

Schad et al (2020)に学ぶ名義尺度のコーディング

Schad et al (2020)日本語注解

小川雅貴

東京大学

2021/12/17 (updated: 2021-12-09)

このHTMLスライドをご覧になる際は、ChromeまたはFirefoxをお使いになることをお勧めいたします。また、HTMLスライドの中身が正しく読み込まれない場合には、ブラウザの更新ボタンを押して、再度スライドを読み込んでください。なお、本スライドのPDF版は [こちら](#) にございます。

必要なパッケージ・関数の読み込み

```
library(tidyverse)
```

次のパッケージ群を呼び出す

- **ggplot2**: データ可視化・図生成
- **dplyr**: データ操作
- **tidyr**: データ整然化
- **readr**: データ読み込み
- **purrr**: 関数型プログラミング
- **tibble**: 現代的データフレーム Δ tibbles
- **stringr**: 文字列処理
- **forcats**: カテゴリ変数の操作

```
library(tidyverse)
```

```
library(broom)
```

統計モデルの解析結果をtibbleで表示

```
library(tidyverse)
library(broom)
library(lme4)
```

一般化線形混合モデルの構築

```
library(tidyverse)
library(broom)
library(lme4)
library(MASS)
```

対比行列の構築

```
library(tidyverse)
library(broom)
library(lme4)
library(MASS)
library(hypr)
```

「仮説」行列の構築


```
library(tidyverse)
library(broom)
library(lme4)
library(MASS)
library(hypr)
library(knitr)
```

Rでの動的文書生成

```
library(tidyverse)
library(broom)
library(lme4)
library(MASS)
library(hypr)
library(knitr)

source(
  "http://read.psych.uni-potsdam.de/attachments/arti
  local = knitr::knit_global()
)
```

mixedDesign()の読み込み

Rmarkdownをknitする際には, `source()`の項で`local = knitr::knit_global()`を指定する
<https://bookdown.org/yihui/rmarkdown-cookbook/source-script.html>

```
library(tidyverse)
library(broom)
library(lme4)
library(MASS)
library(hypr)
library(knitr)

source(
  "http://read.psych.uni-potsdam.de/attachments/arti
  local = knitr::knit_global()
)

#source("http://read.psych.uni-potsdam.de/pmr2/attac
```

Schad et al (2020)の元コードは[こちら](#)

乱数の固定

```
library(tidyverse)
library(broom)
library(lme4)
library(MASS)
library(hypr)
library(knitr)

source(
  "http://read.psych.uni-potsdam.de/attachments/arti
  local = knitr::knit_global()
)

#source("http://read.psych.uni-potsdam.de/pmr2/attac

set.seed(1212)
```

Conceptual explanation of default contrasts (p.3)

以下，各対比の和訳は **対訳辞書(英語対和訳)**に基づく

Treatment contrast 処理対比

- 実験的介入treatmentの効果を，非介入群との比較で検討
 - 実験群 vs 統制群
- 各実験群を同一の統制群と比べる

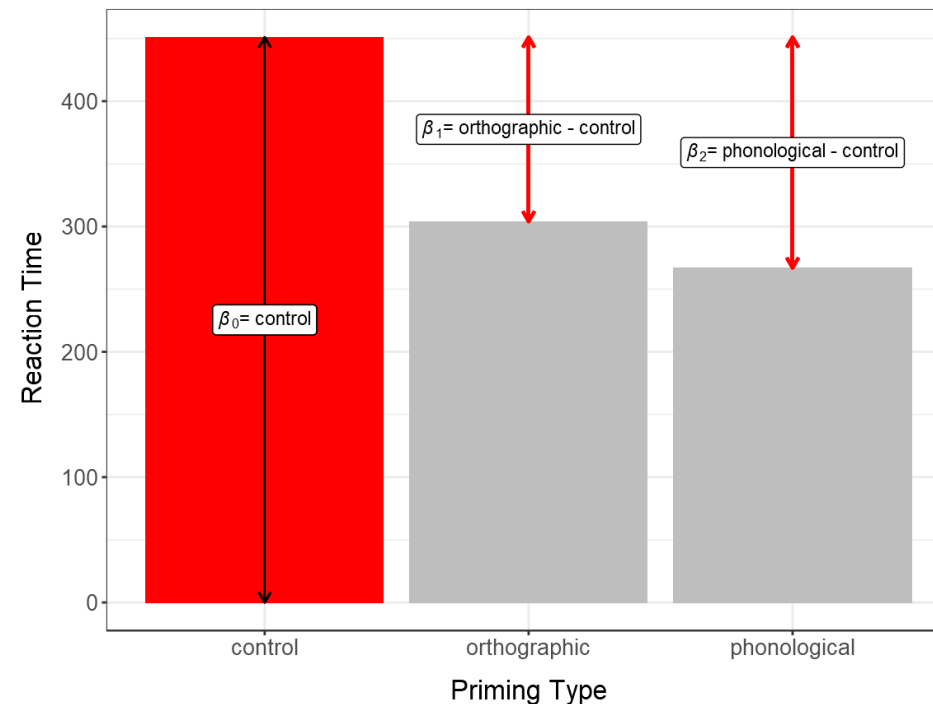
Treatment contrast 処理対比

- 実験的介入treatmentの効果を，非介入群との比較で検討
 - 実験群 vs 統制群
- 各実験群を同一の統制群と比べる
- 例：様々なプライミング効果を，同一の統制群と比べる
 1. 音韻的プライミング
 2. 綴字プライミング
 3. プライミングなし（統制群）

Treatment contrast 処理対比

- 実験的介入treatmentの効果を，非介入群との比較で検討
 - 実験群 vs 統制群
- 各実験群を同一の統制群と比べる
- 例：様々なプライミング効果を，同一の統制群と比べる
 1. 音韻的プライミング
 2. 綴字プライミング
 3. プライミングなし（統制群）
- 対比は次の2つ
 1. 音韻的プライミング vs プライミングなし（統制群）

dummy contrastとも呼ばれる



Sum contrast 零和対比

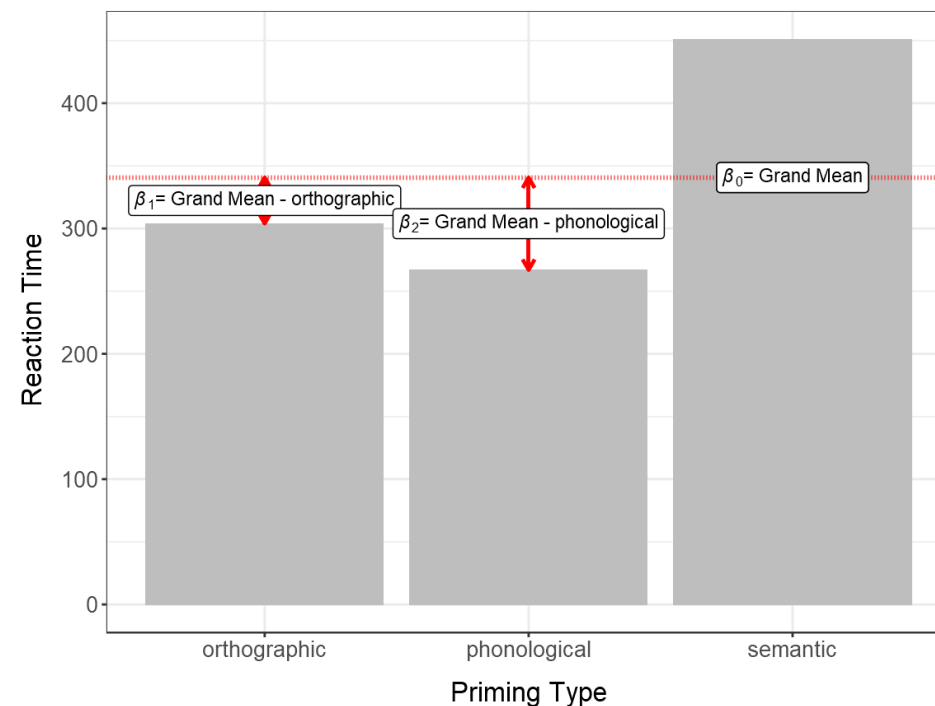
- 実験的介入treatmentの効果を, 全ての群の平均との比較で検討
 - 実験群 vs 全体平均

Sum contrast 零和対比

- 実験的介入treatmentの効果を，全ての群の平均との比較で検討
 - 実験群 vs 全体平均
- 例：様々なプライミング効果を，全体平均と比べる
 1. 音韻的プライミング
 2. 綴字プライミング
 3. プライミングなし（統制群）

Sum contrast 零和対比

- 実験的介入treatmentの効果を，全ての群の平均との比較で検討
 - 実験群 vs 全体平均
- 例：様々なプライミング効果を，全体平均と比べる
 - 音韻的プライミング
 - 綴字プライミング
 - プライミングなし（統制群）
- 対比は次の2つ
 - 音韻的プライミング vs 全体平均
 - 綴字プライミング vs 全体平均



Repeated contrast

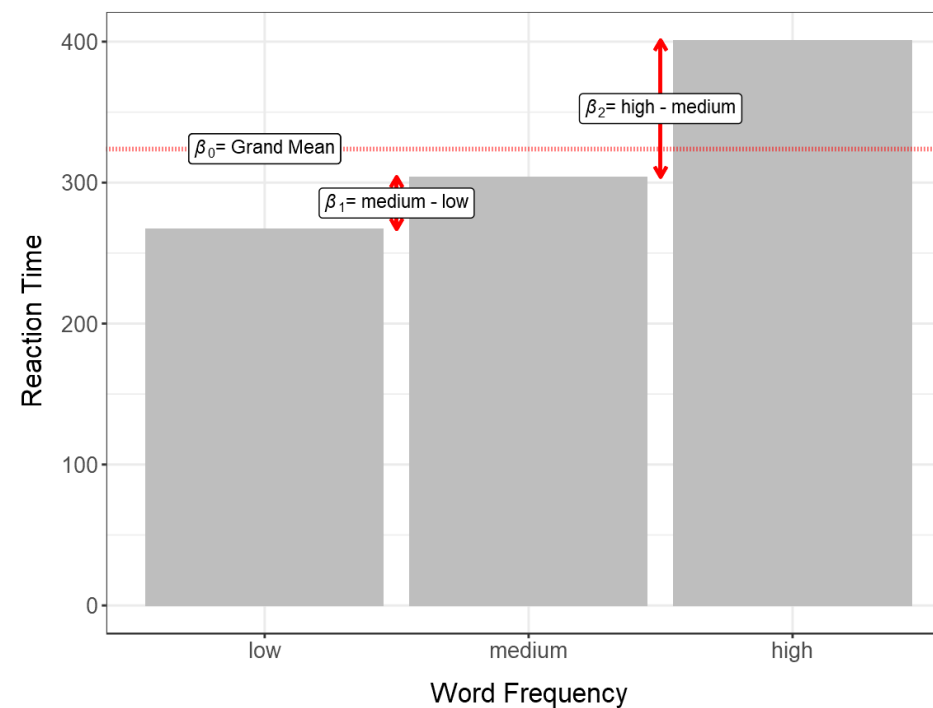
- （順序が決まっている）隣り合う水準同士を比較
 - 実験群1 vs 実験群2
 - 実験群2 vs 実験群3...

Repeated contrast

- （順序が決まっている）隣り合う水準同士を比較
 - 実験群1 vs 実験群2
 - 実験群2 vs 実験群3...
- 例：語の頻度の効果を，隣り合う水準と比べる
 1. 低頻度
 2. 中頻度
 3. 高頻度

Repeated contrast

- (順序が決まっている) 隣り合う水準同士を比較
 - 実験群1 vs 実験群2
 - 実験群2 vs 実験群3...
- 例：語の頻度の効果を，隣り合う水準と比べる
 1. 低頻度
 2. 中頻度
 3. 高頻度
- 対比は次の2つ
 1. 中頻度 vs 低頻度
 2. 高頻度 vs 中頻度



Polynomial contrast 多項式対比

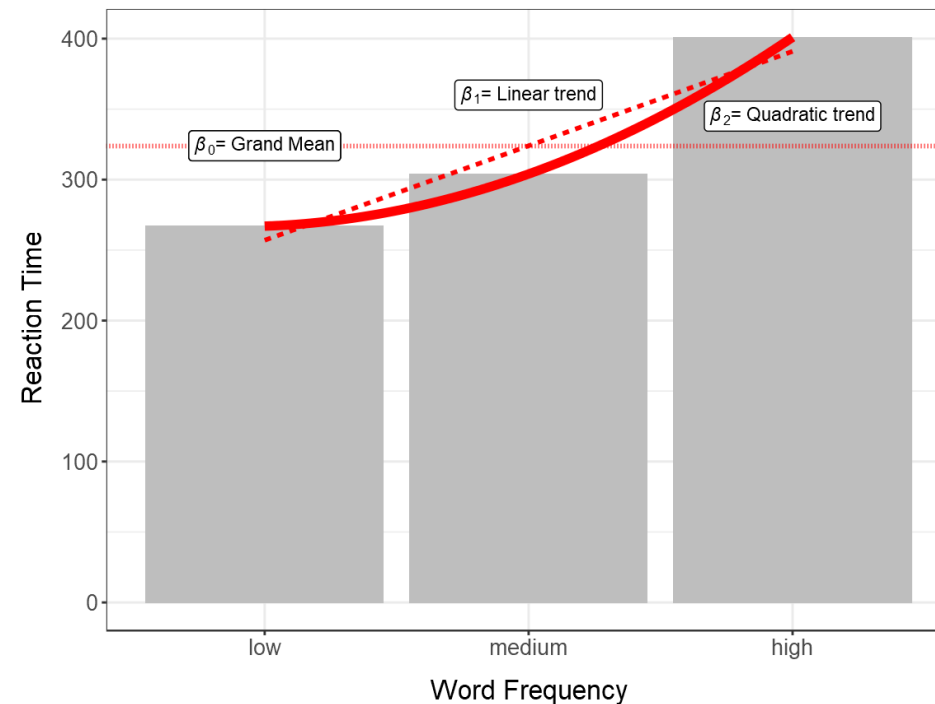
- （順序と間隔が決まっている）隣り合う水準間の増減傾向を比較
 - 水準間で同じペースで増加・減少？
 - 水準間毎に、増加・減少のペースが変わる？

