2023.12.15

短期集中研修
『Rデータ解析自由自在（入門編）』

⑤回帰モデル入門

三村　喬生

統計数理研究所
医療健康データ科学研究センター

Day1

- データ科学とは
- データ科学のツール
- Rを始めよう

  基礎知識、データ読み書き

  データの操作、データ可視化

Day2

- 確率の話
- 回帰モデルの話
- 分散分析の話
- 線形混合モデルの話

- 確率の話
  - 確率論の基礎
  - 確率分布 in R
- 回帰モデルの話

回帰の歴史

1650頃
近代的な**「データ」**の概念の確立
ジョン・グラント(英)
ウィリアム・ペティ(英)

1805
「最小二乗法」の発明 **“Ordinary Least Squares”**
アドリアン＝マリ・ルジャンドル(仏)

~1823
棒グラフ, 円グラフ, 時系列グラフ, etc...
ウィリアム・プレイフェア(英)

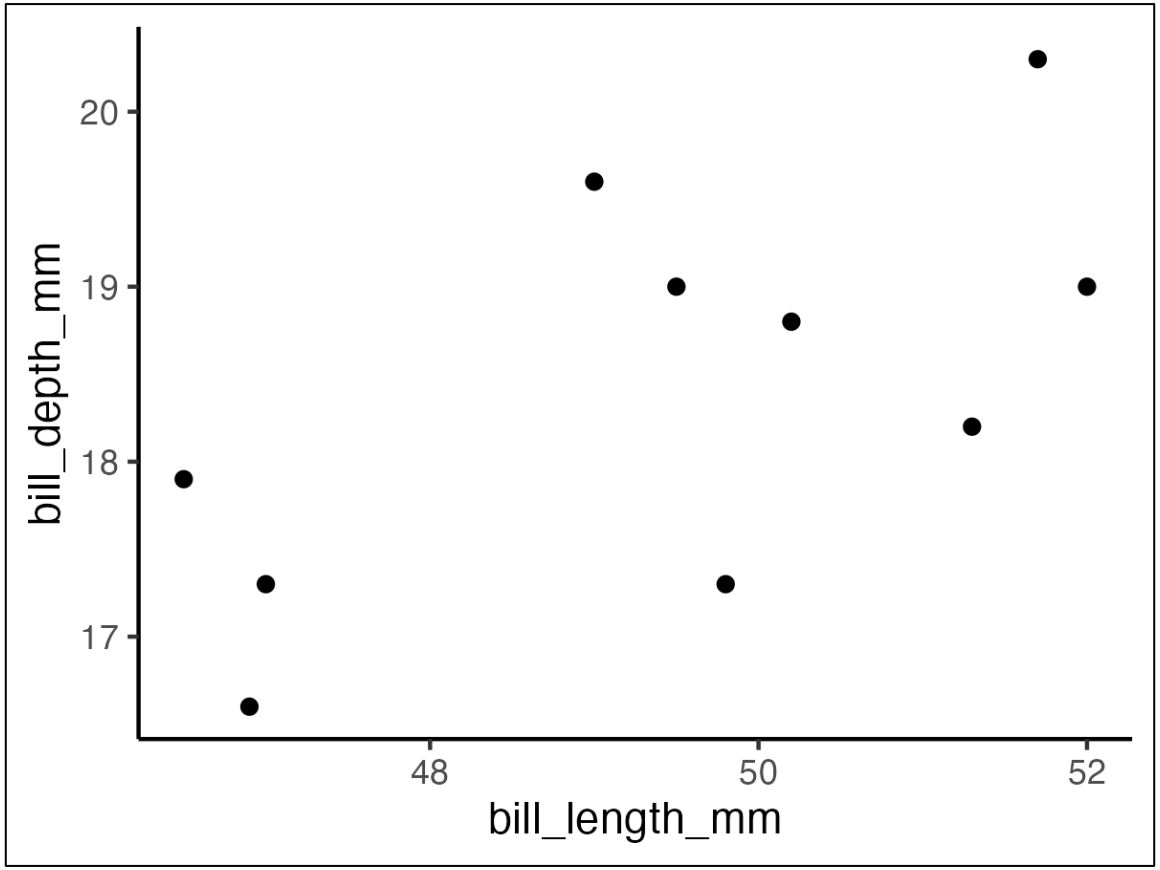
1833
「散布図」概念の誕生
ウィリアム・ハーシェル(英)

1859
『種の起源』出版
チャールズ・ダーウィン(英)

1886
「回帰」概念の確立
フランシス・ゴルトン(英)
“regression to the mean”
“law of universal regression”

データ視覚化の人類史
グラフの発明から時間と空間の可視化まで
Michael Friendly, Howrad Wainer, 飯嶋貴子 (訳) 青土社, 2021

散布図 scatter plot



散布図 scatter plot

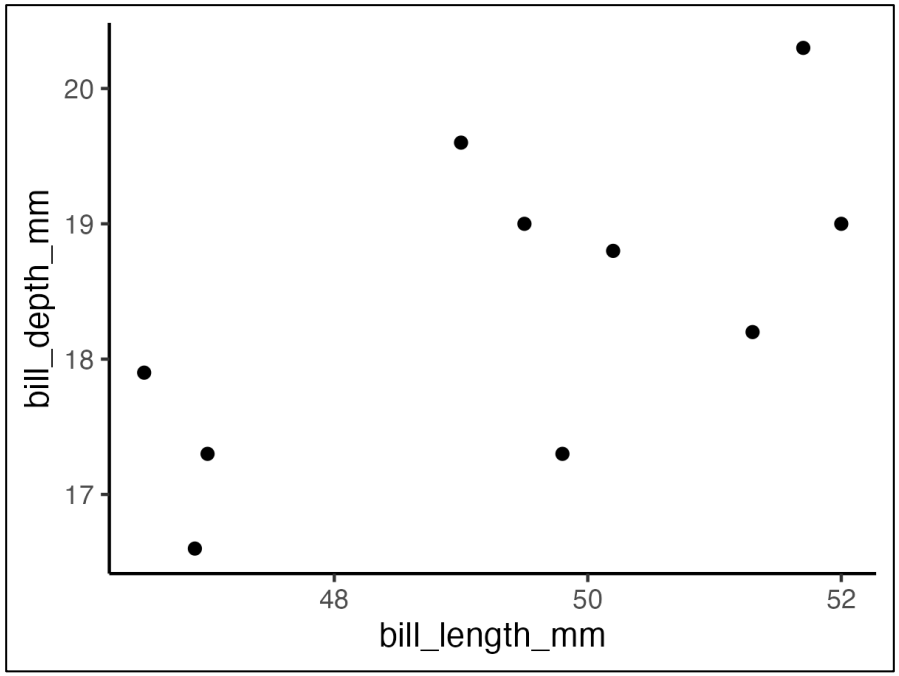
やってみよう

```
## environments ----
library(tidyverse)
library(palmerpenguins)
set.seed(71) # 乱数のシードを指定
theme_set( # ggplotのthemeを一括で指定
  theme_classic() +
    theme(strip.background = element_blank())
)

## data ----
df_raw <-
  penguins %>%
  na.omit() # NAを含む列の除去

df <-
  df_raw %>%
  filter(species == "Chinstrap") %>%
  sample_n(size = 10) # 10行をランダム抽出
```

散布図 scatter plot

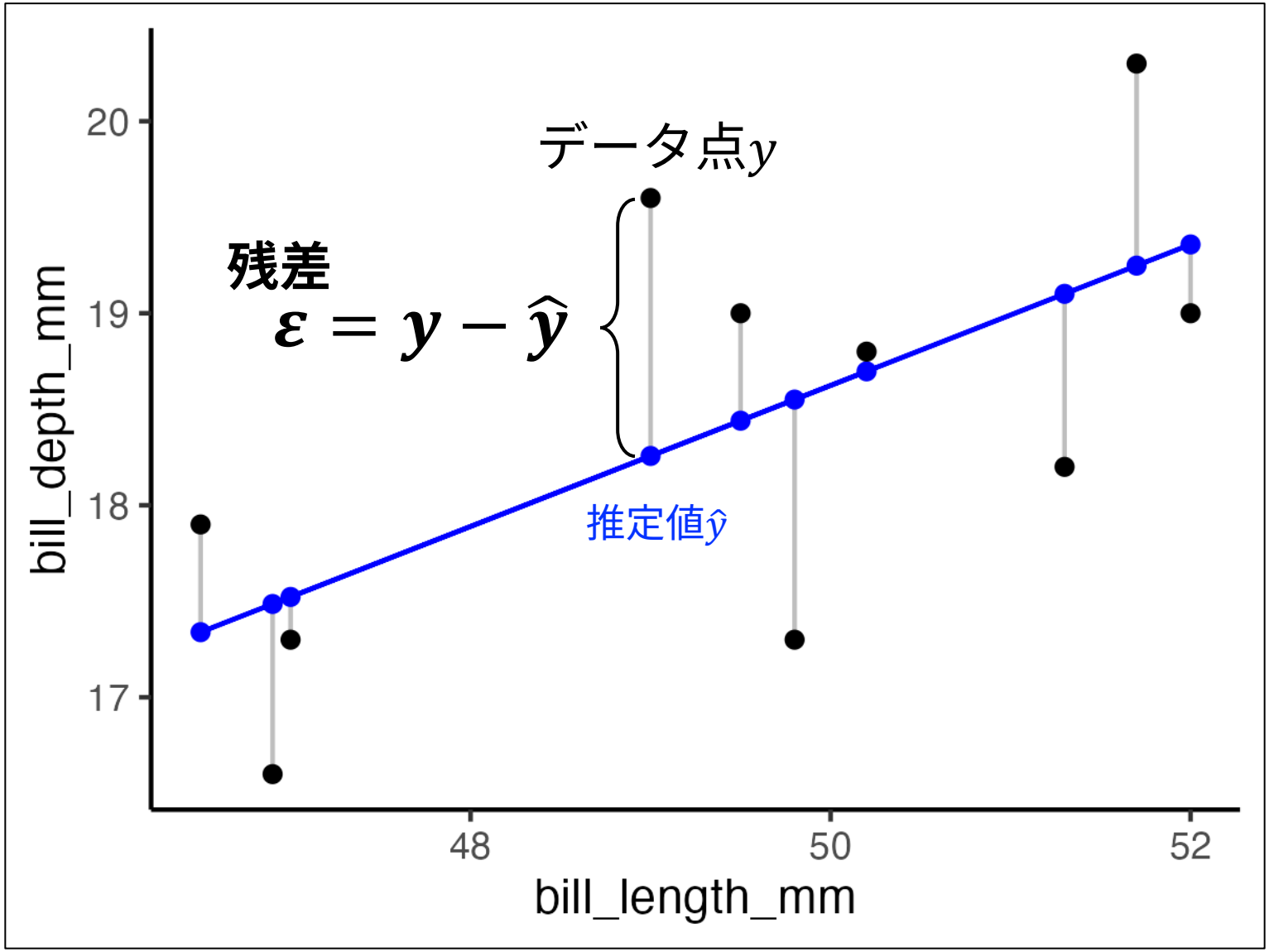
やってみよう

```
## visualization ----
ggplot(data = df) +
  aes(x = bill_length_mm, # mappingの指定
      y = bill_depth_mm) +
  geom_point() # 散布図を描画

ggsave(filename = "fig/fig_lmm_001.png",
       width = 5, height = 3) # 保存
```



