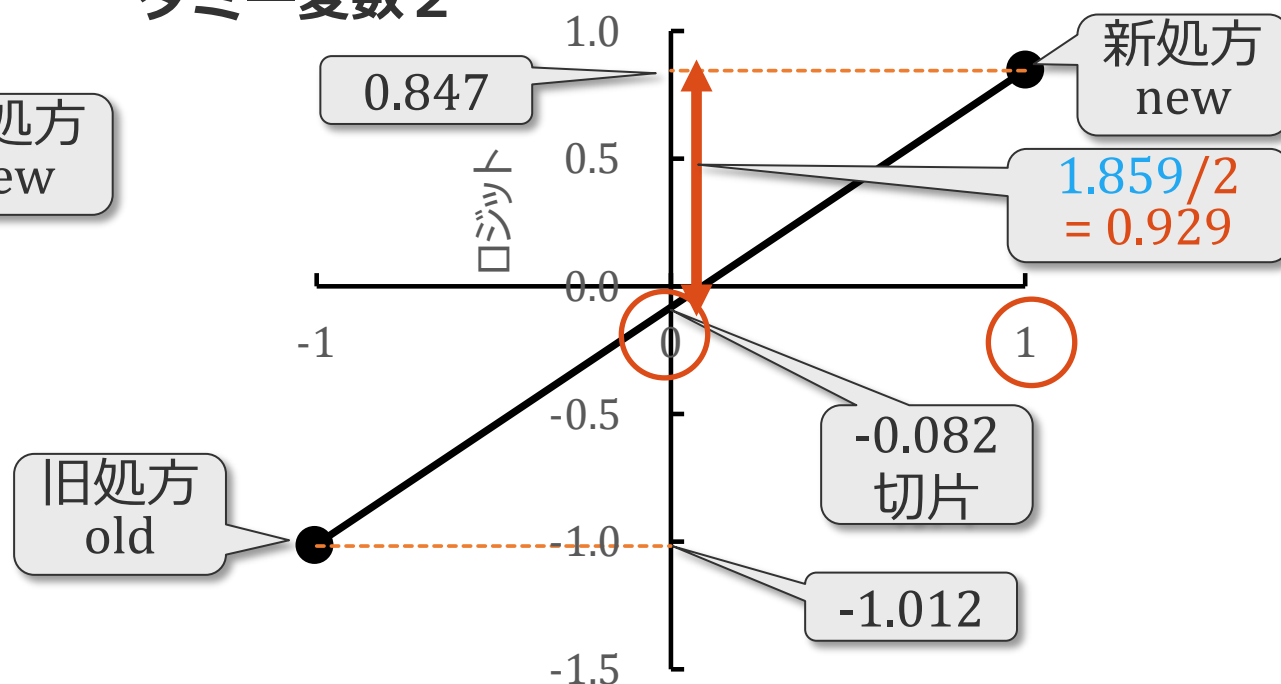
	効果		計	p	対数オッズ (ロジット)
	有効	無効			
旧処方	4	11	15	0.267	-1.012
新処方	7	3	10	0.700	0.847