Polynomial contrast 多項式対比

- （順序と間隔が決まっている）隣り合う水準間の増減傾向を比較
 - 水準間で同じペースで増加・減少？
 - 水準間毎に，増加・減少のペースが変わる？
- 例：語の頻度の効果を，隣り合う水準と比べる
 1. 低頻度
 2. 中頻度
 3. 高頻度

Polynomial contrast 多項式対比

- （順序と間隔が決まっている）隣り合う水準間の増減傾向を比較
 - 水準間で同じペースで増加・減少？
 - 水準間毎に，増加・減少のペースが変わる？
- 例：語の頻度の効果を，隣り合う水準と比べる
 - 低頻度
 - 中頻度
 - 高頻度
- 対比は次の2つ
 - 1次関数ペース（Linear; どの水準間でも一定ペースで増加・減少？）
 - 2次関数ペース（Quadratic; 次の水準間ほどに増加・減少が激しい・穏やか？）



This type of coding system should be used only with an ordinal variable in which the levels are equally spaced

Helmert contrast ヘルマート対比

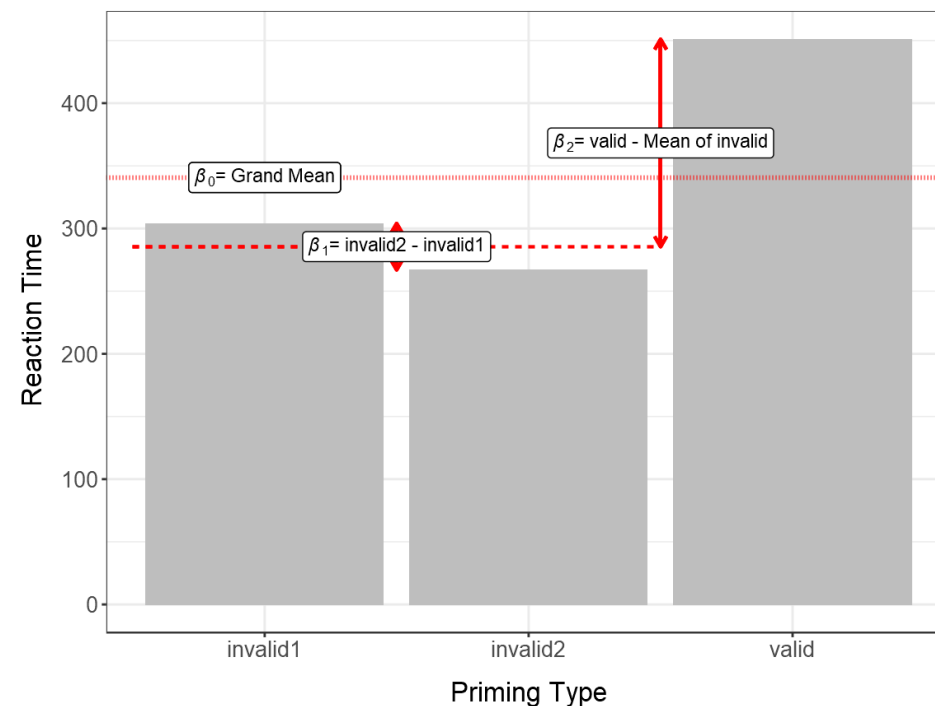
- 「1つの水準を残りの水準の平均と比較する」 ことを繰り返す
 - 水準1 vs 残りの水準（水準2と水準3の平均）
 - 水準2 vs 残りの水準（水準3）

Helmert contrast ヘルマート対比

- 「1つの水準を残りの水準の平均と比較する」ことを繰り返す
 - 水準1 vs 残りの水準（水準2と水準3の平均）
 - 水準2 vs 残りの水準（水準3）
- 例：様々なプライミング効果を，同一の統制群と比べる
 1. 無効なプライミング1
 2. 無効なプライミング2
 3. 有効なプライミング

Helmert contrast ヘルマート対比

- 「1つの水準を残りの水準の平均と比較する」ことを繰り返す
 - 水準1 vs 残りの水準 (水準2と水準3の平均)
 - 水準2 vs 残りの水準 (水準3)
- 例：様々なプライミング効果を，同一の統制群と比べる
 - 無効なプライミング1
 - 無効なプライミング2
 - 有効なプライミング
- 対比は次の2つ
 - 無効なプライミング1 vs 無効なプライミング2
 - 無効なプライミング vs 有効なプライミング



Basic concepts illustrated using a two-level factor (pp.3--8)

1要因2水準の実験で、応答変数の平均値を比較

1. シミュレーション用データを作成
2. シミュレーションの結果を図や表で確認
3. シミュレーション用データに対し線形モデルを構築
 - 。 とりあえず線形モデルを作るとどうなるか試す
4. Default contrast coding: treatment contrasts
5. Defining hypotheses
6. Sum contrast
7. Cell means parameterization

シミュレーション用データを作成

- `mixedDesign()` を使う
- 1要因2水準
 1. F1: 0.8秒
 2. F2: 0.4秒
- 実験参加者数10人
 - 被験者間計画 (between-subject design)
 - 実験参加者は, F1の条件だけ, またはF2の条件だけに接する
 1. F1: 5人
 2. F2: 5人


```

set.seed(1212); simdat <- mixedDesign(
#### 被験者間計画での水準数 ####
  B = 2,
#### 被験者内計画での水準数 ####
  W = NULL,
#### 被験者間計画での1水準あたりの実験参加者数 ####
  ##### 実験参加者総数は、`B * n` ##### ##### 実験参加者総数
  n = 5,
#### 水準ごとの応答変数の平均を示した行列 ####
#### 各行に水準ごとの応答変数の平均を入れる ####
  M = matrix(
    c(0.8, 0.4),
    nrow = 2,
    ncol = 1,
    byrow = FALSE
  ),
#### 要因の標準偏差 ####
  SD = 0.20,
#### 「1試行1データ」の整然データでデータフレームを作成 ####
  long = TRUE
)

set.seed(1212); simdat

```

	B_A	id	DV
1	A1	1	0.7370717
2	A1	2	0.5544852
3	A1	3	0.9634010
4	A1	4	1.0436117
5	A1	5	0.7014305
6	A2	6	0.2616614
7	A2	7	0.3948117
8	A2	8	0.1731352
9	A2	9	0.4832820
10	A2	10	0.6871097

```

set.seed(1212); simdat <- mixedDesign(
#### 被験者間計画での水準数 ####
  B = 2,
#### 被験者内計画での水準数 ####
  W = NULL,
#### 被験者間計画での1水準あたりの実験参加者数 ####
  ##### 実験参加者総数は、`B * n` ##### 実験参加者総数
  n = 5,
#### 水準ごとの応答変数の平均を示した行列 ####
#### 各行に水準ごとの応答変数の平均を入れる ####
  M = matrix(
    c(0.8, 0.4),
    nrow = 2,
    ncol = 1,
    byrow = FALSE
  ),
#### 要因の標準偏差 ####
  SD = 0.20,
#### 「1試行1データ」の整然データでデータフレームを作成 ####
  long = TRUE
) |>
  rename(F = B_A)

set.seed(1212); simdat

```

	F	id	DV
1	A1	1	0.7370717
2	A1	2	0.5544852
3	A1	3	0.9634010
4	A1	4	1.0436117
5	A1	5	0.7014305
6	A2	6	0.2616614
7	A2	7	0.3948117
8	A2	8	0.1731352
9	A2	9	0.4832820
10	A2	10	0.6871097

```

set.seed(1212); simdat <- mixedDesign(
#### 被験者間計画での水準数 ####
  B = 2,
#### 被験者内計画での水準数 ####
  W = NULL,
#### 被験者間計画での1水準あたりの実験参加者数 ####
  ##### 実験参加者総数は、`B * n` ##### 実験参加者総数
  n = 5,
#### 水準ごとの応答変数の平均を示した行列 ####
#### 各行に水準ごとの応答変数の平均を入れる ####
  M = matrix(
    c(0.8, 0.4),
    nrow = 2,
    ncol = 1,
    byrow = FALSE
  ),
#### 要因の標準偏差 ####
  SD = 0.20,
#### 「1試行1データ」の整然データでデータフレームを作成 ####
  long = TRUE
) |>
  rename(F = B_A) |>
  mutate(
    F = fct_recode(
      F,
      F1 = "A1",
      F2 = "A2"
    )
  )

set.seed(1212); simdat

```

	F	id	DV
1	F1	1	0.7370717
2	F1	2	0.5544852
3	F1	3	0.9634010
4	F1	4	1.0436117
5	F1	5	0.7014305
6	F2	6	0.2616614
7	F2	7	0.3948117
8	F2	8	0.1731352
9	F2	9	0.4832820
10	F2	10	0.6871097

```

set.seed(1212); simdat <- mixedDesign(
#### 被験者間計画での水準数 ####
  B = 2,
#### 被験者内計画での水準数 ####
  W = NULL,
#### 被験者間計画での1水準あたりの実験参加者数 ####
  ##### 実験参加者総数は、`B * n` ##### 実験参加者総数
  n = 5,
#### 水準ごとの応答変数の平均を示した行列 ####
#### 各行に水準ごとの応答変数の平均を入れる ####
  M = matrix(
    c(0.8, 0.4),
    nrow = 2,
    ncol = 1,
    byrow = FALSE
  ),
#### 要因の標準偏差 ####
  SD = 0.20,
#### 「1試行1データ」の整然データでデータフレームを作成 ####
  long = TRUE
) |>
  rename(F = B_A) |>
  mutate(
    F = fct_recode(
      F,
      F1 = "A1",
      F2 = "A2"
    )
  )

str(simdat)

set.seed(1212); simdat

```

```

'data.frame':  10 obs. of  3 variables:
 $ F : Factor w/ 2 levels "F1","F2": 1 1 1 1 1 2 2 2 2 2
 $ id: Factor w/ 10 levels "1","2","3","4",...: 1 2 3 4 5 6 7 8 9 10
 $ DV: num  0.737 0.554 0.963 1.044 0.701 ...

```

	F	id	DV
1	F1	1	0.7370717
2	F1	2	0.5544852
3	F1	3	0.9634010
4	F1	4	1.0436117
5	F1	5	0.7014305
6	F2	6	0.2616614
7	F2	7	0.3948117
8	F2	8	0.1731352
9	F2	9	0.4832820
10	F2	10	0.6871097

シミュレーションの結果を図や表で確認

- 1要因2水準
 1. F1: 0.8秒 (標準偏差0.2秒)
 2. F2: 0.4秒 (標準偏差0.2秒)

```
simdat
```

	F	id	DV
1	F1	1	0.7370717
2	F1	2	0.5544852
3	F1	3	0.9634010
4	F1	4	1.0436117
5	F1	5	0.7014305
6	F2	6	0.2616614
7	F2	7	0.3948117
8	F2	8	0.1731352
9	F2	9	0.4832820
10	F2	10	0.6871097

```
simdat |>
```

```
group_by(F)
```

```
# A tibble: 10 x 3
```

```
# Groups:   F [2]
```

	F	id	DV
	<fct>	<fct>	<dbl>
1	F1	1	0.737
2	F1	2	0.554
3	F1	3	0.963
4	F1	4	1.04
5	F1	5	0.701
6	F2	6	0.262
7	F2	7	0.395
8	F2	8	0.173
9	F2	9	0.483
10	F2	10	0.687

```
simdat |>
  group_by(F) |>
  summarise(
    N = length(DV),
    M = mean(DV),
    SD = sd(DV),
    SE = SD / sqrt(N)
  )
```

```
# A tibble: 2 x 5
  F         N     M   SD   SE
<fct> <int> <dbl> <dbl> <dbl>
1 F1         5  0.8  0.2 0.0894
2 F2         5  0.4  0.2 0.0894
```