最小二乗法 Ordinary Least Squares

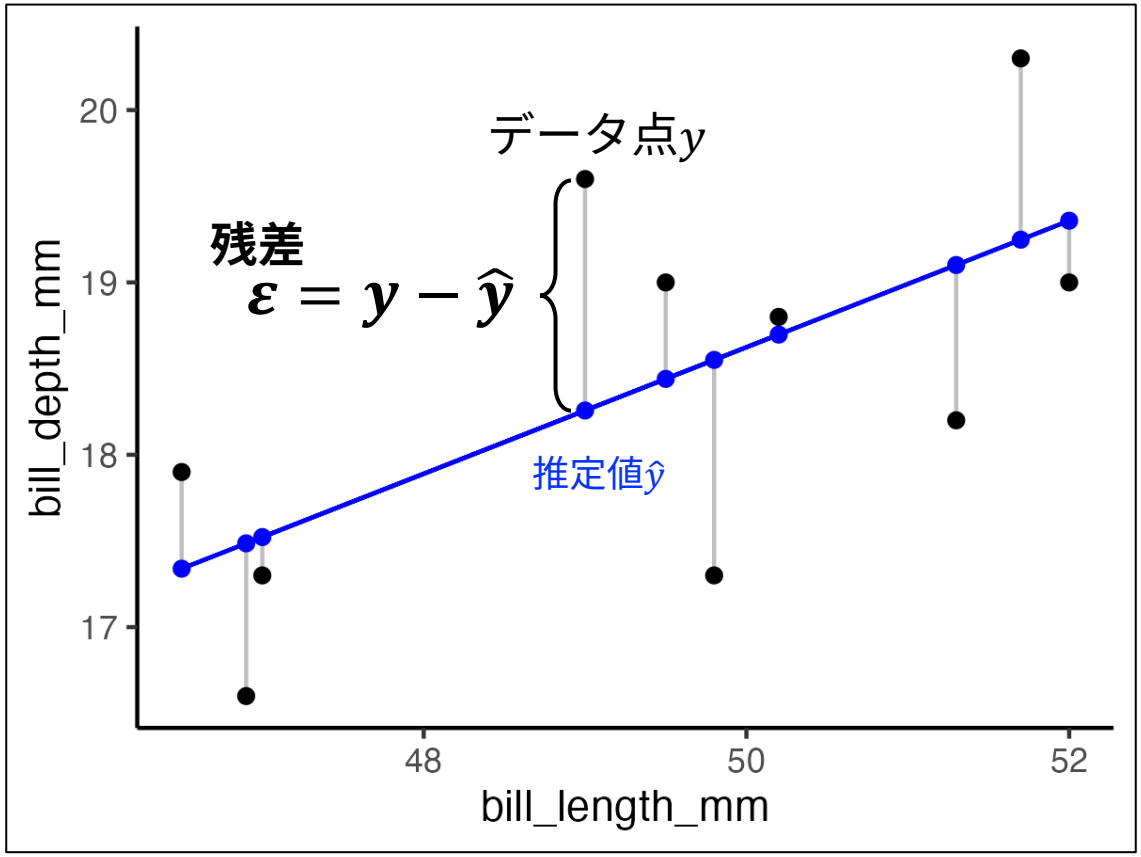
データ点群に対する最も確からしい近似直線を、残差二乗和の最小化により求める



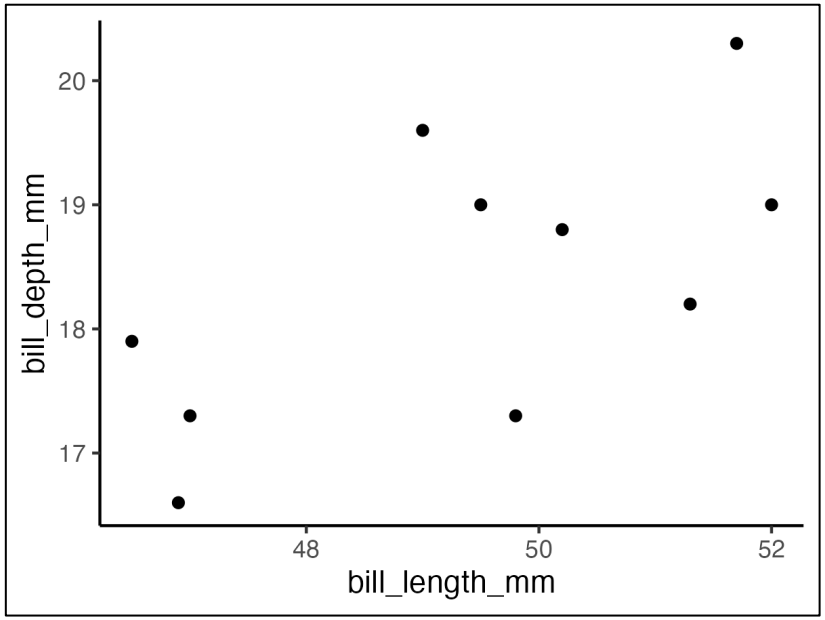
最小二乗法 Ordinary Least Squares

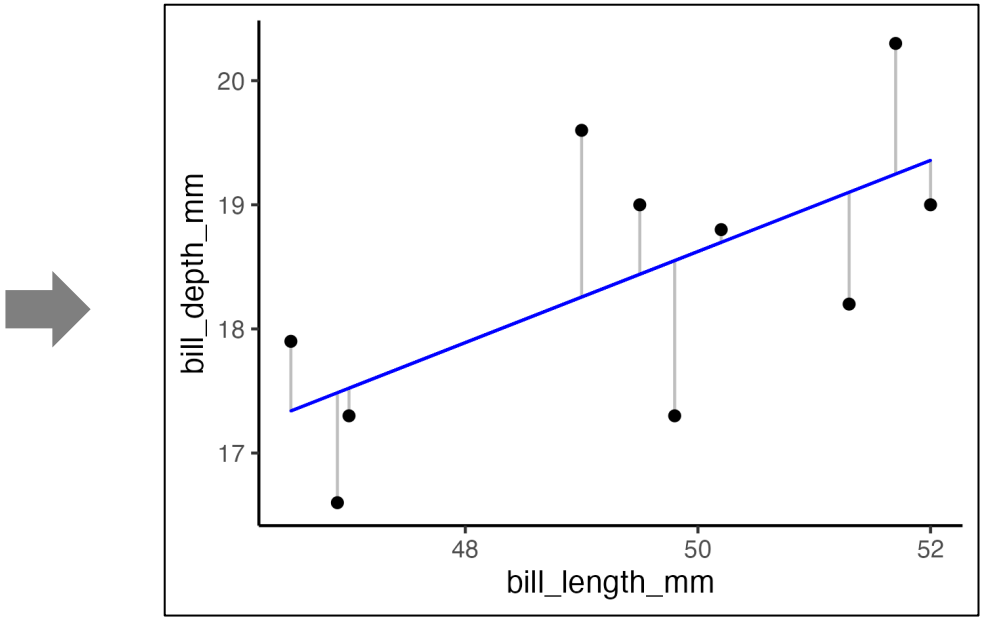
データ点群に対する最も確からしい近似直線を、残差二乗和の最小化により求める

$$y = a_0 + a_1 x + \varepsilon$$

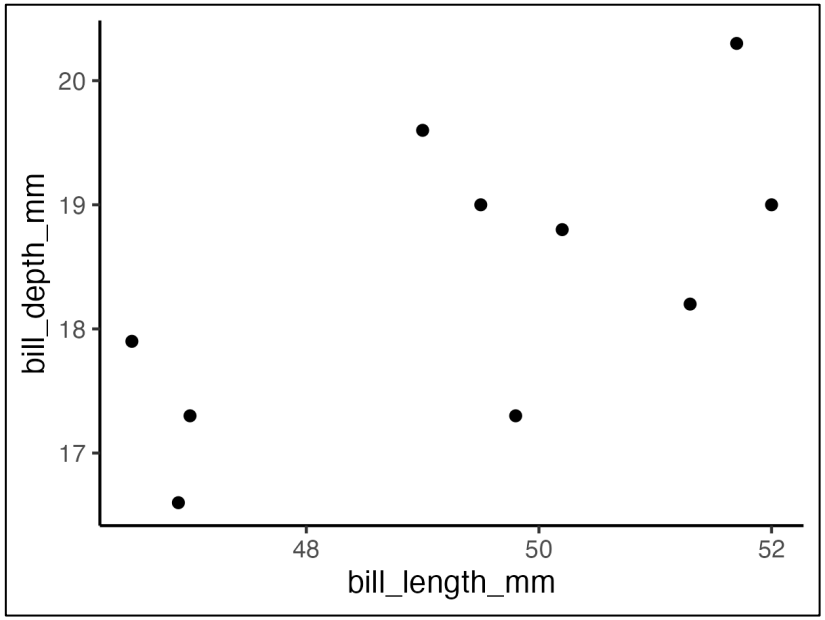


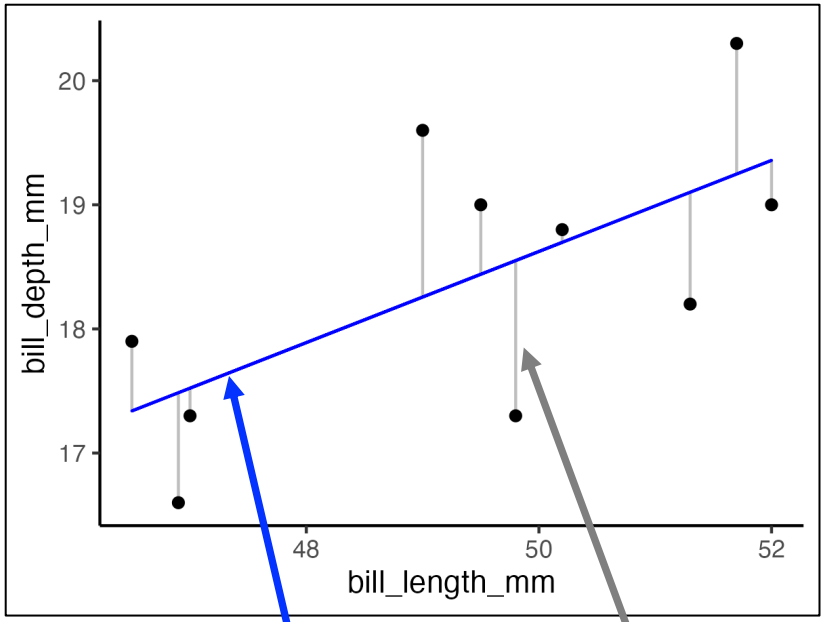
最小二乗法 Ordinary Least Squares