差 1.859 0.929  
 exp(差) : オッズ比 6.417 2.533

ダミー変数 1



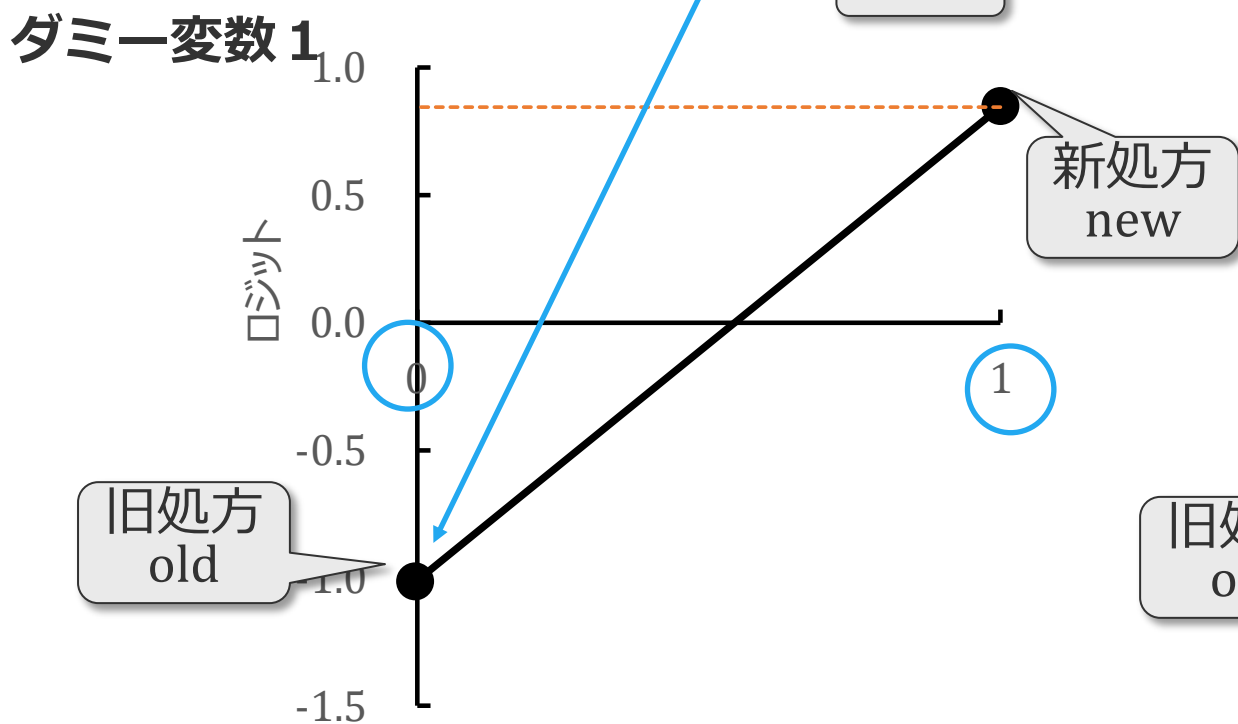
ダミー変数 2



# 名義ロジスティックのあてはめ (2つの割合)

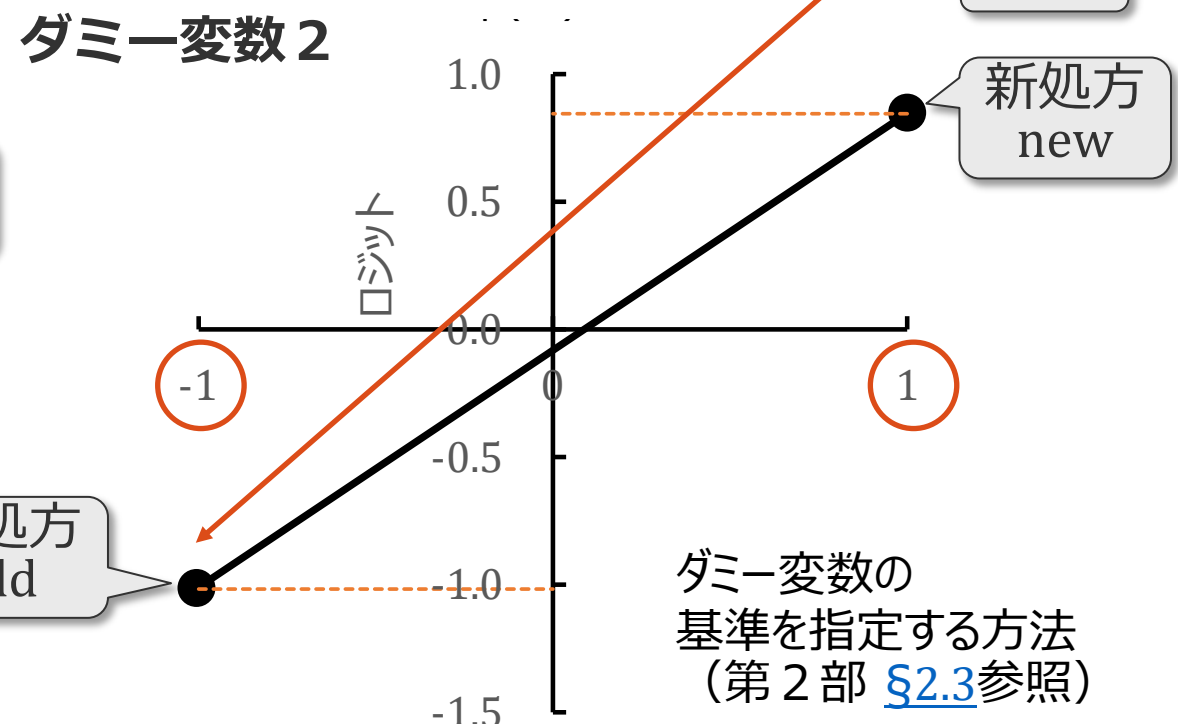
質的変数の contrasts が "contr.treatment" の場合  
`df$group <- factor(df$group, levels = c("old", "new"))`