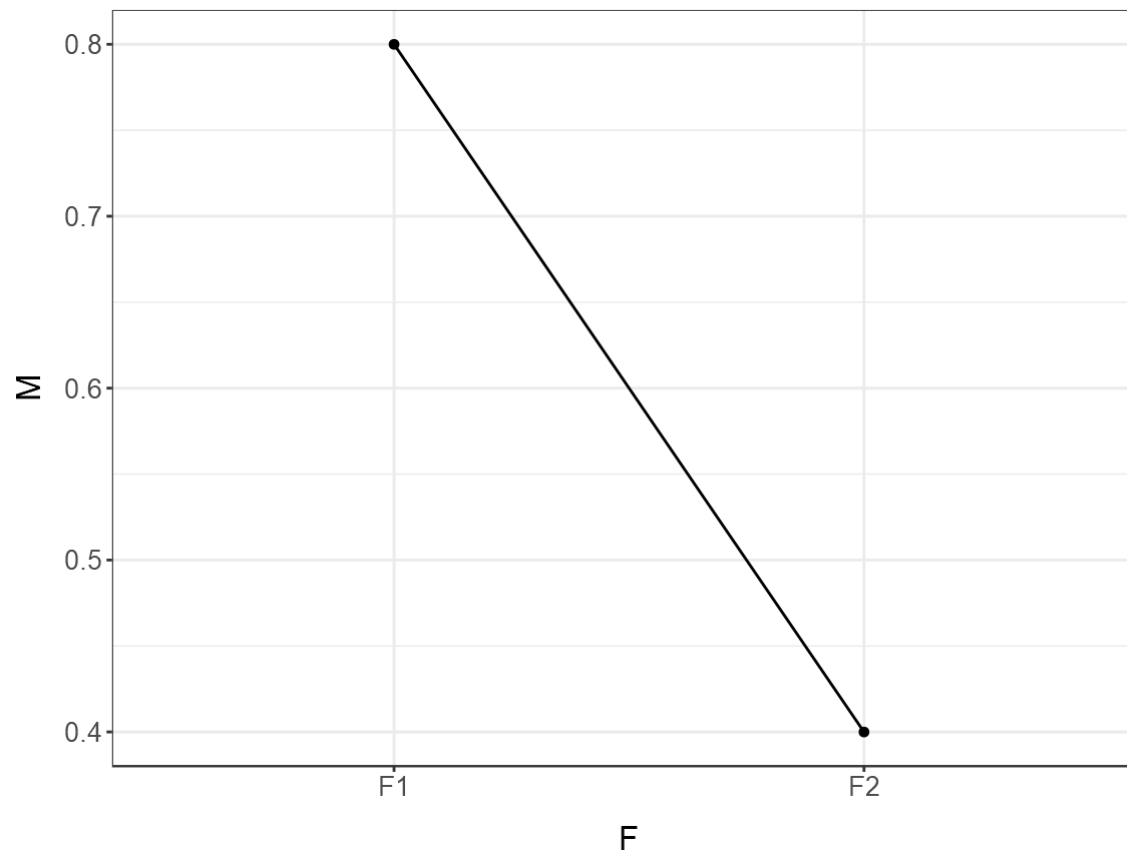


```
simdat |>
  group_by(F) |>
  summarise(
    N = length(DV),
    M = mean(DV),
    SD = sd(DV),
    SE = SD / sqrt(N)
  ) |>
  ungroup()
```

```
# A tibble: 2 x 5
  F         N     M   SD   SE
<fct> <int> <dbl> <dbl> <dbl>
1 F1         5  0.8  0.2 0.0894
2 F2         5  0.4  0.2 0.0894
```

```
simdat |>
  group_by(F) |>
  summarise(
    N = length(DV),
    M = mean(DV),
    SD = sd(DV),
    SE = SD / sqrt(N)
  ) |>
  ungroup() |>
```

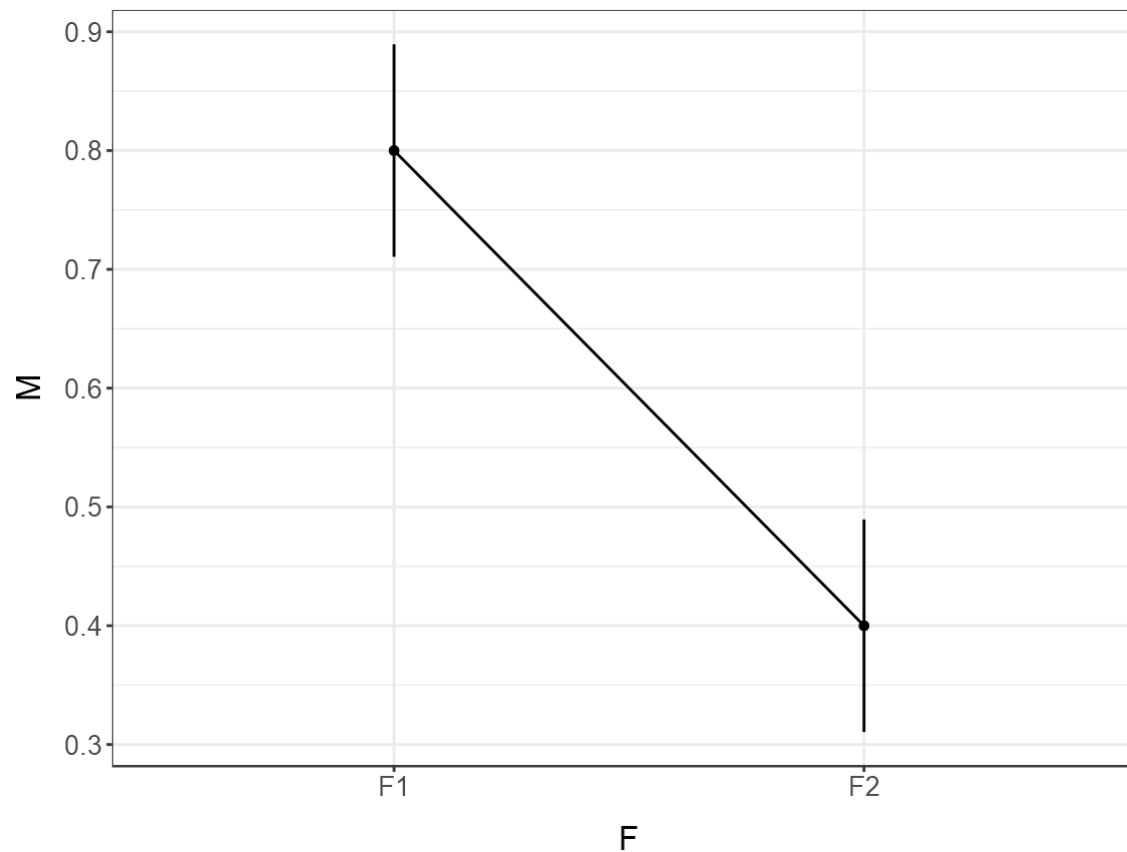
```
(
  \ (d) {
    qplot(
      x = F, y = M,
      group = 1,
      data = d,
      geom = c("point", "line")
    )
  }
) ()
```



```

simdat |>
  group_by(F) |>
  summarise(
    N = length(DV),
    M = mean(DV),
    SD = sd(DV),
    SE = SD / sqrt(N)
  ) |>
  ungroup() |>
  (
    \ (d) {
      qplot(
        x = F, y = M,
        group = 1,
        data = d,
        geom = c("point", "line")
      )
    }
  ) () +
  geom_errorbar(
    aes(
      max = M + SE,
      min = M - SE
    ),
    width = 0
  )

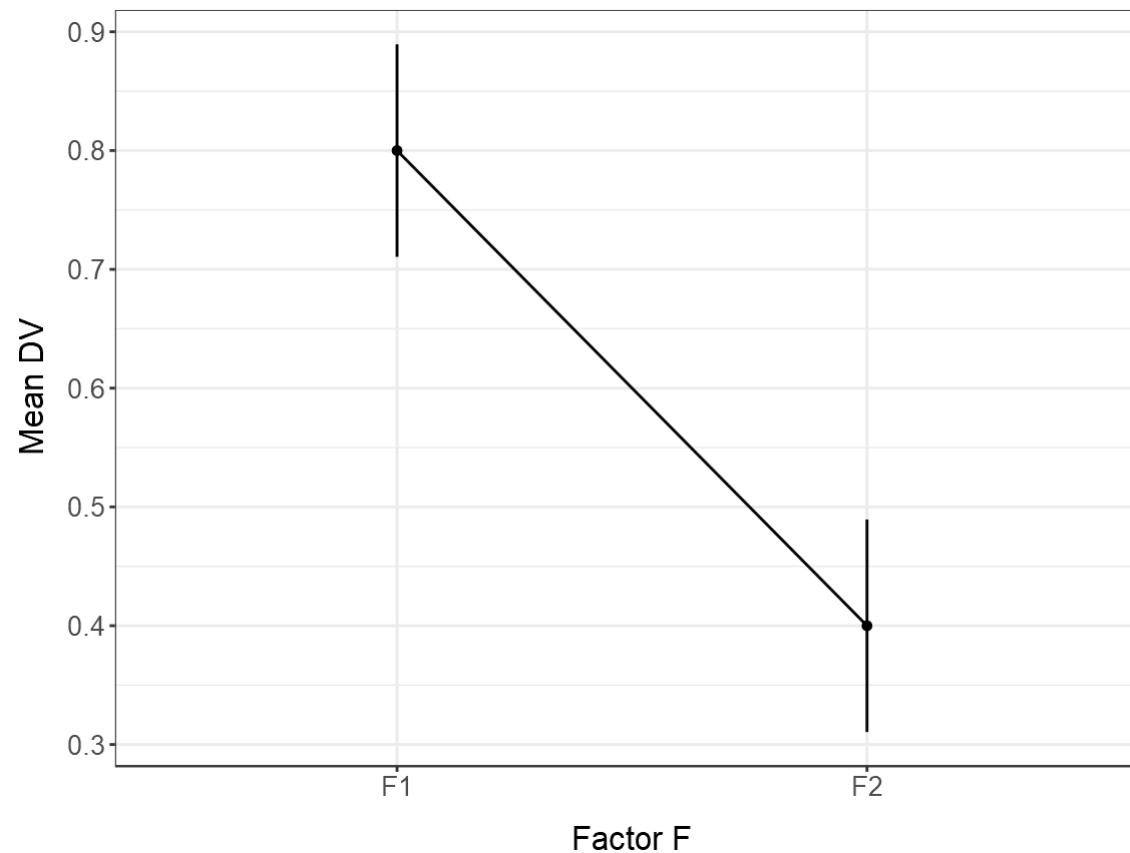
```



```

simdat |>
  group_by(F) |>
  summarise(
    N = length(DV),
    M = mean(DV),
    SD = sd(DV),
    SE = SD / sqrt(N)
  ) |>
  ungroup() |>
  (
    \ (d) {
      qplot(
        x = F, y = M,
        group = 1,
        data = d,
        geom = c("point", "line")
      )
    }
  ) () +
  geom_errorbar(
    aes(
      max = M + SE,
      min = M - SE
    ),
    width = 0
  ) +
  labs(
    y = "Mean DV",
    x = "Factor F"
  )

```



```
simdat
```

	F	id	DV
1	F1	1	0.7370717
2	F1	2	0.5544852
3	F1	3	0.9634010
4	F1	4	1.0436117
5	F1	5	0.7014305
6	F2	6	0.2616614
7	F2	7	0.3948117
8	F2	8	0.1731352
9	F2	9	0.4832820
10	F2	10	0.6871097

```
simdat |>
```

```
group_by(F)
```

```
# A tibble: 10 x 3
```

```
# Groups:   F [2]
```

	F	id	DV
	<fct>	<fct>	<dbl>
1	F1	1	0.737
2	F1	2	0.554
3	F1	3	0.963
4	F1	4	1.04
5	F1	5	0.701
6	F2	6	0.262
7	F2	7	0.395
8	F2	8	0.173
9	F2	9	0.483
10	F2	10	0.687

```
simdat |>
  group_by(F) |>
  summarise(
    N = length(DV),
    M = mean(DV),
    SD = sd(DV),
    SE = SD / sqrt(N)
  )
```

```
# A tibble: 2 x 5
  F         N     M   SD   SE
<fct> <int> <dbl> <dbl> <dbl>
1 F1         5  0.8  0.2 0.0894
2 F2         5  0.4  0.2 0.0894
```

```
simdat |>
  group_by(F) |>
  summarise(
    N = length(DV),
    M = mean(DV),
    SD = sd(DV),
    SE = SD / sqrt(N)
  ) |>
  ungroup()
```

```
# A tibble: 2 x 5
  F         N     M   SD   SE
<fct> <int> <dbl> <dbl> <dbl>
1 F1         5  0.8  0.2 0.0894
2 F2         5  0.4  0.2 0.0894
```



```
simdat |>
  group_by(F) |>
  summarise(
    N = length(DV),
    M = mean(DV),
    SD = sd(DV),
    SE = SD / sqrt(N)
  ) |>
  ungroup() |>
```

```
kable(
  digits = 2,
  col.names = c(
    'Levels of Factor',
    'N. of data points',
    'Mean RT',
    'Std. Dev.',
    'Std. Err.'
  )
)
```

Levels of Factor	N. of data points	Mean RT	Std. Dev.	Std. Err.
F1	5	0.8	0.2	0.09
F2	5	0.4	0.2	0.09

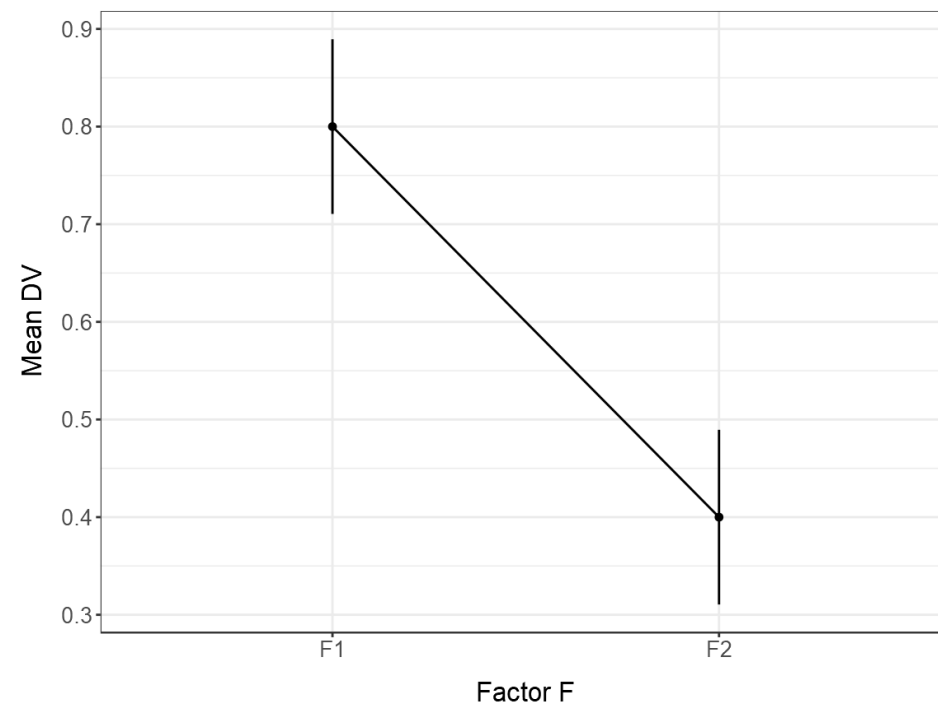
シミュレーション用データに対し線形モデルを構築

- 水準間で応答変数の平均が有意に異なるか、線形モデルで確認

```
m_F <- lm(DV ~ F, simdat)
```

- 切片 (intercept, 0.8)
 - F1の下での応答変数の平均 $\widehat{\mu}_{F1}$
- 傾き (intercept, -0.4)
 - F2の下での応答変数の平均とF1の下での応答変数の平均の差 $\widehat{\mu}_{F2} - \widehat{\mu}_{F1}$