最小二乗法 Ordinary Least Squares

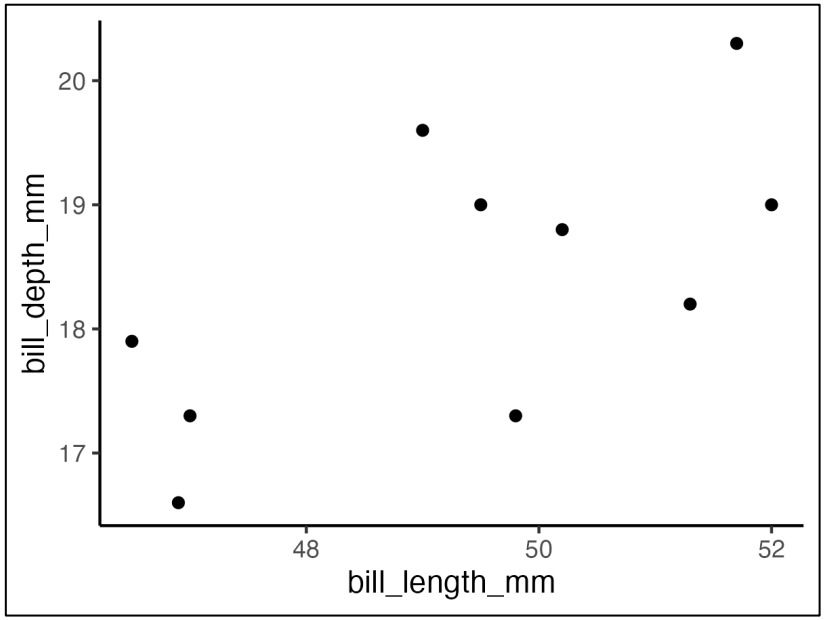


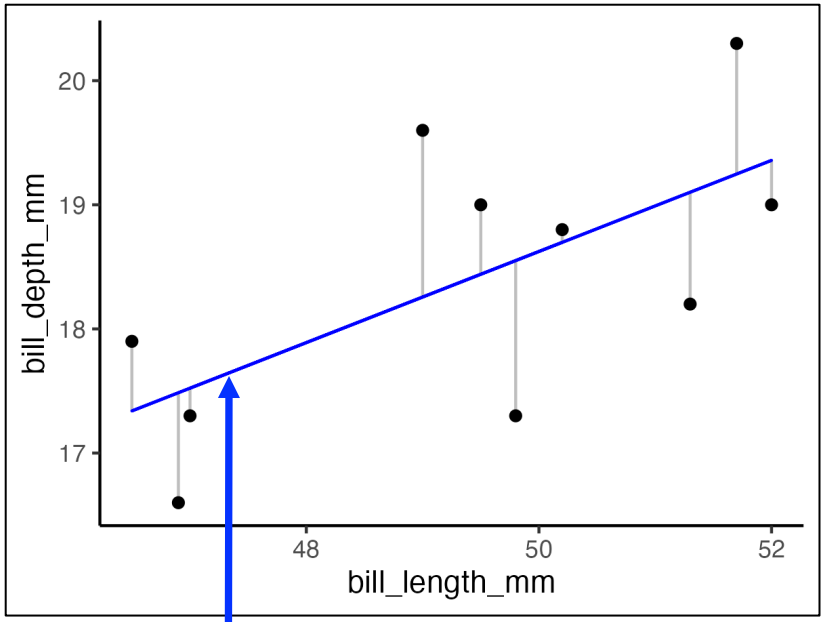


これはgeom_path()関数

灰色線はgeom_segment()関数?

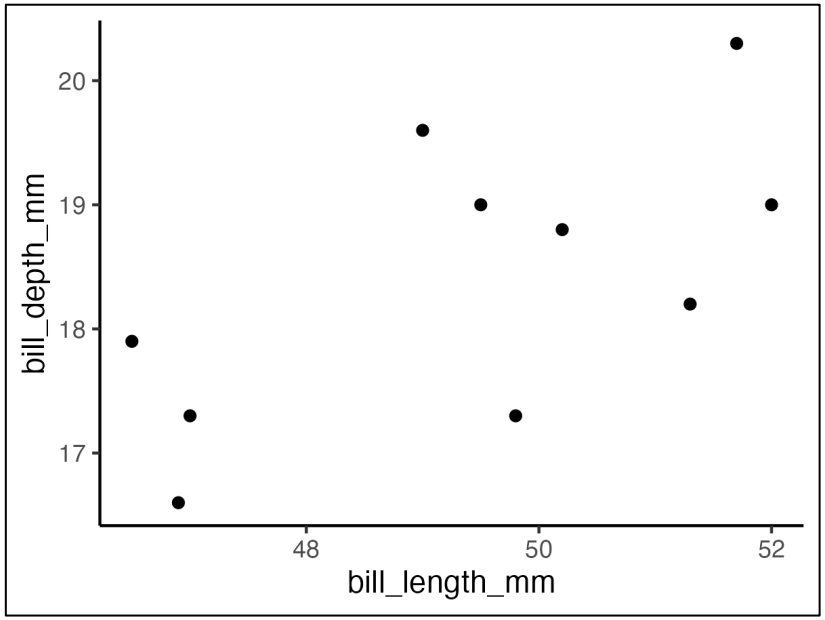
最小二乗法 Ordinary Least Squares

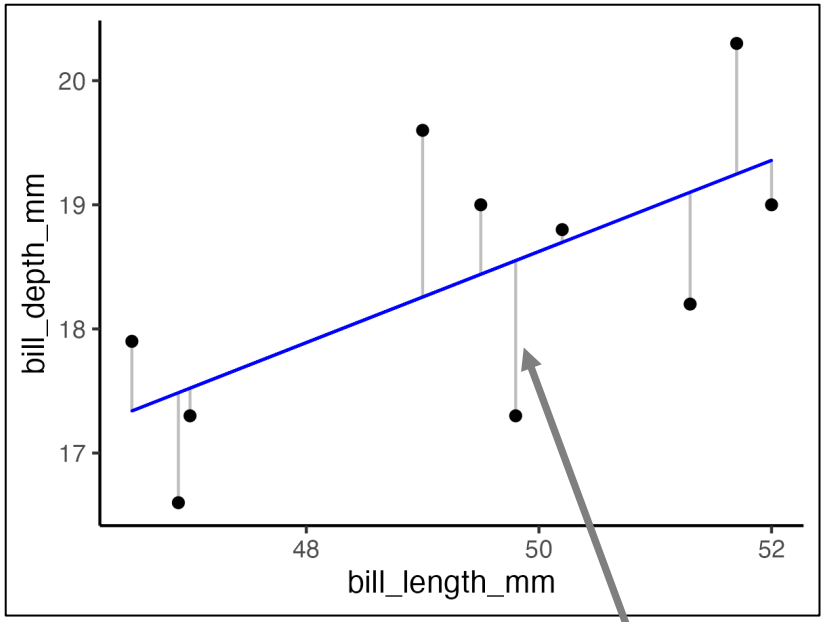





```
# 多分こんな感じにすれば良い
geom_path(
  data = ***
  mapping = aes(x = bill_length_mm,
                y = predicted_bill_depth_mm)
  )
```