基準は levels で指定した先頭の水準  
0 1  
先頭



質的変数の contrasts が "contr.sum" の場合  
`df$group <- factor(df$group, levels = c("new", "old"))`

基準は levels で指定した末尾の水準  
1 -1  
末尾



ダミー変数の基準を指定する方法 (第2部 §2.3参照)

# 名義ロジスティックのあてはめ（2つの割合）

p.180

## ●表示3.3.12 JMP[モデルのあてはめ]による解析結果

スクリプトファイル：Green3-3-3c.R

利用した関数

glm、anova、car::Anova

方法

効果の尤度比検定

あてはめたモデルの結果と、  
nullモデルをあてはめた結果を、  
anova関数で比較

car::Anova関数の利用

切片のみのモデル(null model)

```
glm_out0 <- glm(event ~ 1, data = df,  
                family = binomial(link = "logit"))
```

```
anova(glm_out0, glm_out, test = "Chisq")
```

```
## Analysis of Deviance Table
```

```
## Model 1: event ~ 1
```

```
## Model 2: event ~ group
```

```
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
```

```
## 1         3      34.296
```

```
## 2         2      29.615 1  4.6817 0.03048 *
```

```
Anova(glm_out, type = 2)
```

```
## Analysis of Deviance Table (Type II tests)
```

```
## Response: event
```

```
##   LR Chisq Df Pr(>Chisq)
```

```
## group 4.6817 1 0.03048 *
```

# 名義ロジスティックのあてはめ（2つの割合）

- 表示3.3.12 JMP[モデルのあてはめ]による解析結果

スクリプトファイル：Green3-3-3c.R

利用した関数：epiDisplay::logistic.display

	効果		計	p	対数オッズ (ロジット)	
	有効	無効				
旧処方	4	11	15	0.267	-1.012	
新処方	7	3	10	0.700	0.847	
差					1.859	0.929
exp(差) : オッズ比					6.417	2.533

```
logistic.display(glm_out)
```

```
## Logistic regression predicting event : yes vs no
```

```
##
```

```
## OR(95%CI) P(Wald's test) P(LR-test)
```

```
## group: new vs old 6.42 (1.09,37.73) 0.04 0.03
```

```
## Log-likelihood = -14.8074
```

```
## No. of observations = 25
```

```
## AIC value = 33.6147
```

```
exp(cbind(coef(glm_out), confint(glm_out)))
```

```
## 2.5 % 97.5 %
```

```
## (Intercept) 0.3636364 0.1007758 1.063378
```

```
## groupnew 6.4166667 1.1850716 43.860574
```



## $a \times 2$ 分割表

名義ロジスティックのあてはめ（3つ以上の割合）

# 名義ロジスティックのあてはめ（3つ以上の割合）

## ● 表示3.3.15 JMP[モデルのあてはめ]の出力

スクリプトファイル：Green3-3-3d.R

利用した関数：read\_excel::read\_excel、factor

方法

Excelの2つのシートからデータを読み込み

(データフレーム：df1、df2)

ダミー変数1とダミー変数2（第2部 [§2.3](#)）

4 × 2 分割表で、4つの割合の比較

##	yes	no	Sum	p
## A0	4	11	15	0.2666667
## A1	8	5	13	0.6153846
## A2	12	8	20	0.6000000
## A3	10	6	16	0.6250000

df1

ダミー変数

##	group	event	freq	dA1	dA2	dA3
## 1	A0	yes	4	0	0	0
## 2	A0	no	11	0	0	0
## 3	A1	yes	8	1	0	0
## 4	A1	no	5	1	0	0
## 5	A2	yes	12	0	1	0
## 6	A2	no	8	0	1	0
## 7	A3	yes	10	0	0	1
## 8	A3	no	6	0	0	1

df2

ダミー変数2

##	group	event	freq	ddA1	ddA2	ddA3
## 1	A0	yes	4	-1	-1	-1
## 2	A0	no	11	-1	-1	-1
## 3	A1	yes	8	1	0	0
## 4	A1	no	5	1	0	0
## 5	A2	yes	12	0	1	0
## 6	A2	no	8	0	1	0
## 7	A3	yes	10	0	0	1
## 8	A3	no	6	0	0	1