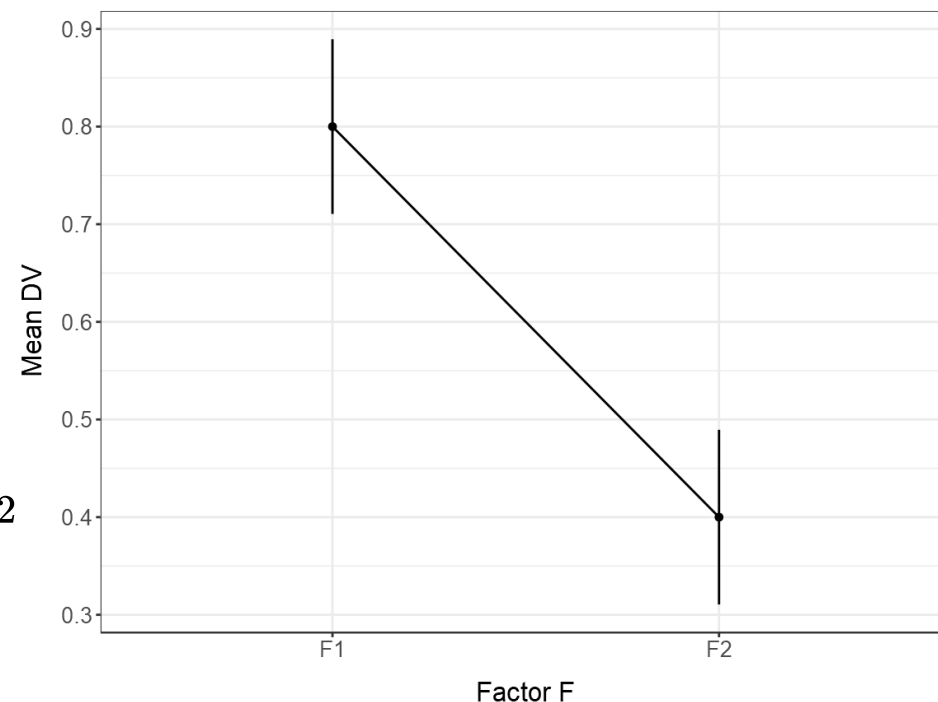
Predictor	Estimate	95% CI	t-value	p-value
(Intercept)	0.8	[0.594, 1.006]	8.94	0.00
FF2	-0.4	[-0.692, -0.108]	-3.16	0.01



シミュレーション用データに 対し線形モデルを構築

- 95%信頼区間 (95% Confidence Interval)
 - 平均値 $\pm 1.96 \times$ 標準誤差 (Standard Error)
 - 「100回実験して95回は真値を含むような区間」
- t 値
 - この t 値は, 自由度が
((その水準から得られたデータポイント数) $- 1$) $\times 2$
の t 分布に従う
 - この t 分布の両側2.5%の範囲にあれば, 有意
 - $(5 - 1) \times 2 = 8$ なので, 今回の t 値は, $t(8)$
に従う
 - $t(8)$ の両側2.5%の範囲は, $t \leq 2.31, 2.31 \leq t$
 - t 値が上記区間に入っていれば, 有意

Predictor	Estimate	95% CI	t-value	p-value
(Intercept)	0.8	[0.594, 1.006]	8.94	0.00
FF2	-0.4	[-0.692, -0.108]	-3.16	0.01



m_F

Call:

```
lm(formula = DV ~ F, data = simdat)
```

Coefficients:

(Intercept)	FF2
0.8	-0.4

```
m_F |>
  tidy(
    conf.int = TRUE,
    conf.level = 0.95
  )
```

```
# A tibble: 2 x 7
  term          estimate std.error statistic  p.value conf.low conf.high
<chr>         <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1 (Intercept)     0.8    0.0894     8.94 0.0000194  0.594    1.01
2 FF2            -0.4    0.126    -3.16 0.0133    -0.692   -0.108
```

```
m_F |>
  tidy(
    conf.int = TRUE,
    conf.level = 0.95
  ) |>
```

```
mutate(
  `95% CI` = paste0(
    "[",
    conf.low |> round(3),
    ", ",
    conf.high |> round(3),
    "]"
  )
)
```

```
.panel2-tab-lm-simdat-auto[
```

```
# A tibble: 2 x 8
```

	term	estimate	std.error	statistic	p.value	conf.low	conf.high
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	(Intercept)	0.8	0.0894	8.94	0.0000194	0.594	1.006
2	FF2	-0.4	0.126	-3.16	0.0133	-0.692	-0.108

```
]
```

```

m_F |>
  tidy(
    conf.int = TRUE,
    conf.level = 0.95
  ) |>
  mutate(
    `95% CI` = paste0(
      "[",
      conf.low |> round(3),
      ", ",
      conf.high |> round(3),
      "]"
    )
  ) |>
  relocate(
    `95% CI`, .after = estimate
  )

```

.panel2-tab-lm-simdat-auto[

```

# A tibble: 2 x 8
  term          estimate `95% CI`   std.error statistic p.value conf
  <chr>          <dbl> <chr>         <dbl>      <dbl>   <dbl>   <dbl>
1 (Intercept)      0.8 [0.594, 1~    0.0894      8.94 1.94e-5    0
2 FF2             -0.4 [-0.692, ~    0.126     -3.16 1.33e-2   -0

```

]

```

m_F |>
  tidy(
    conf.int = TRUE,
    conf.level = 0.95
  ) |>
  mutate(
    `95% CI` = paste0(
      "[",
      conf.low |> round(3),
      ", ",
      conf.high |> round(3),
      "]"
    )
  ) |>
  relocate(
    `95% CI`, .after = estimate
  ) |>
  dplyr::select(
    -c(
      std.error,
      conf.low,
      conf.high
    )
  )

```

```

# A tibble: 2 x 5
  term          estimate `95% CI`      statistic    p.value
  <chr>          <dbl> <chr>          <dbl>      <dbl>
1 (Intercept)      0.8 [0.594, 1.006]      8.94 0.0000194
2 FF2             -0.4 [-0.692, -0.108]     -3.16 0.0133

```



```

m_F |>
  tidy(
    conf.int = TRUE,
    conf.level = 0.95
  ) |>
  mutate(
    `95% CI` = paste0(
      "[",
      conf.low |> round(3),
      ", ",
      conf.high |> round(3),
      "]"
    )
  ) |>
  relocate(
    `95% CI`, .after = estimate
  ) |>
  dplyr::select(
    -c(
      std.error,
      conf.low,
      conf.high
    )
  ) |>

```

```

kable(
  digits = 2,
  escape = FALSE,
  col.names = c(
    'Predictor',
    'Estimate',
    '95% CI',
    '$t$-value',
    '$p$-value'
  )
)

```

Predictor	Estimate	95% CI	t-value	p-value
(Intercept)	0.8	[0.594, 1.006]	8.94	0.00
FF2	-0.4	[-0.692, -0.108]	-3.16	0.01

Default contrast coding: treatment contrasts

- 切片 (intercept, 0.8)
 - F1の下での応答変数の平均 $\widehat{\mu}_{F1}$
- 傾き (intercept, -0.4)
 - F2の下での応答変数の平均とF1の下での応答変数の平均の差 $\widehat{\mu}_{F2} - \widehat{\mu}_{F1}$

なぜ, 切片と傾きがこの値になるのか??

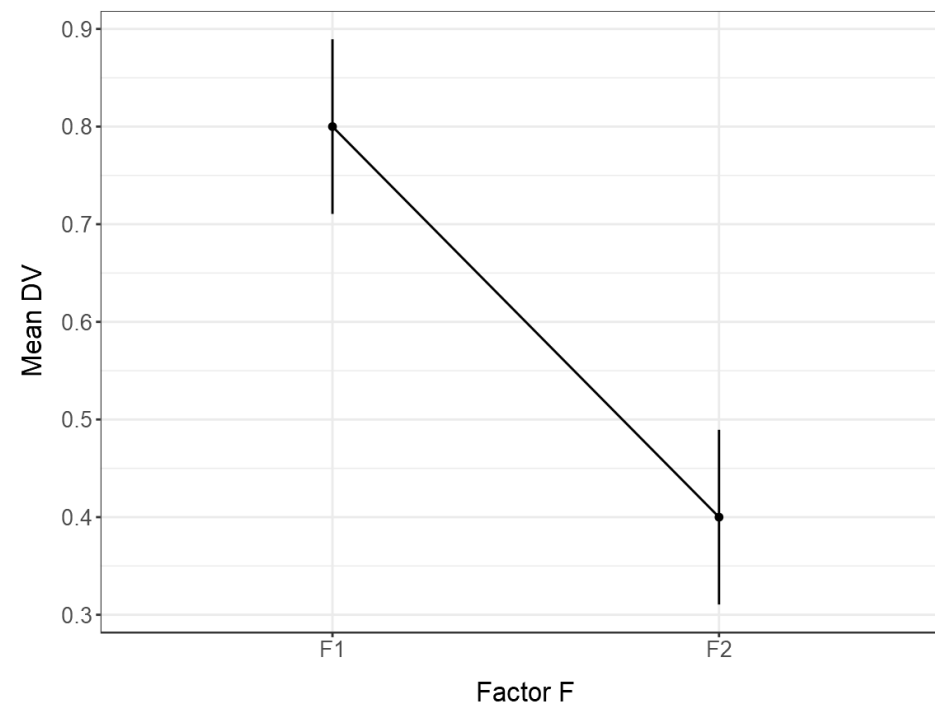
Default contrast coding: treatment contrasts

- 切片 (intercept, 0.8)
 - F1の下での応答変数の平均 $\widehat{\mu}_{F1}$
- 傾き (intercept, -0.4)
 - F2の下での応答変数の平均とF1の下での応答変数の平均の差 $\widehat{\mu}_{F2} - \widehat{\mu}_{F1}$

なぜ、切片と傾きがこの値になるのか？

∴ デフォルトでは、Rは、名義尺度に **treatment contrast** を適用するため

```
m_F <- lm(DV ~ F, simdat)
```



Treatment contrastsとその意味

```
contrasts(simdat$F)
```

```
      F2  
F1      0  
F2      1
```

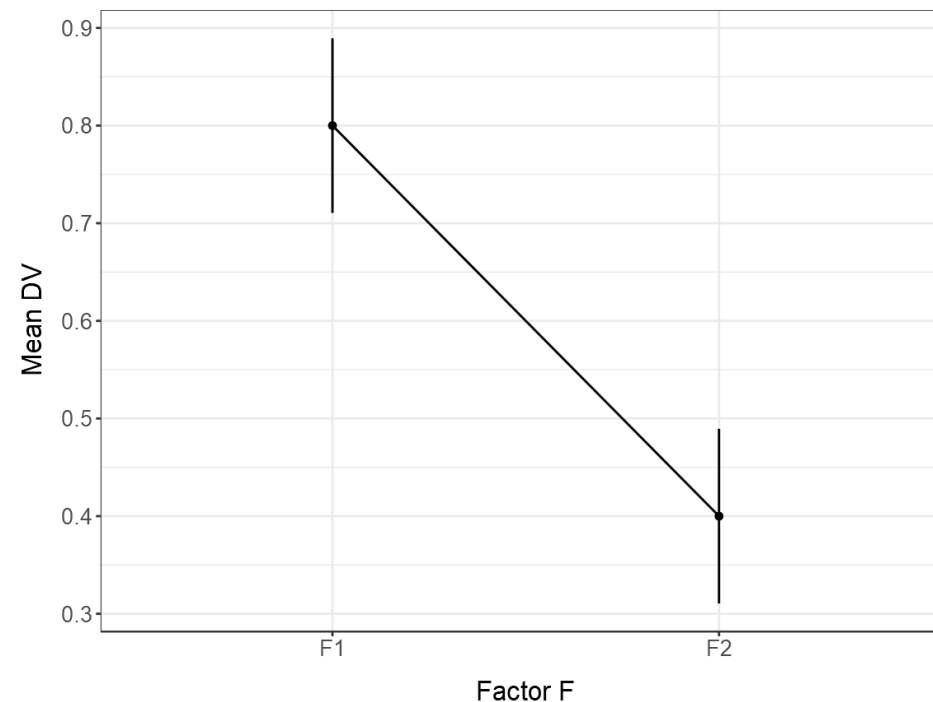
$$x = \begin{cases} 0 & \text{F1 のとき} \\ 1 & \text{F2 のとき} \end{cases}$$

Default contrast coding: treatment contrasts

Rがデフォルトで名義尺度に適用する対比

$$x = \begin{cases} 0 & \text{F1 のとき} \\ 1 & \text{F2 のとき} \end{cases}$$

$$\begin{aligned} \underbrace{\text{応答変数}}_y &= \underbrace{\text{切片}}_{\beta_0} + \underbrace{\text{傾き}}_{\beta_1 x} \\ &= \begin{cases} \beta_0 + \beta_1 \times 0 & \text{F1 のとき} \\ \beta_0 + \beta_1 \times 1 & \text{F2 のとき} \end{cases} \\ \therefore &= \begin{cases} \beta_0 & \text{F1 のとき} \\ \beta_0 + \beta_1 & \text{F2 のとき} \end{cases} \\ &= \begin{cases} 0.8 & \text{F1 のとき} \\ 0.8 + (-0.4) & \text{F2 のとき} \end{cases} \\ &= \begin{cases} 0.8 & \text{F1 のとき} \\ 0.4 & \text{F2 のとき} \end{cases} \end{aligned}$$