最小二乗法 Ordinary Least Squares






```
# 多分こんな感じにすれば良い
geom_segment(
  data = ***
  mapping = aes(xend = bill_length_mm,
                yend = predicted_bill_depth_mm)
  )
```

最小二乗法 Ordinary Least Squares

```
# fit -----
df_fit_lm <-
  lm(formula = bill_depth_mm ~ bill_length_mm,
     data = df)
```

チルダ記号

データフレーム

目的変数
（左辺）

説明変数
（右辺）

最小二乗法 Ordinary Least Squares

```
# fit -----
df_fit_lm <-
  lm(formula = bill_depth_mm ~ bill_length_mm,
     data = df)
```

チルダ記号

データフレーム

目的変数
（左辺）

説明変数
（右辺）

ラムダ式

最小二乗法 Ordinary Least Squares

```
# fit -----
df_fit_lm <-
  lm(formula = bill_depth_mm ~ bill_length_mm,
     data = df)
```

チルダ記号

データフレーム

目的変数
（左辺）

説明変数
（右辺）

ラムダ式

```
# fit -----
df_fit_lm <-
  df %>%
  lm(bill_depth_mm ~ bill_length_mm, data = .)
```

最小二乗法 Ordinary Least Squares

```
# fit -----
df_fit_lm <-
  df %>%
  lm(bill_depth_mm ~ bill_length_mm, data = .)

#> > df_fit_lm
#>
#> Call:
#> lm(formula = bill_depth_mm ~ bill_length_mm, data = .)
#>
#> Coefficients: # 係数
#>   (Intercept)  bill_length_mm
#>        0.2698          0.3671
```

最小二乗法 Ordinary Least Squares

```
# fit -----
df_fit_lm <-
  df %>%
  lm(bill_depth_mm ~ bill_length_mm, data = .)

df_fit_lm %>% names() #.$でアクセスできる要素
#> [1] "coefficients" # 係数
#> [2] "residuals" # 残差
#> [3] "effects" # 効果(QR分解の応答値)
#> [4] "rank" # 順序(説明変数が離散量の時)
#> [5] "fitted.values" # 推定値
#> [6] "assign" # 演算用オブジェクト
#> [7] "qr" # QR分解
#> [8] "df.residual" # 残差自由度
#> [9] "xlevels" # xの水準(説明変数が離散量の時)
#>[10] "call" # ラムダ式
#>[11] "terms" # 演算用オブジェクト
#>[12] "model" # 演算用の数値が格納されたdata.frame
```

最小二乗法 Ordinary Least Squares

```
# fit -----
df_fit_lm <-
  df %>%
  lm(bill_depth_mm ~ bill_length_mm, data = .)

df_fit_lm$coefficients
#>   (Intercept) bill_length_mm
#>     0.2698480      0.3670814
df_fit_lm$residuals
#>          1          2          3          4          5
#>  0.1026640  0.5596210  1.3431618  0.5608653 -0.8859672
#>          6          7          8          9         10
#> -1.2505034 -0.9011255  1.0520419 -0.3580825 -0.2226754
df_fit_lm$fitted.values
#>        1        2        3        4        5        6
#> 18.69734 18.44038 18.25684 17.33913 17.48597 18.55050
#>        7        8        9       10
#> 19.10113 19.24796 19.35808 17.52268
```

最小二乗法 Ordinary Least Squares

```r
# fit -----
df_fit_lm <-
  df %>%
  lm(bill_depth_mm ~ bill_length_mm, data = .)

a0 <- df_fit_lm$coefficients[1] # 切片
a1 <- df_fit_lm$coefficients[2] # 傾き
```

```r
# data transform -----
df_pred <-
  df %>%
  mutate(predict = a0 + a1 * bill_length_mm)
```

最小二乗法 Ordinary Least Squares

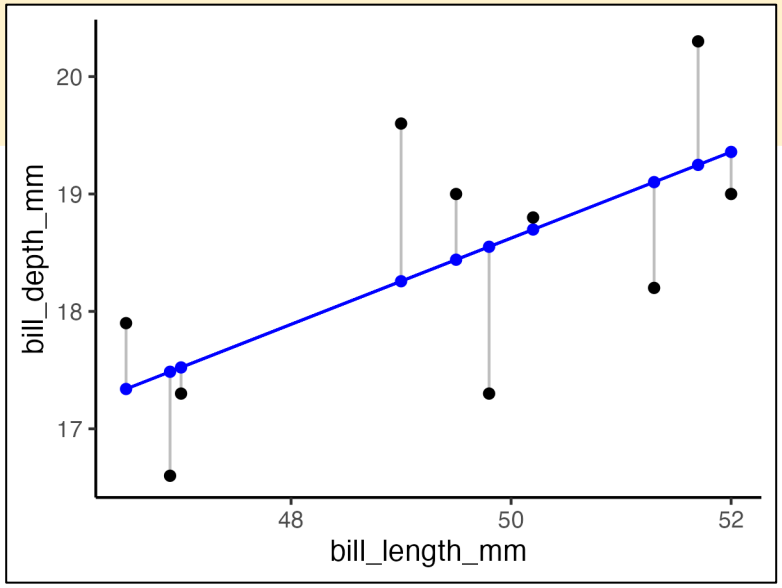
```
# fit ----
df_fit_lm <-
  df %>%
  lm(bill_depth_mm ~ bill_length_mm, data = .)
```

```
# data transform ----
df_pred <-
  df %>%
  mutate(predict = df_fit_lm$fitted.values)
```

```
df_pred <-
  df %>%
  mutate(predict = predict(df_fit_lm))
```

最小二乗法 Ordinary Least Squares

```
## visualization ----
ggplot(data = df_pred) +
  aes(x = bill_length_mm, # mappingの指定
      y = bill_depth_mm) +
  geom_segment(mapping = aes(xend = bill_length_mm,
                             yend = predict),
               color = "grey") +
  geom_path(mapping = aes(y = predict),
            color = "blue") +
  geom_point(mapping = aes(y = predict),
             color = "blue") +
  geom_point()
```



最小二乗法 Ordinary Least Squares

```
## data ----
df <-
 penguins %>%
  na.omit() %>% # NAを含む列の除去
  filter(species == "Chinstrap") %>%
  sample_n(size = 10)# 10行をランダム抽出

# fit ----
df_fit_lm <-
  df %>%
  lm(bill_depth_mm ~ bill_length_mm, data = .)

# data transform ----
df_pred <-
  df %>%
  mutate(predict = predict(df_fit_lm))
```

最小二乗法 Ordinary Least Squares

```
## data ----
df <-
 penguins %>%
  na.omit() %>% # NAを含む列の除去
  filter(species == "Chinstrap") %>%
  sample_n(size = 10) # 10行をランダム抽出

df_pred <-
  df %>%
  group_nest() %>%
  mutate(fit = purrr::map(
    data,
    ~lm(bill_depth_mm ~ bill_length_mm, data = .)
    )) %>%
  mutate(predict = purrr::map(fit, predict)) %>%
  select(-fit) %>%
  unnest(everything())
```

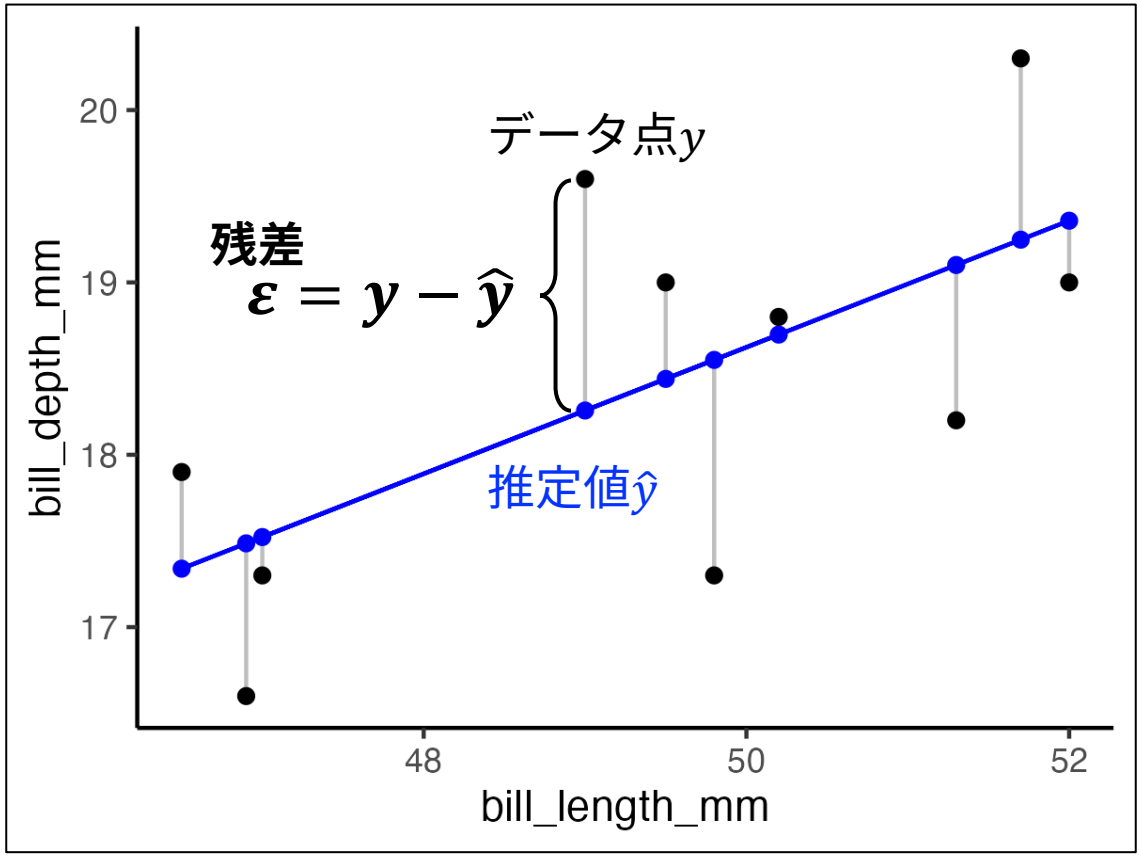
あとで説明します(間に合うか!?)