# 名義ロジスティックのあてはめ（3つ以上の割合）

- 表示3.3.15 JMP[モデルのあてはめ]の出力  
スクリプトファイル：Green3-3-3d.R  
利用した関数：read\_excel::read\_excel、factor  
方法

```
df1 <- read_excel("Green3-3.xlsx",  
                  sheet = "33-ax2-1")  
df1 <- data.frame(df1)  
  
df1$group <- factor(df1$group,  
                   levels = c("A0", "A3", "A2", "A1"))  
  
df1$event <- factor(df1$event,  
                   levels = c("no", "yes"))
```

"A0"を基準  
デフォルト：先頭  
contr.sum：末尾

オッズ yes/no

df1				ダミー変数		
##	group	event	freq	dA1	dA2	dA3
## 1	A0	yes	4	0	0	0
## 2	A0	no	11	0	0	0
## 3	A1	yes	8	1	0	0
## 4	A1	no	5	1	0	0
## 5	A2	yes	12	0	1	0
## 6	A2	no	8	0	1	0
## 7	A3	yes	10	0	0	1
## 8	A3	no	6	0	0	1

df2				ダミー変数2		
##	group	event	freq	ddA1	ddA2	ddA3
## 1	A0	yes	4	-1	-1	-1
## 2	A0	no	11	-1	-1	-1
## 3	A1	yes	8	1	0	0
## 4	A1	no	5	1	0	0
## 5	A2	yes	12	0	1	0
## 6	A2	no	8	0	1	0
## 7	A3	yes	10	0	0	1
## 8	A3	no	6	0	0	1

# 名義ロジスティックのあてはめ（3つ以上の割合）

- 表示3.3.15 JMP[モデルのあてはめ]の出力

スクリプトファイル

Green3-3-3d.R

利用した関数

glm、summary、

方法

質的変数 group の contrasts を  
デフォルトの contr.treatment で  
解析

JMP の結果と不一致

group A0を基準とした差が表示

表示3.3.15  
A0 に対する  
ロジットの差

説明変数  
質的変数 group

```
glm_out <- glm(event ~ group,  
               weights = freq, data = df1,  
               family = binomial(link = "logit"))
```

```
summary(glm_out)  
##  
## Coefficients:  
##           Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -1.0116    0.5839  -1.733  0.0832 .  
## groupA3      1.5224    0.7795   1.953  0.0508 .  
## groupA2      1.4171    0.7411   1.912  0.0559 .  
## groupA1      1.4816    0.8160   1.816  0.0694 .
```



- 表示3.3.15 JMP[モデルのあてはめ]の出力

スクリプトファイル

Green3-3-3d.R

利用した関数

glm、summary

方法

質的変数の代わりに

ダミー変数 1 を使用

JMP の結果と不一致

質的変数の contrasts を

デフォルトの contr.treatment で

解析した結果と一致

説明変数  
ダミー変数 1

```
glm_out1 <- glm(event ~ dA1 + dA2 + dA3,  
                weights = freq, data = df1,  
                family = binomial(link = "logit"))
```

```
summary(glm_out1)  
##  
## Coefficients:  
##           Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -1.0116    0.5839  -1.733   0.0832 .  
## dA1          1.4816    0.8160   1.816   0.0694 .  
## dA2          1.4171    0.7411   1.912   0.0559 .  
## dA3          1.5224    0.7795   1.953   0.0508 .
```

- 表示3.3.15 JMP[モデルのあてはめ]の出力

スクリプトファイル

Green3-3-3d.R

利用した関数

glm、summary

方法

質的変数の代わりに

ダミー変数 2 を使用

JMP の結果と一致

説明変数  
ダミー変数 2

```
glm_out2 <- glm(event ~ ddA1 + ddA2 + ddA3,  
                weights = freq, data = df2,  
                family = binomial(link = "logit"))
```

```
summary(glm_out2)
```

```
## Coefficients:
```

##	Estimate	Std. Error	z value	Pr(> z )
## (Intercept)	0.09367	0.26703	0.351	0.726
## ddA1	0.37633	0.48354	0.778	0.436
## ddA2	0.31179	0.41890	0.744	0.457
## ddA3	0.41715	0.45237	0.922	0.356