Defining hypotheses

Treatment contrastの下では,

- 傾き β_1
 - 「実験条件の応答変数の平均値」と「統制条件での応答変数の平均値」の差
 - $(\mu_{F2} - \mu_{F1})$
- 切片 β_0
 - 統制条件での応答変数の平均値
 - (μ_{F1})

Defining hypotheses

Treatment contrastの下では,

- 傾き β_1
 - 「実験条件の応答変数の平均値」と「統制条件での応答変数の平均値」の差
 - $(\mu_{F2} - \mu_{F1})$
- 切片 β_0
 - 統制条件での応答変数の平均値
 - (μ_{F1})

↑切片や傾きの意味を言語化しただけ

Defining hypotheses

Treatment contrastの下では,

- 傾き β_1
 - 「実験条件の応答変数の平均値」と「統制条件での応答変数の平均値」の差
 - $(\mu_{F2} - \mu_{F1})$
- 切片 β_0
 - 統制条件での応答変数の平均値
 - (μ_{F1})

↑切片や傾きの意味を言語化しただけ

切片や傾きを通じて、どのような帰無仮説を棄却しようとしているのか？

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

$$H_0 : \mu_{F2} - \mu_{F1} = 0$$

切片 β_0 に関する帰無仮説 H_0

帰無仮説：切片が存在しない

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

$$H_0 : \mu_{F2} - \mu_{F1} = 0$$

切片 β_0 に関する帰無仮説 H_0

帰無仮説：切片が存在しない

$$H_0 : \beta_0 = 0$$

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

$$H_0 : \mu_{F2} - \mu_{F1} = 0$$

切片 β_0 に関する帰無仮説 H_0

帰無仮説：切片が存在しない

$$H_0 : \beta_0 = 0$$

$$H_0 : \mu_{F1} = 0$$

統制条件とする水準を変えるには

- Rは, 水準をアルファベット順に読み込み, アルファベット順に数値 (0,1) を付与する
- アルファベット順に関係なく, 特定の水準を統制条件としたい場合には

統制条件とする水準を変えるには

```
simdat <- simdat
```

```
simdat
```

	F	id	DV
1	F1	1	0.7370717
2	F1	2	0.5544852
3	F1	3	0.9634010
4	F1	4	1.0436117
5	F1	5	0.7014305
6	F2	6	0.2616614
7	F2	7	0.3948117
8	F2	8	0.1731352
9	F2	9	0.4832820
10	F2	10	0.6871097

統制条件とする水準を変えるには

```
simdat <- simdat |>
```

```
  mutate(  
    Fb = fct_relevel(  
      F,  
      "F2",  
      "F1"  
    )  
  )
```

```
simdat
```

	F	id	DV	Fb
1	F1	1	0.7370717	F1
2	F1	2	0.5544852	F1
3	F1	3	0.9634010	F1
4	F1	4	1.0436117	F1
5	F1	5	0.7014305	F1
6	F2	6	0.2616614	F2
7	F2	7	0.3948117	F2
8	F2	8	0.1731352	F2
9	F2	9	0.4832820	F2
10	F2	10	0.6871097	F2

統制条件とする水準を変えるには

```
simdat <- simdat |>
  mutate(
    Fb = fct_relevel(
      F,
      "F2",
      "F1"
    )
  )
```

```
contrasts(simdat$Fb)
```

```
simdat
```

```
F1
F2  0
F1  1
```

	F	id	DV	Fb
1	F1	1	0.7370717	F1
2	F1	2	0.5544852	F1
3	F1	3	0.9634010	F1
4	F1	4	1.0436117	F1
5	F1	5	0.7014305	F1
6	F2	6	0.2616614	F2
7	F2	7	0.3948117	F2
8	F2	8	0.1731352	F2
9	F2	9	0.4832820	F2
10	F2	10	0.6871097	F2

統制条件とする水準を変えるには

```
simdat <- simdat |>
  mutate(
    Fb = fct_relevel(
      F,
      "F2",
      "F1"
    )
  )
```

```
contrasts(simdat$Fb)
```

```
contrasts(simdat$F)
```

```
simdat
```

```
F1
F2 0
F1 1
```

```
F2
F1 0
F2 1
```

	F	id	DV	Fb
1	F1	1	0.7370717	F1
2	F1	2	0.5544852	F1
3	F1	3	0.9634010	F1
4	F1	4	1.0436117	F1
5	F1	5	0.7014305	F1
6	F2	6	0.2616614	F2
7	F2	7	0.3948117	F2
8	F2	8	0.1731352	F2
9	F2	9	0.4832820	F2
10	F2	10	0.6871097	F2

F2を統制条件にした場合

```
m1_mr <- lm(DV ~ Fb, simdat)
```

Predictor	Estimate	95% CI	t-value	p-value
(Intercept)	0.4	[0.194, 0.606]	4.47	0.00
FbF1	0.4	[0.108, 0.692]	3.16	0.01

$$x = \begin{cases} 0 & \text{F2 のとき} \\ 1 & \text{F1 のとき} \end{cases}$$

$$\begin{aligned} y &= \beta_0 + \beta_1 x \\ &= \begin{cases} \beta_0 + \beta_1 \times 0 & \text{F2 のとき} \\ \beta_0 + \beta_1 \times 1 & \text{F1 のとき} \end{cases} \\ \therefore &= \begin{cases} 0.4 & \text{F2 のとき} \\ 0.4 + (0.4) & \text{F1 のとき} \end{cases} \\ &= \begin{cases} 0.4 & \text{F2 のとき} \\ 0.8 & \text{F1 のとき} \end{cases} \end{aligned}$$

F1が統制条件である場合

```
m_F <- lm(DV ~ F, simdat)
```

Predictor	Estimate	95% CI	t-value	p-value
(Intercept)	0.8	[0.594, 1.006]	8.94	0.00
FF2	-0.4	[-0.692, -0.108]	-3.16	0.01

$$x = \begin{cases} 0 & \text{F1 のとき} \\ 1 & \text{F2 のとき} \end{cases}$$

$$\begin{aligned} y &= \beta_0 + \beta_1 x \\ &= \begin{cases} \beta_0 + \beta_1 \times 0 & \text{F1 のとき} \\ \beta_0 + \beta_1 \times 1 & \text{F2 のとき} \end{cases} \\ \therefore &= \begin{cases} 0.8 & \text{F1 のとき} \\ 0.8 + (-0.4) & \text{F2 のとき} \end{cases} \\ &= \begin{cases} 0.8 & \text{F1 のとき} \\ 0.4 & \text{F2 のとき} \end{cases} \end{aligned}$$

F2を統制条件にした場合

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

F2を統制条件にした場合

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

F2を統制条件にした場合

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

$$H_0 : \mu_{F1} - \mu_{F2} = 0$$

切片 β_0 に関する帰無仮説 H_0

帰無仮説：切片が存在しない

F2を統制条件にした場合

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

$$H_0 : \mu_{F1} - \mu_{F2} = 0$$

切片 β_0 に関する帰無仮説 H_0

帰無仮説：切片が存在しない

$$H_0 : \beta_0 = 0$$

F2を統制条件にした場合

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

$$H_0 : \mu_{F1} - \mu_{F2} = 0$$

切片 β_0 に関する帰無仮説 H_0

帰無仮説：切片が存在しない

$$H_0 : \beta_0 = 0$$

$$H_0 : \mu_{F2} = 0$$

(参考) F1が統制条件である場合

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

$$H_0 : \mu_{F2} - \mu_{F1} = 0$$

切片 β_0 に関する帰無仮説 H_0

帰無仮説：切片が存在しない

$$H_0 : \beta_0 = 0$$

$$H_0 : \mu_{F1} = 0$$

Sum contrast

- (2水準の場合) 一方の水準に-1を, 他方の水準に1を充てる
- 効果全体を全体平均に「中心化」する
 - 全体平均 (Grand Mean): 各水準の平均を平均した値
- scaled sum contrasts; effect coding
 - (2水準の場合) 一方の水準に-0.5を, 他方の水準に0.5を充てる
 - 傾きが, 2水準の差と同じ値になる

Scaled sum contrastsとその意味

```
contrasts(simdat$F) <- c(-0.5, 0.5)
```

```
contrasts(simdat$F)
```

```
 [,1]  
F1 -0.5  
F2  0.5
```

$$x = \begin{cases} -0.5 & \text{F1 のとき} \\ 0.5 & \text{F2 のとき} \end{cases}$$

Sum contrast

- (2水準の場合) 一方の水準に-1を, 他方の水準に1を充てる
- 効果全体を全体平均に「中心化」する
 - 全体平均 (Grand Mean): 各水準の平均を平均した値
- scaled sum contrasts; effect coding
 - (2水準の場合) 一方の水準に-0.5を, 他方の水準に0.5を充てる
 - 傾きが, 2水準の差と同じ値になる

```
m1_mr <- lm(DV ~ F, simdat)
```

Predictor	Estimate	95% CI	t-value	p-value
(Intercept)	0.6	[0.454, 0.746]	9.49	0.00
F1	-0.4	[-0.692, -0.108]	-3.16	0.01

(Scaled) Sum contrasts

$$x = \begin{cases} -0.5 & \text{F1 のとき} \\ 0.5 & \text{F2 のとき} \end{cases}$$

$$\underbrace{\text{応答変数}}_y = \underbrace{\text{切片}}_{\beta_0} + \underbrace{\text{傾き}}_{\beta_1} x$$

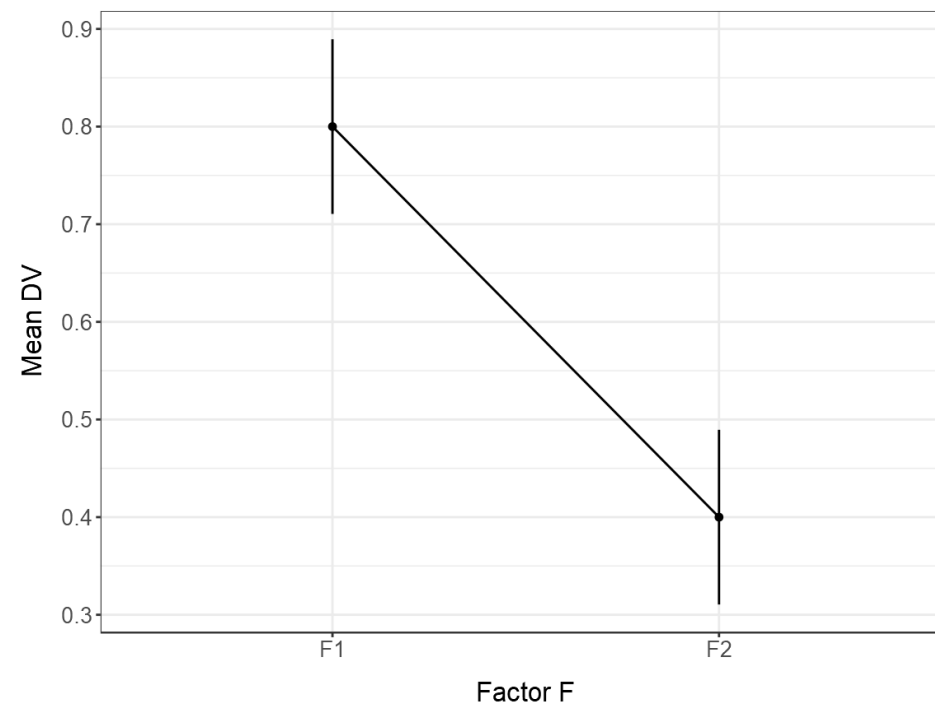
$$= \begin{cases} \beta_0 + \beta_1 \times (-0.5) & \text{F1 のとき} \\ \beta_0 + \beta_1 \times 0.5 & \text{F2 のとき} \end{cases}$$

$$\therefore = \begin{cases} \beta_0 - 0.5\beta_1 & \text{F1 のとき} \\ \beta_0 + 0.5\beta_1 & \text{F2 のとき} \end{cases}$$

$$= \begin{cases} 0.6 - 0.5 \times (-0.4) & \text{F1 のとき} \\ 0.6 + 0.5 \times (-0.4) & \text{F2 のとき} \end{cases}$$

$$= \begin{cases} 0.8 & \text{F1 のとき} \\ 0.4 & \text{F2 のとき} \end{cases}$$

Predictor	Estimate	95% CI	t-value	p-value
(Intercept)	0.6	[0.454, 0.746]	9.49	0.00
F1	-0.4	[-0.692, -0.108]	-3.16	0.01



(Scaled) Sum contrasts

(Scaled) Sum contrastsの下では,

- 傾き β_1
 - 「一方の条件の応答変数の平均値」と「他方の条件での応答変数の平均値」の差
 - $(\mu_{F2} - \mu_{F1})$
- 切片 β_0
 - 「一方の条件の応答変数の平均値」と「他方の条件での応答変数の平均値」の平均値
 - $(\frac{\mu_{F1} + \mu_{F2}}{2})$