絶対にパイプを抜けたくないヒト用。

最小二乗法 Ordinary Least Squares

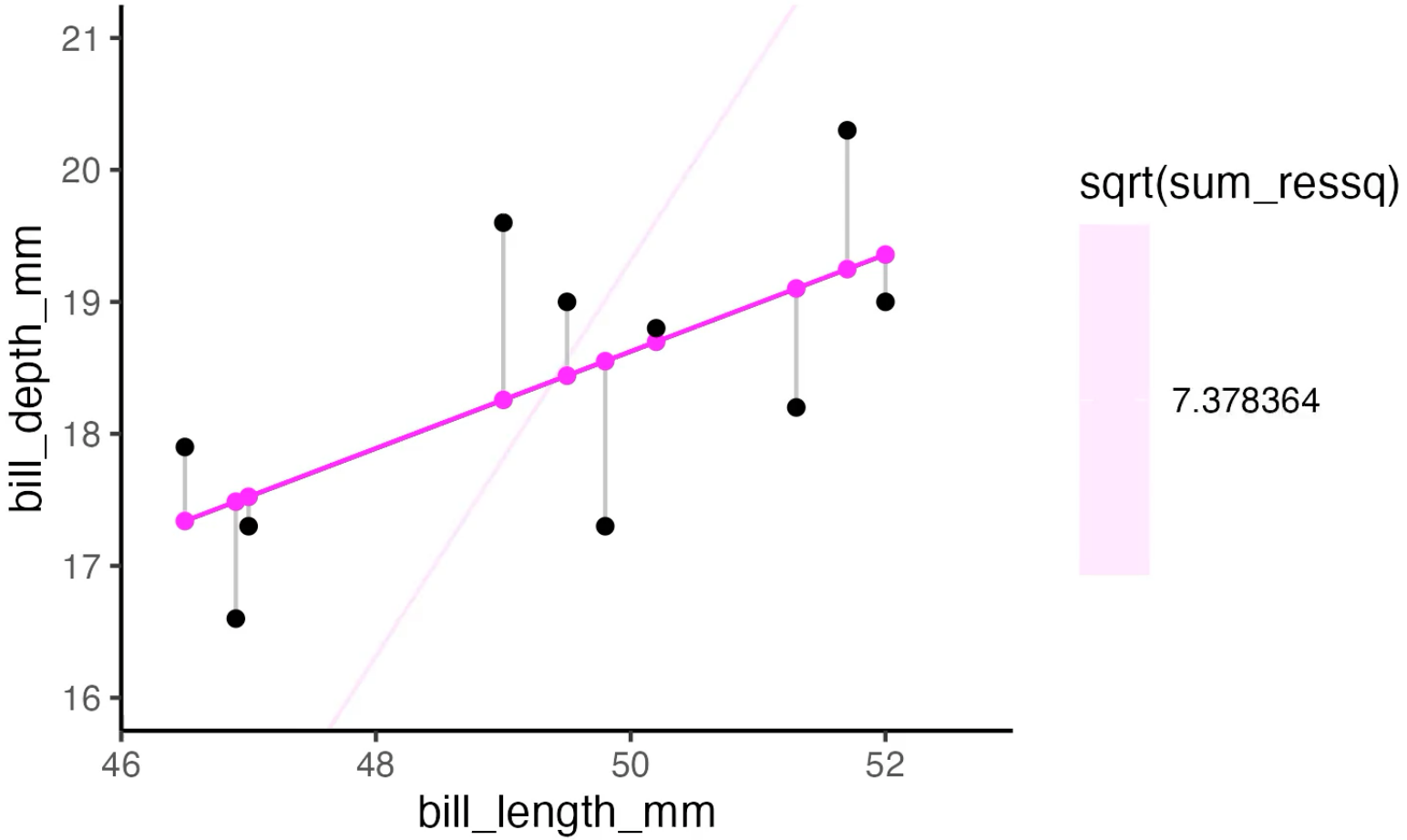
データ点群に対する最も確からしい近似直線を、残差二乗和の最小化により求める

$$y = a_0 + a_1 x + \varepsilon$$



最小二乗法 Ordinary Least Squares

データ点群に対する最も確からしい近似直線を、残差二乗和の最小化により求める



平均点を通る直線のうち傾きが(-1.5, 1.5)範囲のものを、
データ点に対する残差二乗和の平方根に基づいて色分けて表示

最小二乗法 Ordinary Least Squares

推定値　推定値　平均値

$$\mathrm{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2 \Rightarrow \hat{a}_1 = \frac{S_{xy}}{S_{xx}}, \quad \hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}}\bar{x}$$

$$S_{xx} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$S_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

最小二乗法 Ordinary Least Squares

$$\mathrm{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2 = \sum_{i=1}^{n} (y_i - \hat{a}_0 - \hat{a}_1 x_i)^2$$

$$\frac{\partial \sum_{i=1}^{n} (y_i - \hat{a}_0 - \hat{a}_1 x_i)^2}{\partial \hat{\boldsymbol{a}}_{\boldsymbol{0}}} = -2 \sum_{i=1}^{n} (y_i - \hat{a}_0 - \hat{a}_1 x_i)$$

$$= -2 \sum_{i=1}^{n} y_i + 2n\hat{a}_0 + 2\hat{a}_1 \sum_{i=1}^{n} x_i = 0$$

$$a_0 = \frac{1}{n} \sum_{i=1}^{n} y_i - \frac{1}{n} \sum_{i=1}^{n} \hat{a}_1 x_i$$

$$= \bar{y} - \hat{a}_1 \bar{x}$$

最小二乗法 Ordinary Least Squares

$$\mathrm{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2 = \sum_{i=1}^{n} (y_i - \hat{a}_0 - \hat{a}_1 x_i)^2$$

$$\frac{\partial \sum_{i=1}^{n} (y_i - \hat{a}_0 - \hat{a}_1 x_i)^2}{\partial \hat{\boldsymbol{a}}_1} = -2 \sum_{i=1}^{n} (y_i - \hat{a}_0 - \hat{a}_1 x_i) x_i$$

$$= -2 \sum_{i=1}^{n} x_i y_i + 2\hat{a}_0 \sum_{i=1}^{n} x_i + 2\hat{a}_1 \sum_{i=1}^{n} x_i^2$$

$$= -2 \sum_{i=1}^{n} x_i y_i + 2(\bar{y} - \hat{a}_1 \bar{x}) \sum_{i=1}^{n} x_i + 2\hat{a}_1 \sum_{i=1}^{n} x_i^2$$

$$= -2 \sum_{i=1}^{n} x_i y_i + 2\bar{y} \sum_{i=1}^{n} x_i - 2\hat{a}_1 \bar{x} \sum_{i=1}^{n} x_i + 2\hat{a}_1 \sum_{i=1}^{n} x_i^2 = 0$$

最小二乗法 Ordinary Least Squares

$$\mathrm{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2 = \sum_{i=1}^{n} (y_i - \hat{a}_0 - \hat{a}_1 x_i)^2$$

$$-2\sum_{i=1}^{n} x_i y_i + 2\bar{y}\sum_{i=1}^{n} x_i - \hat{a}_1 \bar{x}\sum_{i=1}^{n} x_i + \hat{a}_1 \sum_{i=1}^{n} x_i^2 = 0$$

$$-\sum_{i=1}^{n} x_i y_i + n\bar{x}\bar{y} - \hat{a}_1 \left(n\bar{x}^2 - \sum_{i=1}^{n} x_i^2 \right) = 0$$

$$\hat{a}_1 \left(\sum_{i=1}^{n} x_i^2 - n\bar{x}^2 \right) = \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}$$

$$\hat{a}_1 \left(\sum_{i=1}^{n} (x_i - \bar{x})^2 \right) = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

最小二乗法 Ordinary Least Squares

推定値　推定値　平均値

$$\mathrm{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2 \Rightarrow \hat{a}_1 = \frac{S_{xy}}{S_{xx}}, \quad \hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}}\bar{x}$$

$$S_{xx} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$S_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