# 名義ロジスティックのあてはめ（3つ以上の割合）

## ●表示3.3.15 JMP[モデルのあてはめ]の出力

スクリプトファイル

Green3-3-3d.R

利用した関数

glm、summary

方法

factor 関数の levels 引数で  
基準の A0 を末尾に移動

質的変数 group の contrasts を  
contr.sum に設定して解析

JMP の結果と一致

A0, A1, A2, A3 との対応は非表示

基準

```
df1$group <- factor(df1$group,  
                    levels = c("A1", "A2", "A3", "A0"))  
  
glm_out3 <- glm(event ~ group,  
                weights = freq, data = df1,  
                family = binomial(link = "logit"),  
                contrasts = list(group = contr.sum))
```

```
summary(glm_out3)  
##  
## Coefficients:  
##           Estimate Std. Error z value Pr(>|z|)  
## (Intercept)  0.09367   0.26703   0.351   0.726  
## group1      0.37633   0.48354   0.778   0.436  
## group2      0.31179   0.41890   0.744   0.457  
## group3      0.41715   0.45237   0.922   0.356
```

contrasts の設定

# 名義ロジスティックのあてはめ（3つ以上の割合）

- 表示3.3.15 JMP[モデルのあてはめ]の出力

表示3.3.16 Excelによる尤度比検定

スクリプトファイル：Green3-3-3d.R

利用した関数

glm、summary、anova、car::Anova

方法

効果の尤度比検定

anova 関数でnullモデルと比較

JMPの結果と一致

car::Anova 関数の利用

切片だけのモデル  
(nullモデル)

```
glm_out0 <- glm(event ~ 1, data = df1,
                family = binomial(link = "logit"))

anova(glm_out0, glm_out, test = "Chisq")

## Analysis of Deviance Table
##
## Model 1: cbind(yes, no) ~ 1
## Model 2: cbind(yes, no) ~ group
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         3     5.6615
## 2         0     0.0000  3   5.6615  0.1293

Anova(glm_out3, type = 2)

## Response: event
##   LR Chisq Df Pr(>Chisq)
## group  5.6615  3   0.1293
```

# 名義ロジスティックのあてはめ（3つ以上の割合）

p.184

## ●表示3.3.15 JMP[モデルのあてはめ]の出力

スクリプトファイル：Green3-3-3d.R

利用した関数

glm、epiDisplay::logistic.display

方法

オッズ比とその信頼区間の推定

group の contrasts をデフォルト ("contr.treatment") で実行した glm の結果を利用

```
df1$group <- factor(df1$group,  
                    levels = c("A0", "A3", "A2", "A1"))
```

```
df1$event <- factor(df1$event,  
                    levels = c("no", "yes"))
```

```
logistic.display(glm_out)  
## Logistic regression predicting event : yes vs no  
##  
## OR(95%CI) P(Wald's test) P(LR-test)  
## group: ref.=A0 0.129  
## A3 4.58 (0.99,21.12) 0.051  
## A2 4.12 (0.97,17.63) 0.056  
## A1 4.4 (0.89,21.78) 0.069
```



# $a \times 2$ 分割表

割合の差の検定

割合の差の検定 (多重比較)