(Scaled) Sum contrastsの帰無仮説

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

(Scaled) Sum contrastsの帰無仮説

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

(Scaled) Sum contrastsの帰無仮説

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

$$H_0 : \mu_{F2} - \mu_{F1} = 0$$

切片 β_0 に関する帰無仮説 H_0

帰無仮説：切片が存在しない

(Scaled) Sum contrastsの帰無仮説

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

$$H_0 : \mu_{F2} - \mu_{F1} = 0$$

切片 β_0 に関する帰無仮説 H_0

帰無仮説：切片が存在しない

$$H_0 : \beta_0 = 0$$

(Scaled) Sum contrastsの帰無仮説

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

$$H_0 : \mu_{F2} - \mu_{F1} = 0$$

切片 β_0 に関する帰無仮説 H_0

帰無仮説：切片が存在しない

$$H_0 : \beta_0 = 0$$

$$H_0 : \frac{\mu_{F1} + \mu_{F2}}{2} = 0$$

(参考) F1を統制条件とする Treatment contrastの場合

傾き β_1 に関する帰無仮説 H_0

帰無仮説：傾きが存在しない

$$H_0 : \beta_1 = 0$$

$$H_0 : \mu_{F2} - \mu_{F1} = 0$$

切片 β_0 に関する帰無仮説 H_0

帰無仮説：切片が存在しない

$$H_0 : \beta_0 = 0$$

$$H_0 : \mu_{F1} = 0$$

(Un)balanced dataでのGrand Meanの意味

```
simdat
```

	F	id	DV	Fb
1	F1	1	0.7370717	F1
2	F1	2	0.5544852	F1
3	F1	3	0.9634010	F1
4	F1	4	1.0436117	F1
5	F1	5	0.7014305	F1
6	F2	6	0.2616614	F2
7	F2	7	0.3948117	F2
8	F2	8	0.1731352	F2
9	F2	9	0.4832820	F2
10	F2	10	0.6871097	F2

データセット

(Un)balanced dataでのGrand Meanの意味

```
simdat |>
```

```
  mutate(  
    gross.mean = mean(DV)  
  )
```

	F	id	DV	Fb	gross.mean
1	F1	1	0.7370717	F1	0.6
2	F1	2	0.5544852	F1	0.6
3	F1	3	0.9634010	F1	0.6
4	F1	4	1.0436117	F1	0.6
5	F1	5	0.7014305	F1	0.6
6	F2	6	0.2616614	F2	0.6
7	F2	7	0.3948117	F2	0.6
8	F2	8	0.1731352	F2	0.6
9	F2	9	0.4832820	F2	0.6
10	F2	10	0.6871097	F2	0.6

応答変数全ての平均を計算

(Un)balanced dataでのGrand Meanの意味

```
simdat |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F)
```

```
# A tibble: 10 x 5
# Groups:   F [2]
   F      id      DV Fb    gross.mean
  <fct> <fct> <dbl> <fct>    <dbl>
1 F1     1    0.737 F1      0.6
2 F1     2    0.554 F1      0.6
3 F1     3    0.963 F1      0.6
4 F1     4    1.04  F1      0.6
5 F1     5    0.701 F1      0.6
6 F2     6    0.262 F2      0.6
7 F2     7    0.395 F2      0.6
8 F2     8    0.173 F2      0.6
9 F2     9    0.483 F2      0.6
10 F2    10    0.687 F2      0.6
```

説明変数の水準毎にデータをまとめ上げ

(Un)balanced dataでのGrand Meanの意味

```
simdat |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  )
```

```
# A tibble: 10 x 6
# Groups:   F [2]
   F      id      DV Fb      gross.mean lev.wise.mean
  <fct> <fct> <dbl> <fct>      <dbl>      <dbl>
1 F1     1      0.737 F1          0.6          0.8
2 F1     2      0.554 F1          0.6          0.8
3 F1     3      0.963 F1          0.6          0.8
4 F1     4      1.04  F1          0.6          0.8
5 F1     5      0.701 F1          0.6          0.8
6 F2     6      0.262 F2          0.6          0.4
7 F2     7      0.395 F2          0.6          0.4
8 F2     8      0.173 F2          0.6          0.4
9 F2     9      0.483 F2          0.6          0.4
10 F2    10      0.687 F2          0.6          0.4
```

水準毎に応答変数の平均値を計算

(Un)balanced dataでのGrand Meanの意味

```
simdat |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup()
```

```
# A tibble: 10 x 6
      F      id      DV Fb      gross.mean lev.wise.mean
  <fct> <fct> <dbl> <fct>      <dbl>      <dbl>
1 F1     1      0.737 F1          0.6          0.8
2 F1     2      0.554 F1          0.6          0.8
3 F1     3      0.963 F1          0.6          0.8
4 F1     4      1.04  F1          0.6          0.8
5 F1     5      0.701 F1          0.6          0.8
6 F2     6      0.262 F2          0.6          0.4
7 F2     7      0.395 F2          0.6          0.4
8 F2     8      0.173 F2          0.6          0.4
9 F2     9      0.483 F2          0.6          0.4
10 F2    10      0.687 F2          0.6          0.4
```

まとめ上げを解除

(Un)balanced dataでのGrand Meanの意味

```
simdat |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup() |>
  distinct(
    F, gross.mean, lev.wise.mean
  )
```

```
# A tibble: 2 x 3
  F      gross.mean lev.wise.mean
<fct>      <dbl>      <dbl>
1 F1          0.6          0.8
2 F2          0.6          0.4
```

応答変数全ての平均と水準毎の平均を抽出

(Un)balanced dataでのGrand Meanの意味

```
simdat |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup() |>
  distinct(
    F, gross.mean, lev.wise.mean
  ) |>
  summarise(
    gross.mean = gross.mean,
    mean.of.lev.wise.mean = mean(lev.wi
  )
```

```
# A tibble: 2 x 2
  gross.mean mean.of.lev.wise.mean
    <dbl>         <dbl>
1     0.6         0.6
2     0.6         0.6
```

水準毎の平均の平均を計算

(Un)balanced dataでのGrand Meanの意味

```
simdat |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup() |>
  distinct(
    F, gross.mean, lev.wise.mean
  ) |>
  summarise(
    gross.mean = gross.mean,
    mean.of.lev.wise.mean = mean(lev.wi
  ) |>
  distinct(
    gross.mean, mean.of.lev.wise.mean
  )
```

```
# A tibble: 1 x 2
  gross.mean mean.of.lev.wise.mean
    <dbl>         <dbl>
1     0.6           0.6
```

Grand Mean : 「各水準で得られた応答変数の平均値」の平均値

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)
```

乱数固定

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)
```

```
simdat
```

	F	id	DV	Fb
1	F1	1	0.7370717	F1
2	F1	2	0.5544852	F1
3	F1	3	0.9634010	F1
4	F1	4	1.0436117	F1
5	F1	5	0.7014305	F1
6	F2	6	0.2616614	F2
7	F2	7	0.3948117	F2
8	F2	8	0.1731352	F2
9	F2	9	0.4832820	F2
10	F2	10	0.6871097	F2

データセット

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)
```

```
simdat |>
```

```
group_split(F)
```

```
<list_of<
  tbl_df<
    F : factor<1ffb8>
    id: factor<49edf>
    DV: double
    Fb: factor<e285c>
  >
>[2]>
[[1]]
# A tibble: 5 x 4
  F      id      DV Fb
  <fct> <fct> <dbl> <fct>
1 F1     1      0.737 F1
2 F1     2      0.554 F1
3 F1     3      0.963 F1
4 F1     4      1.04  F1
5 F1     5      0.701 F1

[[2]]
# A tibble: 5 x 4
  F      id      DV Fb
  <fct> <fct> <dbl> <fct>
1 F2     6      0.262 F2
2 F2     7      0.395 F2
3 F2     8      0.173 F2
4 F2     9      0.483 F2
5 F2    10      0.687 F2
```

データを水準ごとに分割

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1から2つ, F2から1つデータを取り出す ####
    .y = c(2, 1),
    ~ slice_sample(.x, n = .y)
  )
```

```
# A tibble: 3 x 4
  F      id      DV Fb
<fct> <fct> <dbl> <fct>
1 F1     1      0.737 F1
2 F1     4      1.04  F1
3 F2     6      0.262 F2
```

水準毎のデータから、指定した分だけ無作為抽出（F1とF2でデータ数が異なるように抽出）

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1から2つ, F2から1つデータを取り出す ####
    .y = c(2, 1),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  )
```

```
# A tibble: 3 x 5
  F      id      DV Fb      gross.mean
<fct> <fct> <dbl> <fct>      <dbl>
1 F1     1      0.737 F1          0.681
2 F1     4      1.04  F1          0.681
3 F2     6      0.262 F2          0.681
```

無作為抽出後のデータについて、応答変数全ての平均を計算

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1から2つ, F2から1つデータを取り出す ####
    .y = c(2, 1),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F)
```

```
# A tibble: 3 x 5
# Groups:   F [2]
  F      id      DV Fb    gross.mean
<fct> <fct> <dbl> <fct>    <dbl>
1 F1     1    0.737 F1      0.681
2 F1     4    1.04  F1      0.681
3 F2     6    0.262 F2      0.681
```

説明変数の水準毎にデータをまとめ上げ

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1から2つ, F2から1つデータを取り出す ####
    .y = c(2, 1),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  )
```

```
# A tibble: 3 x 6
# Groups:   F [2]
  F      id      DV Fb    gross.mean lev.wise.mean
<fct> <fct> <dbl> <fct>    <dbl>         <dbl>
1 F1     1    0.737 F1      0.681         0.890
2 F1     4    1.04  F1      0.681         0.890
3 F2     6    0.262 F2      0.681         0.262
```

水準毎に応答変数の平均値を計算

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1から2つ, F2から1つデータを取り出す ####
    .y = c(2, 1),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup()
```

```
# A tibble: 3 x 6
  F      id      DV Fb      gross.mean lev.wise.mean
<fct> <fct> <dbl> <fct>      <dbl>      <dbl>
1 F1     1      0.737 F1          0.681      0.890
2 F1     4      1.04  F1          0.681      0.890
3 F2     6      0.262 F2          0.681      0.262
```

まとめ上げを解除

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1から2つ, F2から1つデータを取り出す ####
    .y = c(2, 1),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup() |>
  distinct(
    F, gross.mean, lev.wise.mean
  )
```

```
# A tibble: 2 x 3
  F      gross.mean lev.wise.mean
<fct>      <dbl>      <dbl>
1 F1        0.681        0.890
2 F2        0.681        0.262
```

応答変数全ての平均と水準毎の平均を抽出

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1から2つ, F2から1つデータを取り出す ####
    .y = c(2, 1),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup() |>
  distinct(
    F, gross.mean, lev.wise.mean
  ) |>
  summarise(
    gross.mean = gross.mean,
    mean.of.lev.wise.mean = mean(lev.wi
  )
```

```
# A tibble: 2 x 2
  gross.mean mean.of.lev.wise.mean
      <dbl>         <dbl>
1     0.681         0.576
2     0.681         0.576
```

水準毎の平均の平均を計算

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1から2つ, F2から1つデータを取り出す ####
    .y = c(2, 1),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup() |>
  distinct(
    F, gross.mean, lev.wise.mean
  ) |>
  summarise(
    gross.mean = gross.mean,
    mean.of.lev.wise.mean = mean(lev.wise.mean)
  ) |>
  distinct(
    gross.mean, mean.of.lev.wise.mean
  )
```

```
# A tibble: 1 x 2
  gross.mean mean.of.lev.wise.mean
      <dbl>         <dbl>
1    0.681         0.576
```

「各水準で得られた応答変数の平均値」
の平均値と、全データの平均が**一致しない**

Unbalanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1から2つ, F2から1つデータを取り出す ####
    .y = c(2, 1),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup() |>
  distinct(
    F, gross.mean, lev.wise.mean
  ) |>
  summarise(
    gross.mean = gross.mean,
    mean.of.lev.wise.mean = mean(lev.wise.mean)
  ) |>
  distinct(
    gross.mean, mean.of.lev.wise.mean
  ) |>
```

応答変数全ての平均	「各水準の平均」の平均
0.68	0.58

「各水準で得られた応答変数の平均値」の平均値と、全データの平均が**一致しない**

Balanced dataでのGrand Meanの意味

```
set.seed(1)
```

乱数固定

Balanced dataでのGrand Meanの意味

```
set.seed(1)
```

```
simdat
```

	F	id	DV	Fb
1	F1	1	0.7370717	F1
2	F1	2	0.5544852	F1
3	F1	3	0.9634010	F1
4	F1	4	1.0436117	F1
5	F1	5	0.7014305	F1
6	F2	6	0.2616614	F2
7	F2	7	0.3948117	F2
8	F2	8	0.1731352	F2
9	F2	9	0.4832820	F2
10	F2	10	0.6871097	F2

データセット

Balanced dataでのGrand Meanの意味

```
set.seed(1)
```

```
simdat |>
```

```
group_split(F)
```

```
<list_of<
  tbl_df<
    F : factor<1ffb8>
    id: factor<49edf>
    DV: double
    Fb: factor<e285c>
  >
>[2]>
[[1]]
# A tibble: 5 x 4
  F      id      DV Fb
  <fct> <fct> <dbl> <fct>
1 F1     1      0.737 F1
2 F1     2      0.554 F1
3 F1     3      0.963 F1
4 F1     4      1.04  F1
5 F1     5      0.701 F1

[[2]]
# A tibble: 5 x 4
  F      id      DV Fb
  <fct> <fct> <dbl> <fct>
1 F2     6      0.262 F2
2 F2     7      0.395 F2
3 F2     8      0.173 F2
4 F2     9      0.483 F2
5 F2    10      0.687 F2
```

データを水準ごとに分割

Balanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1・F2ともに、2つずつデータを取り出す #
    .y = c(2, 2),
    ~ slice_sample(.x, n = .y)
  )
```

```
# A tibble: 4 x 4
  F      id      DV Fb
<fct> <fct> <dbl> <fct>
1 F1     1      0.737 F1
2 F1     4      1.04  F1
3 F2     6      0.262 F2
4 F2     7      0.395 F2
```

水準毎のデータから、指定した分だけ無作為抽出（F1とF2でデータ数が一致するように抽出）

Balanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1・F2ともに、2つずつデータを取り出す #
    .y = c(2, 2),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  )
```

```
# A tibble: 4 x 5
  F      id      DV Fb      gross.mean
<fct> <fct> <dbl> <fct>      <dbl>
1 F1     1      0.737 F1          0.609
2 F1     4      1.04  F1          0.609
3 F2     6      0.262 F2          0.609
4 F2     7      0.395 F2          0.609
```

無作為抽出後のデータについて、応答変数全ての平均を計算

Balanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1・F2ともに、2つずつデータを取り出す #
    .y = c(2, 2),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F)
```