最小二乗法 Ordinary Least Squares

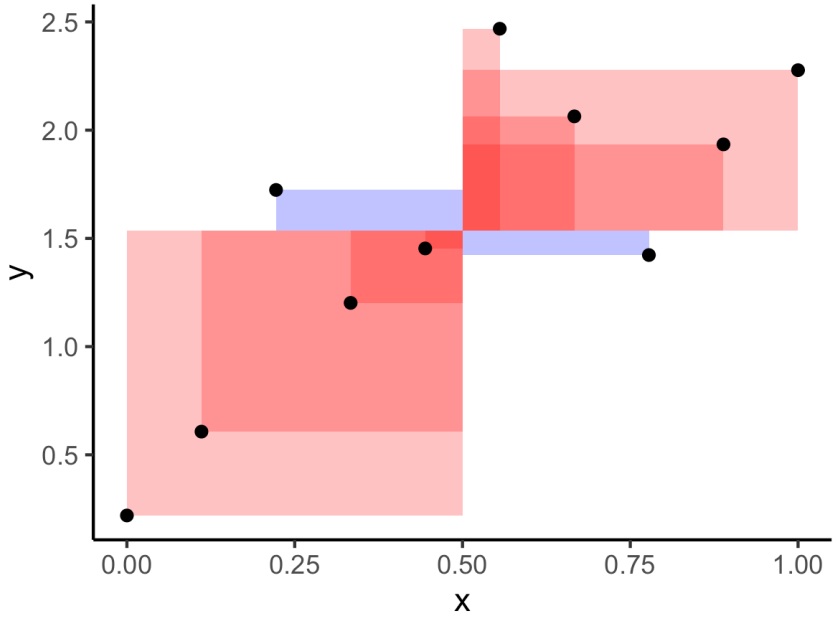
推定値　推定値　平均値

$$\mathrm{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2 \Rightarrow \hat{a}_1 = \frac{S_{xy}}{S_{xx}}, \quad \hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}}\bar{x}$$

$$S_{xx} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$S_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$



最小二乗法

$$\hat{y}_i = a_0 + a_1 x_i,$$
$$\varepsilon_i = y_i - \hat{y}_i$$

$$\xrightarrow{\operatorname{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2}$$

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}},$$
$$\hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}} \bar{x}$$

最小二乗法

$$\hat{y}_i = a_0 + a_1 x_i,$$
$$\varepsilon_i = y_i - \hat{y}_i$$

$$\xrightarrow{\operatorname{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2}$$

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}},$$
$$\hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}} \bar{x}$$

線形回帰モデル

$$Y_i = \alpha_0 + \alpha_1 x_i + u_i,$$
$$u_i \sim_{i.i.d.} N(0, \sigma^2)$$

最小二乗法

$$\hat{y}_i = a_0 + a_1 x_i,$$
$$\varepsilon_i = y_i - \hat{y}_i$$

$$\xrightarrow{\text{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2}$$

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}},$$
$$\hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}} \bar{x}$$

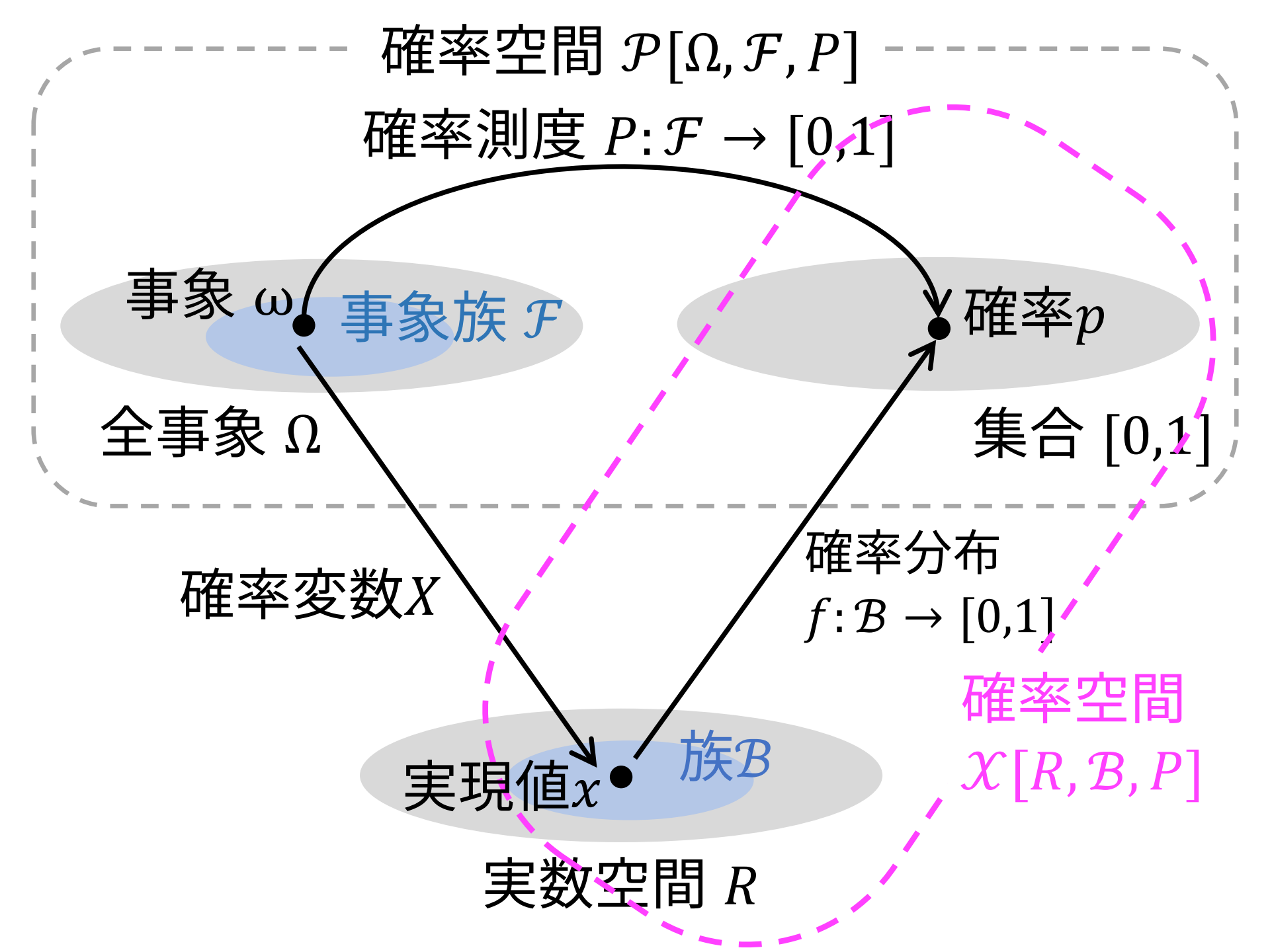
線形回帰モデル

$$Y_i = \alpha_0 + \alpha_1 x_i + u_i,$$

$$u_i \sim_{i.i.d.} N(0, \sigma^2)$$

独立同時分布 independent and identically distributed
標本は個々に独立して同一の分布に従う

確率変数
平均0分散σ^2の正規分布に従う
※ σ^2は未知


確率空間 $\mathcal{P}[\Omega, \mathcal{F}, P]$
確率測度 $P: \mathcal{F} \rightarrow [0,1]$
事象 ω
事象族 $\mathcal{F}$
確率p
全事象 Ω
集合 $[0,1]$
確率変数X
確率分布
$f: \mathcal{B} \rightarrow [0,1]$
確率空間
$\mathcal{X}[R, \mathcal{B}, P]$
実現値x
族$\mathcal{B}$
実数空間 R

最小二乗法

$$\hat{y}_i = a_0 + a_1 x_i,$$
$$\varepsilon_i = y_i - \hat{y}_i$$

$$\xrightarrow{\text{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2}$$

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}},$$
$$\hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}} \bar{x}$$

線形回帰モデル

確率変数

既知の値

$$Y_i = \alpha_0 + \alpha_1 \boldsymbol{x_i} + u_i,$$

$$u_i \sim_{i.i.d.} N(0, \sigma^2)$$

確率変数

独立同時分布 independent and identically distributed

標本は個々に独立して同一の分布に従う

平均0分散σ^2の正規分布に従う

※ σ^2は未知

最小二乗法

$$\hat{y}_i = a_0 + a_1 x_i,$$
$$\varepsilon_i = y_i - \hat{y}_i$$

$$\xrightarrow{\text{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2}$$

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}},$$
$$\hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}} \bar{x}$$