# a × 2 分割表：割合の差の検定

## ●表示3.3.14 データとJMP用データ表

スクリプトファイル：Green3-3-3f.R

利用した関数：proportions、barplot

方法

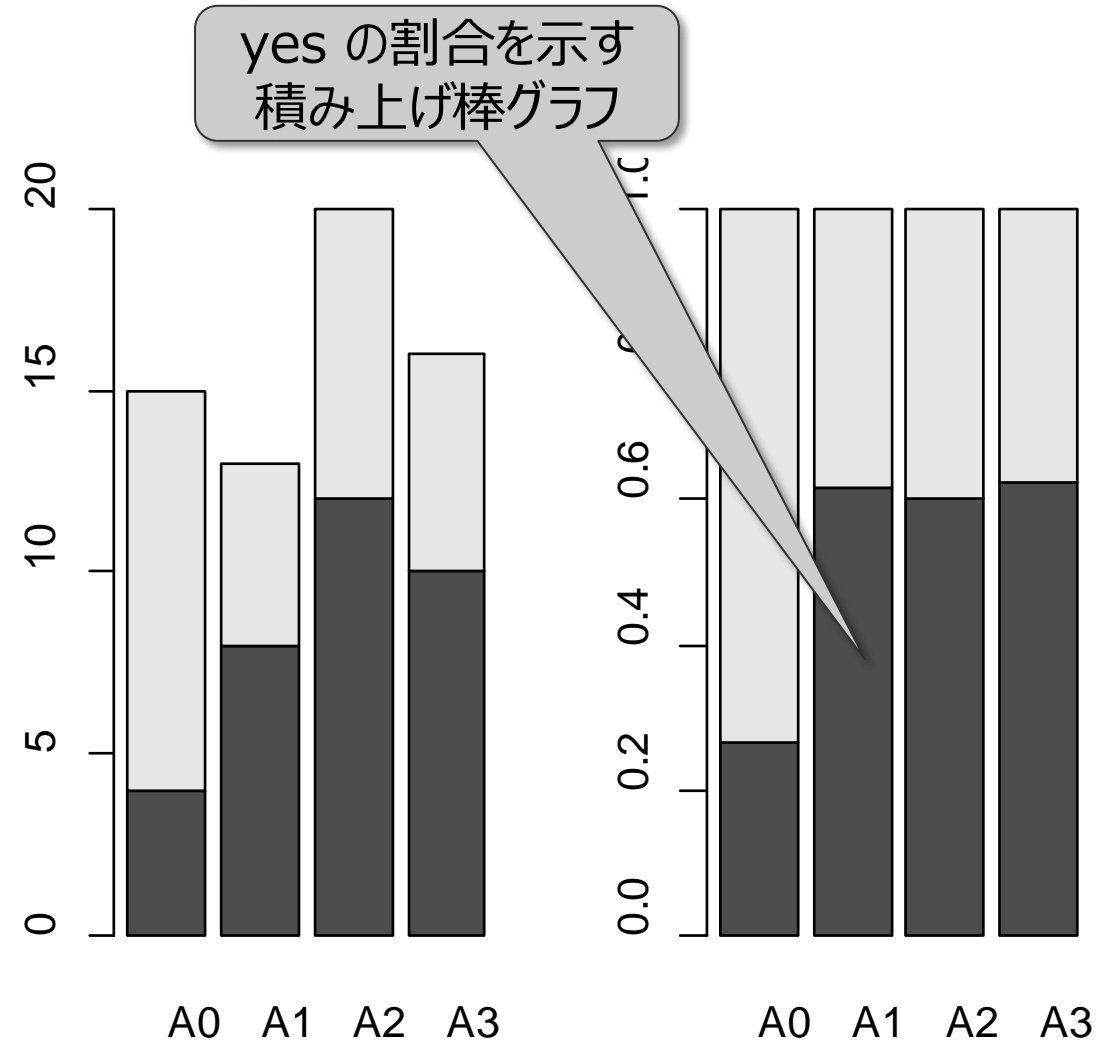
a × 2 分割表の全体の比較、グラフ化

4 × 2 分割表

##	event	yes	no
##	group		
##	A0	4	11
##	A1	8	5
##	A2	12	8
##	A3	10	6

```
mx <- proportions(tb, 1)
barplot(t(tb))
barplot(t(mx))
```

##	event	yes	no
##	group		
##	A0	0.267	0.733
##	A1	0.615	0.385
##	A2	0.600	0.400
##	A3	0.625	0.375





# a × 2 分割表：割合の差の検定

## ● 全体の比較

スクリプトファイル：Green3-3-3f.R

利用した関数：chisq.test、fisher.test

DescTools::GTest

方法：a × 2 分割表の全体の検定

カイ 2 乗検定

尤度比検定

Fisher の正確検定

tb	event		
##	group	yes	no
##	A0	4	11
##	A1	8	5
##	A2	12	8
##	A3	10	6

mx	event		
##	group	yes	no
##	A0	0.266	0.733
##	A1	0.615	0.384
##	A2	0.600	0.400
##	A3	0.625	0.375

```
chisq.test(tb, correct = FALSE)
## Pearson's Chi-squared test
##
## data:  tb
## X-squared = 5.5306, df = 3, p-value = 0.1368

GTest(tb)
## Log likelihood ratio (G-test) test
  of independence without correction
## data:  tb
## G = 5.6615, X-squared df = 3, p-value = 0.1293

fisher.test(tb)
## Fisher's Exact Test for Count Data
##
## data:  tb
## p-value = 0.1498
## alternative hypothesis: two.sided
```