```
# A tibble: 4 x 5
# Groups:   F [2]
  F      id      DV Fb    gross.mean
<fct> <fct> <dbl> <fct>    <dbl>
1 F1     1    0.737 F1      0.609
2 F1     4    1.04  F1      0.609
3 F2     6    0.262 F2      0.609
4 F2     7    0.395 F2      0.609
```

説明変数の水準毎にデータをまとめ上げ

Balanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1・F2ともに、2つずつデータを取り出す #
    .y = c(2, 2),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  )
```

```
# A tibble: 4 x 6
# Groups:   F [2]
  F      id      DV Fb      gross.mean lev.wise.mean
<fct> <fct> <dbl> <fct>      <dbl>          <dbl>
1 F1     1      0.737 F1          0.609          0.890
2 F1     4      1.04  F1          0.609          0.890
3 F2     6      0.262 F2          0.609          0.328
4 F2     7      0.395 F2          0.609          0.328
```

水準毎に応答変数の平均値を計算

Balanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1・F2ともに、2つずつデータを取り出す #
    .y = c(2, 2),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup()
```

```
# A tibble: 4 x 6
  F      id      DV Fb      gross.mean lev.wise.mean
<fct> <fct> <dbl> <fct>      <dbl>         <dbl>
1 F1     1      0.737 F1          0.609         0.890
2 F1     4      1.04  F1          0.609         0.890
3 F2     6      0.262 F2          0.609         0.328
4 F2     7      0.395 F2          0.609         0.328
```

まとめ上げを解除

Balanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1・F2ともに、2つずつデータを取り出す #
    .y = c(2, 2),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup() |>
  distinct(
    F, gross.mean, lev.wise.mean
  )
```

```
# A tibble: 2 x 3
  F      gross.mean lev.wise.mean
<fct>      <dbl>      <dbl>
1 F1         0.609         0.890
2 F2         0.609         0.328
```

応答変数全ての平均と水準毎の平均を抽出

Balanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1・F2ともに、2つずつデータを取り出す #
    .y = c(2, 2),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup() |>
  distinct(
    F, gross.mean, lev.wise.mean
  ) |>
  summarise(
    gross.mean = gross.mean,
    mean.of.lev.wise.mean = mean(lev.wi
  )
```

```
# A tibble: 2 x 2
  gross.mean mean.of.lev.wise.mean
  <dbl>         <dbl>
1   0.609         0.609
2   0.609         0.609
```

水準毎の平均の平均を計算

Balanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1・F2ともに、2つつデータを取り出す #
    .y = c(2, 2),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup() |>
  distinct(
    F, gross.mean, lev.wise.mean
  ) |>
  summarise(
    gross.mean = gross.mean,
    mean.of.lev.wise.mean = mean(lev.wise.mean)
  ) |>
  distinct(
    gross.mean, mean.of.lev.wise.mean
  )
```

```
# A tibble: 1 x 2
  gross.mean mean.of.lev.wise.mean
  <dbl>         <dbl>
1   0.609         0.609
```

「各水準で得られた応答変数の平均値」
の平均値と、全データの平均が一致する

Balanced dataでのGrand Meanの意味

```
set.seed(1)

simdat |>
  group_split(F) |>
  map2_dfr(
#### F1・F2ともに、2つずつデータを取り出す #
    .y = c(2, 2),
    ~ slice_sample(.x, n = .y)
  ) |>
  mutate(
    gross.mean = mean(DV)
  ) |>
  group_by(F) |>
  mutate(
    lev.wise.mean = mean(DV)
  ) |>
  ungroup() |>
  distinct(
    F, gross.mean, lev.wise.mean
  ) |>
  summarise(
    gross.mean = gross.mean,
    mean.of.lev.wise.mean = mean(lev.wise.mean)
  ) |>
  distinct(
    gross.mean, mean.of.lev.wise.mean
  ) |>
```

応答変数全ての平均	「各水準の平均」の平均
0.61	0.61

「各水準で得られた応答変数の平均値」の平均値と、全データの平均が一致する

Balanced dataでのGrand Mean

各水準のデータ数が等しいデータセット（一切欠損値がないデータセット）では、

Grand Meanは、応答変数全体の平均に一致

応答変数全ての平均	「各水準の平均」の平均
0.61	0.61

Unbalanced dataでのGrand Mean

Grand Meanは、**応答変数全体の平均に一致しない**

応答変数全ての平均	「各水準の平均」の平均
0.68	0.58

(*)Cell means parameterization

- 傾き（水準間の差）や切片を推定しない
- 各水準の下で得られる応答変数の平均が0か否かを検定する
 - （2水準の場合）帰無仮説は2つ
 1. $H_0 : \mu_1 = 0$
 2. $H_0 : \mu_2 = 0$

```
m2_mr <- lm(DV ~ -1 + F, simdat)
```

Predictor	Estimate	95% CI	t-value	p-value
FF1	0.8	[0.594, 1.006]	8.94	0
FF2	0.4	[0.194, 0.606]	4.47	0

Examples of different default contrast types (pp.8--9)

さまざまな対比行列

1. Contrast matrices
2. Treatment contrast
3. Sum contrast
4. Repeated contrast
5. Polynomial contrast
6. Helmert contrast
7. 対比行列を充てる方法

Contrast matrices

- 対比を表す係数値（傾き・切片）は，係数値の行列で表現される
- 係数値の行列は，対比行列で実装される

Treatment contrast 処理対比の対比行列

- 例：様々なプライミング効果を，同一の統制群と比べる
 1. プライミングなし（統制群）
 2. 音韻的プライミング（実験群1）
 3. 綴字プライミング（実験群2）

```
#contr.treatment(3)
contr.treatment(demoDataTreatment$priming)
```

	orthographic	phonological
control	0	0
orthographic	1	0
phonological	0	1

Treatment contrast 処理対比の対比行列

- 例：様々なプライミング効果を，同一の統制群と比べる

1. プライミングなし（統制群）
2. 音韻的プライミング（実験群1）
3. 綴字プライミング（実験群2）

```
#contr.treatment(3)
contr.treatment(demoDataTreatment$priming)
```

	orthographic	phonological
control	0	0
orthographic	1	0
phonological	0	1

- 各行：水準 上から
 1. **統制群**：プライミングなし
 2. **実験群1**：音韻的プライミング
 3. **実験群2**：綴字プライミング

Treatment contrast 処理対比の対比行列

- 例：様々なプライミング効果を，同一の統制群と比べる

1. プライミングなし（統制群）
2. 音韻的プライミング（実験群1）
3. 綴字プライミング（実験群2）

```
#contr.treatment(3)
contr.treatment(demoDataTreatment$priming)
```

	orthographic	phonological
control	0	0
orthographic	1	0
phonological	0	1

- 各行：**水準** 上から
 1. **統制群**：プライミングなし
 2. **実験群1**：音韻的プライミング
 3. **実験群2**：綴字プライミング
- 各列：**対比する水準の組合せ** 左から
 1. **実験群1 vs 統制群**
(音韻的プライミング vs プライミングなし)

Sum contrast 零和対比の対比行列

- 例：様々なプライミング効果を，全体平均と比べる
 1. 音韻的プライミング
 2. 綴字プライミング
 3. プライミングなし（統制群）

```
# contr.sum(3)
contr.sum(demoDataSum$priming)
```

	[,1]	[,2]
orthographic	1	0
phonological	0	1
semantic	-1	-1

Sum contrast 零和対比の対比行列

- 例：様々なプライミング効果を，全体平均と比べる
 1. 音韻的プライミング
 2. 綴字プライミング
 3. プライミングなし（統制群）

```
# contr.sum(3)
contr.sum(demoDataSum$priming)
```

	[,1]	[,2]
orthographic	1	0
phonological	0	1
semantic	-1	-1

- 各行：水準 上から
 1. 統制群：プライミングなし
 2. 実験群1：音韻的プライミング
 3. 実験群2：綴字プライミング

Sum contrast 零和対比の対比行列

- 例：様々なプライミング効果を，全体平均と比べる
 1. 音韻的プライミング
 2. 綴字プライミング
 3. プライミングなし（統制群）

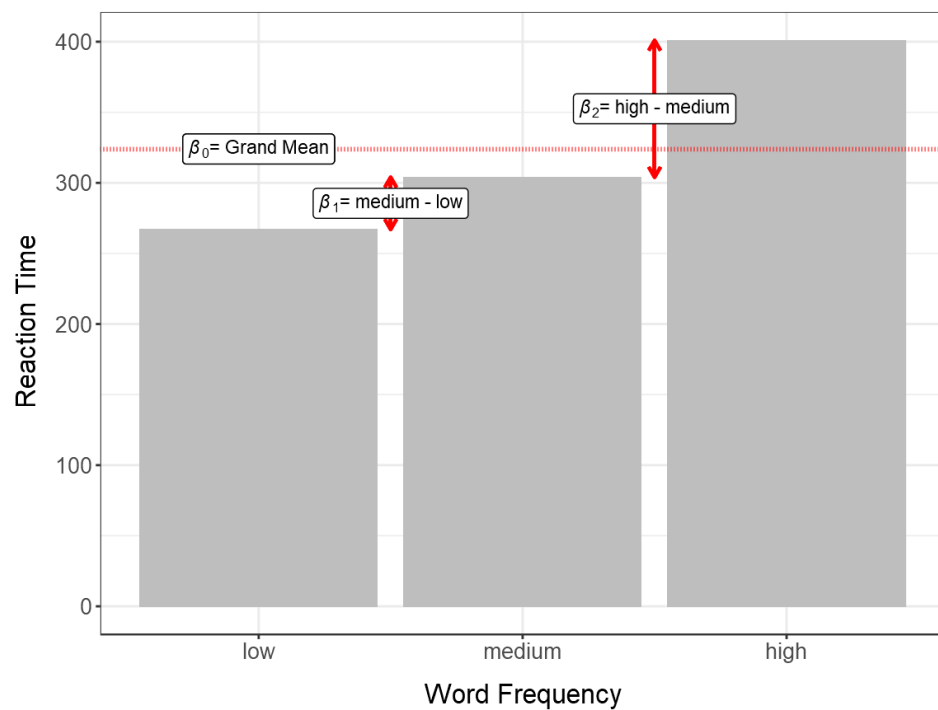
```
# contr.sum(3)
contr.sum(demoDataSum$priming)
```

	[,1]	[,2]
orthographic	1	0
phonological	0	1
semantic	-1	-1

- 各行：水準 上から
 1. 統制群：プライミングなし
 2. 実験群1：音韻的プライミング
 3. 実験群2：綴字プライミング
- 各列：対比する水準の組合せ 左から
 1. 実験群1 vs 全体平均
(音韻的プライミング vs 全体平均)

Repeated contrastの対比行列

- 例：語の頻度の効果を，隣り合う水準と比べる
 1. 低頻度
 2. 中頻度
 3. 高頻度

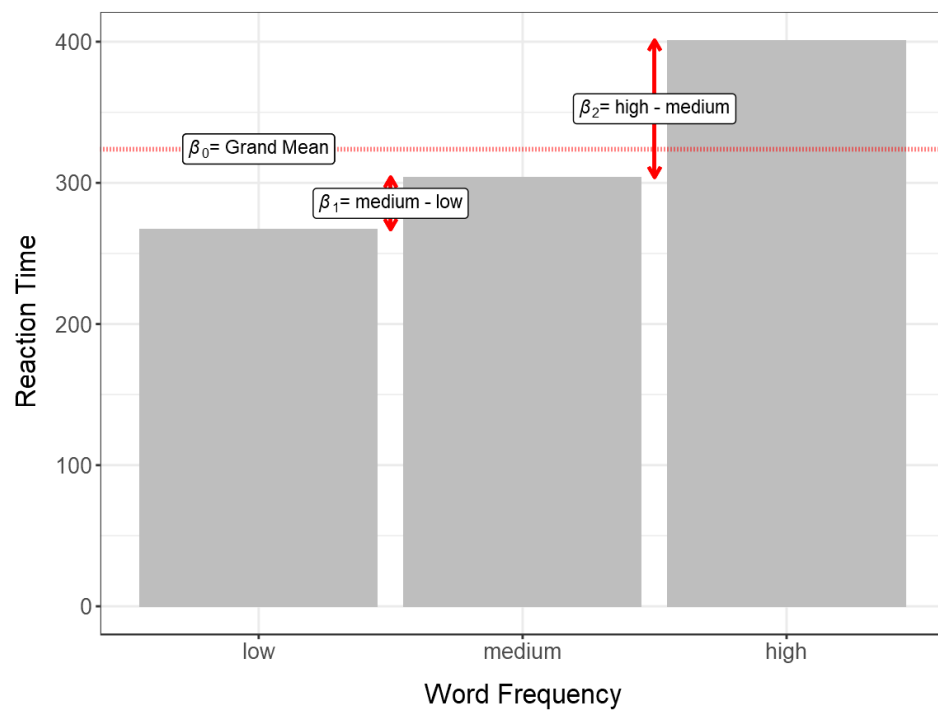


```
#contr.sdif(3)
contr.sdif(demoDataFreq$frequency)
```

	medium-low	high-medium
low	-0.6666667	-0.3333333
medium	0.3333333	-0.3333333
high	0.3333333	0.6666667