線形回帰モデル

確率変数

既知の値

$$Y_i = \alpha_0 + \alpha_1 \boldsymbol{x_i} + u_i,$$

$$u_i \sim_{i.i.d.} N(0, \sigma^2)$$

$$\Rightarrow Y_i \sim_{i.i.d.} N(\alpha \boldsymbol{x_i} + \beta, \sigma^2)$$

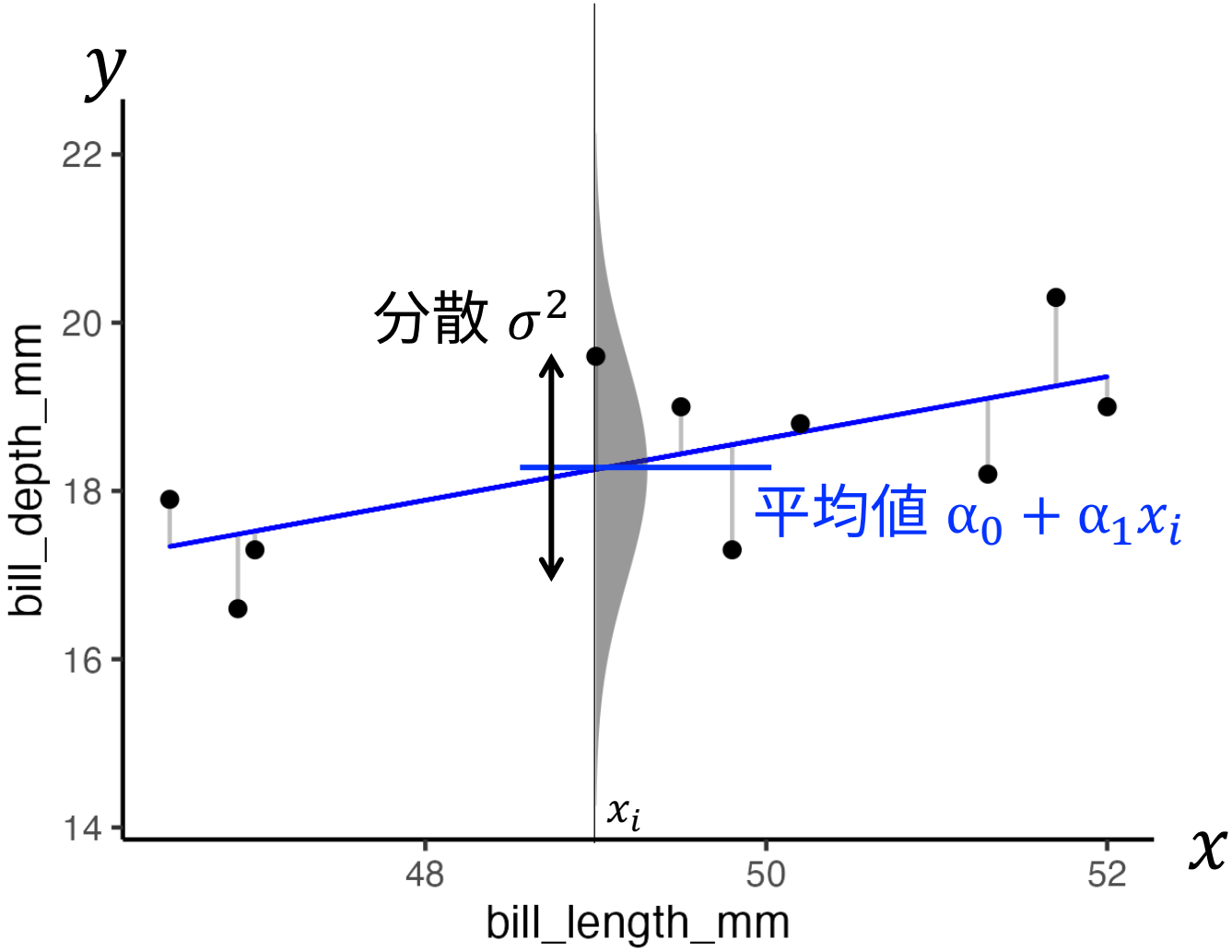
独立同時分布 independent and identically distributed

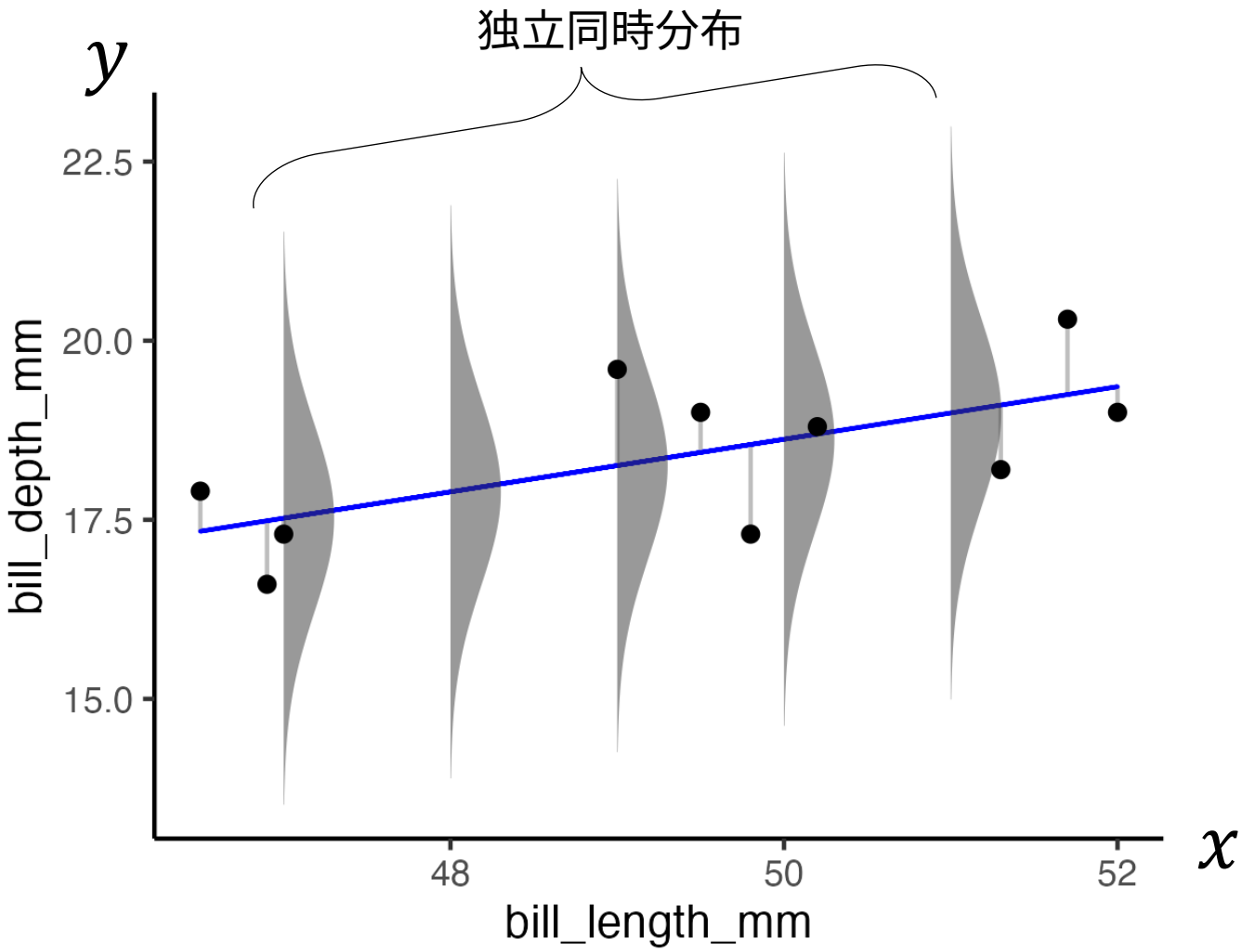
標本は個々に独立して同一の分布に従う

確率変数

平均0分散σ^2の正規分布に従う

※ σ^2は未知


y
22
20
18
16
14
bill_depth_mm
分散 σ^2
平均値 $\alpha_0 + \alpha_1 x_i$
x_i
48
50
52
x
bill_length_mm


独立同時分布
y
22.5
20.0
17.5
15.0
bill_depth_mm
48
50
52
bill_length_mm
x

最小二乗法

$$\hat{y}_i = a_0 + a_1 x_i,$$
$$\varepsilon_i = y_i - \hat{y}_i$$

$$\xrightarrow{\text{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2}$$

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}},$$
$$\hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}} \bar{x}$$

線形回帰モデル

$$Y_i = \alpha_0 + \alpha_1 x_i + u_i,$$
$$u_i \sim_{i.i.d.} N(0, \sigma^2)$$

最小二乗法

$$\hat{y}_i = a_0 + a_1 x_i,$$
$$\varepsilon_i = y_i - \hat{y}_i$$

$$\xrightarrow{\text{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2}$$

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}},$$
$$\hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}} \bar{x}$$

線形回帰モデル

$$Y_i = \alpha_0 + \alpha_1 x_i + u_i,$$
$$u_i \sim_{i.i.d.} N(0, \sigma^2)$$

$$\alpha_0 = E[\hat{a}_0],$$
$$\alpha_1 = E[\hat{a}_1]$$

↑

期待値

あとでやります

最小二乗法

$$\hat{y}_i = a_0 + a_1 x_i,$$
$$\varepsilon_i = y_i - \hat{y}_i$$

$$\xrightarrow{\operatorname{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2}$$

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}},$$
$$\hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}} \bar{x}$$

線形回帰モデル

$$Y_i = \alpha_0 + \alpha_1 x_i + u_i,$$
$$u_i \sim_{i.i.d.} N(0, \sigma^2)$$

$$\alpha_0 = E[\hat{a}_0] = \hat{a}_0,$$
$$\alpha_1 = E[\hat{a}_1] = \hat{a}_1$$

$$S_{xx} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2, \quad S_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

$$\alpha_1 = E[\hat{a}_1] = E\left[\frac{S_{xy}}{S_{xx}}\right] = \frac{\frac{n}{n-1}S_{xy}}{\frac{n}{n-1}S_{xx}} = \frac{\frac{n}{n-1}\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\frac{n}{n-1}\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \frac{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$:= \frac{cov(x,y)}{var(x)}$$

不偏推定量

共分散

$$cov(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}),$$

分散

$$var(x) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

最小二乗法

$$\hat{y}_i = a_0 + a_1 x_i,$$
$$\varepsilon_i = y_i - \hat{y}_i$$

$$\xrightarrow{\text{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2}$$

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}},$$
$$\hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}} \bar{x}$$