# a × 2 分割表：割合の差の検定（多重比較）

- ペアごとの比較

スクリプトファイル

Green3-3-3f.R

利用した関数

pairwise.prop.test

方法

ペアごとのカイ 2 乗検定

多重比較（第 2 部 [§1.3](#) 参照）

4 × 2 分割表と割合 p

```
##      yes no Sum      p
## A0    4 11  15 0.2666667
## A1    8  5  13 0.6153846
## A2   12  8  20 0.6000000
## A3   10  6  16 0.6250000
```

"holm"、"hochberg"、"hommel",  
"bonferroni"、"BH"、"BY"、"fdr"、"none"

```
pairwise.prop.test(tb, p.adjust.method = "holm",
                    correct = FALSE)

## Warning in prop.test(x[c(i, j)], n[c(i, j)], ...):
## Chi-squared approximation may be incorrect

##
## Pairwise comparisons using Pairwise comparison of
## proportions

##
## data:  tb
##      A0  A1  A2
## A1 0.27 -   -
## A2 0.27 1.00 -
## A3 0.27 1.00 1.00
##
## P value adjustment method: holm
```

Holm 法は、p 値の低い  
組み合わせから検定開始、  
有意でなくなった時点で  
検定終了（第 2 部 [§1.3](#) 参照）

# a × 2 分割表：割合の差の検定（多重比較）

- 3つ以上の割合の多重比較

スクリプトファイル

Green3-3-3f.R

利用した関数

RVAideMemoire::fisher.multcomp

方法

ペアごとの Fisher の正確検定

多重比較（第2部 [§1.3](#) 参照）

4 × 2 分割表と割合 p

##		yes	no	Sum	p
##	A0	4	11	15	0.2666667
##	A1	8	5	13	0.6153846
##	A2	12	8	20	0.6000000
##	A3	10	6	16	0.6250000

"holm"、"hochberg"、"hommel",  
"bonferroni"、"BH"、"BY"、"fdr"、"none"

```
fisher.multcomp(tb, p.method = "holm")
##
## Pairwise comparisons using Fisher's exact test
## for count data
##
## data:  tb
##
##           A0  A1  A2
## A1 0.4993  -  -
## A2 0.4390  1  -
## A3 0.4390  1  1
##
## P value adjustment method: holm
```

Holm 法は、p 値の低い組み合わせから検定開始、有意でなくなった時点で検定終了（第2部 [§1.3](#) 参照）

# a × 2 分割表：割合の差の検定（多重比較）

- 3つ以上の割合の多重比較

スクリプトファイル

Green3-3-3f.R

利用した関数

rcompanion::

pairwiseNominalIndependence

方法

ペアごとの Fisher の正確検定、  
尤度比検定、カイ 2 乗検定  
多重比較（第 2 部 [§1.3](#) 参照）

分割表

表側

##		yes	no	Sum	p
##	A0	4	11	15	0.2666667
##	A1	8	5	13	0.6153846
##	A2	12	8	20	0.6000000
##	A3	10	6	16	0.6250000

```
pairwiseNominalIndependence(  
  tb,  
  compare = "row",  
  fisher = FALSE,  
  gtest = TRUE,  
  chisq = TRUE,  
  method = "holm",  
  correct = "none",  
  cramer = FALSE,  
  digits = 4)
```

比較する群の場所  
row / column

検定方法の選択

多重比較法の選択

本来は、実験前に、  
多重比較法を選択

```
## Comparison p.Gtest p.adj.Gtest p.Chisq p.adj.Chisq  
## 1 A0 : A1 0.06055 0.2548 0.1397 0.603  
## 2 A0 : A2 0.04701 0.2548 0.1061 0.603  
## 3 A0 : A3 0.04246 0.2548 0.1005 0.603  
## 4 A1 : A2 0.92950 1.0000 1.0000 1.000  
## 5 A1 : A3 0.95770 1.0000 1.0000 1.000  
## 6 A2 : A3 0.87840 1.0000 1.0000 1.000
```



- 引用文献

藤井良宜（2010）カテゴリカルデータ解析、共立出版

- 作成

片瀬雅彦

- 作成時期

2021年11月20日