Repeated contrastの対比行列

- 例：語の頻度の効果を，隣り合う水準と比べる
 1. 低頻度
 2. 中頻度
 3. 高頻度



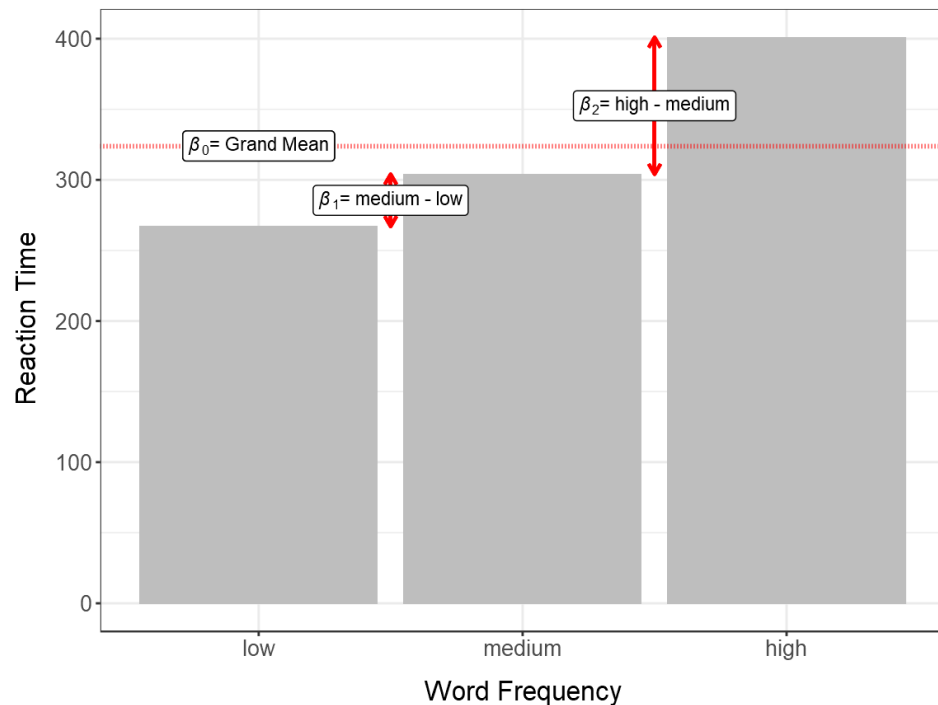
```
#contr.sdif(3)
contr.sdif(demoDataFreq$frequency)
```

	medium-low	high-medium
low	-0.6666667	-0.3333333
medium	0.3333333	-0.3333333
high	0.3333333	0.6666667

- 各行：水準 上から
 1. 低頻度
 2. 中頻度
 3. 高頻度

Repeated contrastの対比行列

- 例：語の頻度の効果を，隣り合う水準と比べる
 1. 低頻度
 2. 中頻度
 3. 高頻度



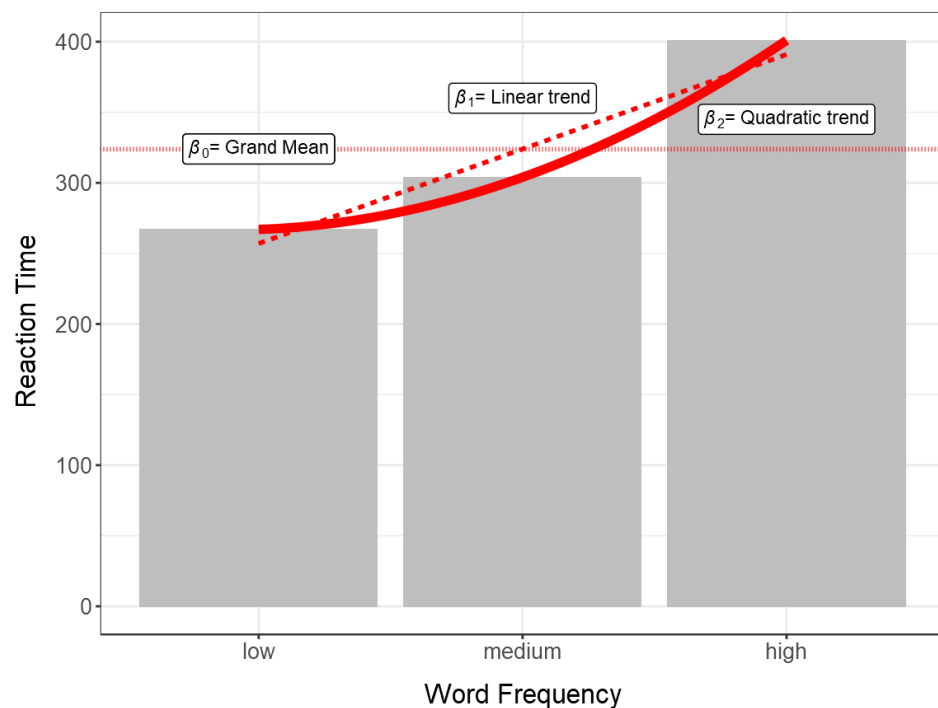
```
#contr.sdif(3)
contr.sdif(demoDataFreq$frequency)
```

	medium-low	high-medium
low	-0.6666667	-0.3333333
medium	0.3333333	-0.3333333
high	0.3333333	0.6666667

- 各行：水準 上から
 1. 低頻度
 2. 中頻度
 3. 高頻度
- 各列：対比する水準の組合せ 左から
 1. 中頻度 vs 低頻度
 2. 高頻度 vs 中頻度

Polynomial contrast 多項式対比の対比行列

- 例：語の頻度の効果を，隣り合う水準と比べる
 1. 低頻度
 2. 中頻度
 3. 高頻度

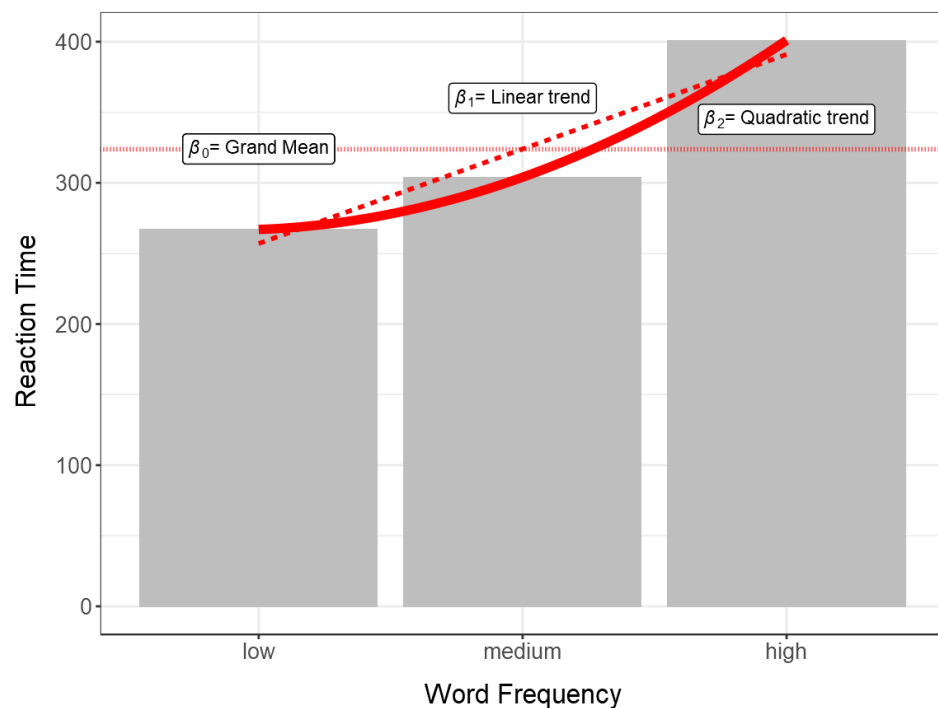


```
#contr.poly(3)
contr.poly(demoDataFreq$frequency)
```

	.L	.Q
[1,]	-7.071068e-01	0.4082483
[2,]	-7.850462e-17	-0.8164966
[3,]	7.071068e-01	0.4082483

Polynomial contrast 多項式対比の対比行列

- 例：語の頻度の効果を，隣り合う水準と比べる
 1. 低頻度
 2. 中頻度
 3. 高頻度



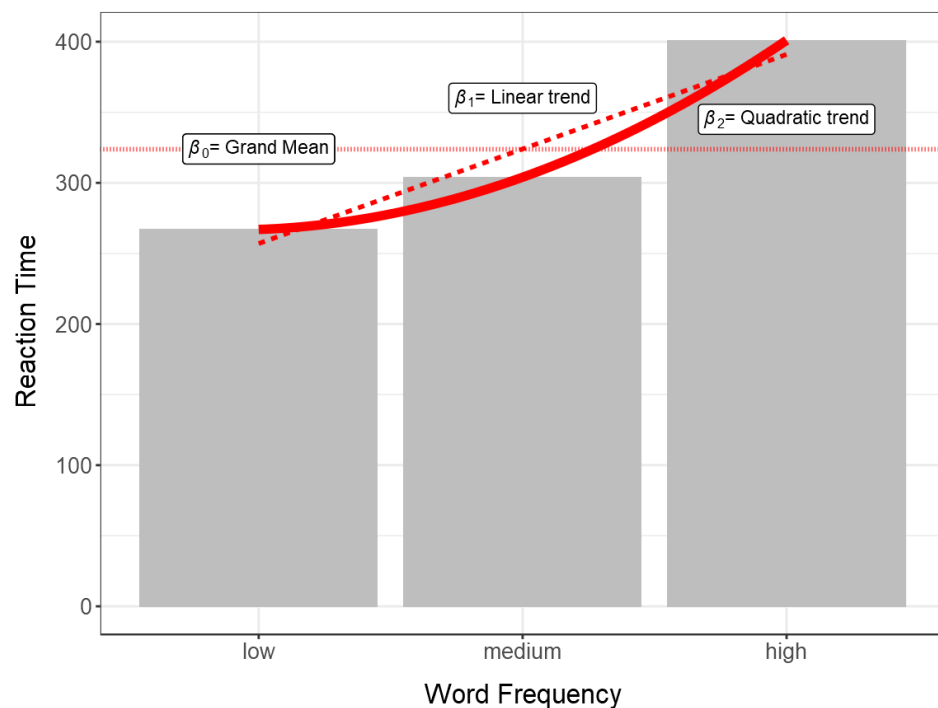
```
#contr.poly(3)
contr.poly(demoDataFreq$frequency)
```

	.L	.Q
[1,]	-7.071068e-01	0.4082483
[2,]	-7.850462e-17	-0.8164966
[3,]	7.071068e-01	0.4082483

- 各行：水準 上から
 1. 低頻度
 2. 中頻度
 3. 高頻度

Polynomial contrast 多項式対比の対比行列

- 例：語の頻度の効果を，隣り合う水準と比べる
 1. 低頻度
 2. 中頻度
 3. 高頻度



```
#contr.poly(3)
contr.poly(demoDataFreq$frequency)
```

	.L	.Q
[1,]	-7.071068e-01	0.4082483
[2,]	-7.850462e-17	-0.8164966
[3,]	7.071068e-01	0.4082483

- 各行：**水準** 上から
 1. 低頻度
 2. 中頻度
 3. 高頻度
- 各列：**対比する水準の組合せ** 左から
 1. 1次関数ペース（Linear; どの水準間でも一定ペースで増加・減少？）

Helmert contrast ヘルマート対比の対比行列

- 例：様々なプライミング効果を，同一の統制群と比べる
 1. 無効なプライミング1
 2. 無効なプライミング2
 3. 有効なプライミング

```
#contr.helmert(3)
contr.helmert(demoDataHelmert$priming)
```

	[,1]	[,2]
invalid1	-1	-1
invalid2	1	-1
valid	0	2

Helmert contrast ヘルマート対比の対比行列

- 例：様々なプライミング効果を，同一の統制群と比べる

1. 無効なプライミング1
2. 無効なプライミング2
3. 有効なプライミング

```
#contr.helmert(3)
contr.helmert(demoDataHelmert$priming)
```

	[,1]	[,2]
invalid1	-1	-1
invalid2	1	-1
valid	0	2

- 各行：**水準** 上から
 1. 無効なプライミング1
 2. 無効なプライミング2
 3. 有効なプライミング

Helmert contrast ヘルマート対比の対比行列

- 例：様々なプライミング効果を，同一の統制群と比べる

1. 無効なプライミング1
2. 無効なプライミング2
3. 有効なプライミング

```
#contr.helmert(3)
contr.helmert(demoDataHelmert$priming)
```

	[,1]	[,2]
invalid1	-1	-1
invalid2	1	-1
valid	0	2

- 各行：**水準** 上から
 1. 無効なプライミング1
 2. 無効なプライミング2
 3. 有効なプライミング
- 各列：**対比する水準の組合せ** 左から
 1. 無効なプライミング1 vs 無効なプライミング2
 2. 無効なプライミング vs 有効なプライミング

```

set.seed(1212); simdatContrasts <- mixedDesign(
#### 被験者間計画での水準数 ####
  B = 3,
#### 被験者内計画での水準数 ####
  W = NULL,
#### 被験者間計画での1水準あたりの実験参加者数 ####
  ##### 実験参加者総数は、`B * n` #####   ##### 実験参加者総変数
  n = 20,
#### 水準ごとの応答変数の平均を示した行列 ####
#### 各行に水準ごとの応答変数の平均を入れる ####
  M = matrix(
    c(600, 400, 300),
    nrow = 3,
    ncol = 1,
    byrow = FALSE
  ),
#### 要因の標準偏差 ####
  SD = 100,
#### 「1試行1データ」の整然データでデータフレームを作成 ####
  long = TRUE
)

```

```

set.seed(1212); simdatContrasts <- mixedDesign(
#### 被験者間計画での水準数 ####
  B = 3,
#### 被験者内計画での水準数 ####
  W = NULL,
#### 被験者間計画での1水準あたりの実験参加者数 ####
  ##### 実験参加者総数は、`B * n` #####   ##### 実験参加者総数
  n = 20,
#### 水準ごとの応答変数の平均を示した行列 ####
#### 各行に水準ごとの応答変数の平均を入れる ####
  M = matrix(
    c(600, 400, 300),
    nrow = 3,
    ncol = 1,
    byrow = FALSE
  ),
#### 要因の標準偏差 ####
  SD = 100,
#### 「1試行1データ」の整然データでデータフレームを作成 ####
  long = TRUE
)

```