線形回帰モデル

$$Y_i = \alpha_0 + \alpha_1 x_i + u_i,$$
$$u_i \sim_{i.i.d.} N(0, \sigma^2)$$

$$\alpha_0 = E[\hat{a}_0] = \bar{y} - \frac{cov(x,y)}{var(x)} \bar{x},$$

$$\alpha_1 = E[\hat{a}_1] = \frac{cov(x,y)}{var(x)}$$

回帰モデル regression

$$Y_i = \alpha_0 + \alpha_1 x_i + u_i,$$

$$u_i \sim_{i.i.d.} N(0, \sigma^2)$$

やってみよう

$$\alpha_0 = E[\hat{a}_0] = \bar{y} - \frac{cov(x,y)}{var(x)}\bar{x},$$

$$\alpha_1 = E[\hat{a}_1] = \frac{cov(x,y)}{var(x)}$$

定義通りに計算し、
演算結果を確認する

共分散

$$cov(x,y) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}),$$

分散

$$var(x) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

回帰モデル regression

```
df_xy <-
  df %>%
  rename(x = bill_length_mm,
         y = bill_depth_mm) %>%
  select(x, y)

#> # A tibble: 10 × 2
#>         x     y
#>     <dbl> <dbl>
#>  1  50.2  18.8
#>  2  49.5  19
#>  3  49    19.6
#>  4  46.5  17.9
#>  5  46.9  16.6
#>  6  49.8  17.3
#>  7  51.3  18.2
#>  8  51.7  20.3
#>  9  52    19
#> 10  47    17.3
```

回帰モデル regression

```
df_xy <-
  df %>%
  rename(x = bill_length_mm,
         y = bill_depth_mm) %>%
  select(x, y) %>%
  mutate(d_x = x - mean(x),
         d_y = y - mean(y))
#> # A tibble: 10 × 4
#>       x     y    d_x    d_y
#>   <dbl> <dbl>  <dbl>  <dbl>
#>  1  50.2  18.8  0.810  0.400
#>  2  49.5  19    0.110  0.600
#>  3  49    19.6 -0.390  1.20
#>  4  46.5  17.9 -2.89  -0.5
#>  5  46.9  16.6 -2.49  -1.80
#>  6  49.8  17.3  0.410 -1.10
#>  7  51.3  18.2  1.91  -0.200
#>  8  51.7  20.3  2.31   1.90
#>  9  52    19    2.61   0.600
#> 10  47    17.3 -2.39  -1.10
```

回帰モデル regression

```
df_xy <-
  df %>%
  rename(x = bill_length_mm,
         y = bill_depth_mm) %>%
  select(x, y) %>%
  mutate(d_x = x - mean(x),
         d_y = y - mean(y)) %>%
  mutate(dxdy= d_x * d_y,
         dx2 = d_x^2)
#> # A tibble: 10 × 6
#>       x     y    d_x    d_y    dxdy    dx2
#>    <dbl> <dbl>  <dbl>  <dbl>   <dbl>  <dbl>
#>  1  50.2  18.8  0.810  0.400  0.324  0.656
#>  2  49.5  19    0.110  0.600  0.0660 0.0121
#>  3  49    19.6 -0.390  1.20  -0.468  0.152
#>  4  46.5  17.9 -2.89  -0.5    1.45   8.35
#>  5  46.9  16.6 -2.49  -1.80   4.48   6.20
#>  6  49.8  17.3  0.410 -1.10  -0.451  0.168
#>  7  51.3  18.2  1.91  -0.200 -0.382  3.65
#>  8  51.7  20.3  2.31   1.90   4.39   5.34
#>  9  52    19    2.61   0.600  1.57   6.81
```

回帰モデル regression

```
df_xy <-
  df %>%
  rename(x = bill_length_mm,
         y = bill_depth_mm) %>%
  select(x, y) %>%
  mutate(d_x = x - mean(x),
         d_y = y - mean(y)) %>%
  mutate(dxdy= d_x * d_y,
         dx2 = d_x^2)

df_xy %>%
  summarise(covxy = sum(dxdy) / (nrow(.) - 1),
            varx = sum(dx2) / (nrow(.) - 1))
#> # A tibble: 1 × 2
#>   covxy  varx
#>   <dbl> <dbl>
#> 1  1.51  4.12
```

回帰モデル regression

```
df_xy %>%
  summarise(covxy = sum(dxdy) / (nrow(.) - 1),
            varx = sum(dx2) / (nrow(.) - 1))
#> # A tibble: 1 × 2
#>   covxy  varx
#>   <dbl> <dbl>
#> 1  1.51  4.12

var(df_xy$x)
#> [1] 4.116556

cov(df_xy$x, df_xy$y)
#> [1] 1.511111
```

回帰モデル regression

```
df_xy %>%
  summarise(covxy = sum(dxdy) / (nrow(.) - 1),
            varx = sum(dx2) / (nrow(.) - 1))
#> # A tibble: 1 × 2
#>   covxy  varx
#>   <dbl> <dbl>
#> 1  1.51  4.12

library(magrittr)

df_xy %$% var(x)
#> [1] 4.116556

df_xy %$% cov(x, y)
#> [1] 1.511111
```

絶対に「data.frame名$変数名」を書きたくないヒト向け。

最小二乗法

$$\hat{y}_i = a_0 + a_1 x_i,$$
$$\varepsilon_i = y_i - \hat{y}_i$$

$$\xrightarrow{\text{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2}$$

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}},$$
$$\hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}} \bar{x}$$

線形回帰モデル

$$Y_i = \alpha_0 + \alpha_1 x_i + u_i,$$
$$u_i \sim_{i.i.d.} N(0, \sigma^2)$$

$$\alpha_0 = E[\hat{a}_0] = \bar{y} - \frac{cov(x,y)}{var(x)} \bar{x},$$

$$\alpha_1 = E[\hat{a}_1] = \frac{cov(x,y)}{var(x)}$$

回帰モデル regression

```
varx <- df_xy %$% var(x)
covxy <- df_xy %$% cov(x, y)

alpha1 <- covxy / varx
#>[1] 0.3670814

alpha0 <- mean(df_xy$y) - alpha1 * mean(df_xy$x)
#>[1] 0.269848

lm(formula = y ~ x, data = df_xy)
#>
#> Call:
#> lm(formula = y ~ x, data = df_xy)
#>
#> Coefficients:
#> (Intercept)           x
#>      0.2698      0.3671
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max 
-1.25050 -0.75400 -0.06001  0.56055  1.34316 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)   0.2698     7.5348   0.036   0.9723  
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478 
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()
```

やってみよう

```
Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

```
df_res <-
  df_xy %>%
  mutate(y_hat = alpha0 + alpha1 * x,
         res = y - y_hat)
#> > df_res
#> # A tibble: 10 × 8
#>        x     y    d_x    d_y    dxdy    dx2 y_hat    res
#>    <dbl> <dbl>  <dbl>  <dbl>   <dbl>  <dbl> <dbl>  <dbl>
#>  1  50.2  18.8  0.810  0.400  0.324  0.656   18.7  0.103
#>  2  49.5  19    0.110  0.600  0.0660 0.0121  18.4  0.560
#>  3  49    19.6 -0.390  1.20  -0.468  0.152   18.3  1.34
#>  4  46.5  17.9 -2.89  -0.5    1.45   8.35    17.3  0.561
#>  5  46.9  16.6 -2.49  -1.80   4.48   6.20    17.5 -0.886
#>  6  49.8  17.3  0.410 -1.10  -0.451  0.168   18.6 -1.25
#>  7  51.3  18.2  1.91  -0.200 -0.382  3.65    19.1 -0.901
#>  8  51.7  20.3  2.31   1.90   4.39   5.34    19.2  1.05
#>  9  52    19    2.61   0.600  1.57   6.81    19.4 -0.358
#> 10  47    17.3 -2.39  -1.10   2.63   5.71    17.5 -0.223
```

回帰モデル regression

```
df_res <-
  df_xy %>%
  mutate(y_hat = alpha0 + alpha1 * x,
         res = y - y_hat)

quantile(df_res$res, probs = c(0.25, 0.75))
#>       25%       75%
#> -0.7539961  0.5605543

max(df_res$res)
#> [1] 1.343162

min(df_res$res)
#> [1] -1.250503

median(df_res$res)
#> [1] -0.06000567
```

回帰モデル regression

```
df_res <-
  df_xy %>%
  mutate(y_hat = alpha0 + alpha1 * x,
         res = y - y_hat)

df_res %>%
  summarize(
    Min = min(res),
    Q1 = quantile(res, probs = 0.25),
    Median = median(res),
    Q3 = quantile(res, probs = 0.75),
    Max = max(res)
  )
#> # A tibble: 1 × 5
#>     Min     Q1  Median    Q3   Max
#>   <dbl>  <dbl>   <dbl> <dbl> <dbl>
#> 1 -1.25 -0.754 -0.0600 0.561  1.34
```

回帰モデル regression

```
df_res <-
  df_xy %>%
  mutate(y_hat = alpha0 + alpha1 * x,
         res = y - y_hat)

df_res %>%
  summarize(
    dplyr::across(
      res,
      list(Min = ~ min(.),
           Q1 = ~ quantile(., probs = 0.25),
           Median = ~ median(.),
           Q3 = ~ quantile(., probs = 0.75),
           Max = ~ max(.)
      )))
#> # A tibble: 1 × 5
#>   res_Min res_Q1 res_Median res_Q3 res_Max
#>     <dbl>  <dbl>      <dbl>  <dbl>   <dbl>
#> 1   -1.25 -0.754    -0.0600  0.561    1.34
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

標準誤差　standard error

$$SE = \frac{SD}{\sqrt{n}}$$

標本平均の標準偏差

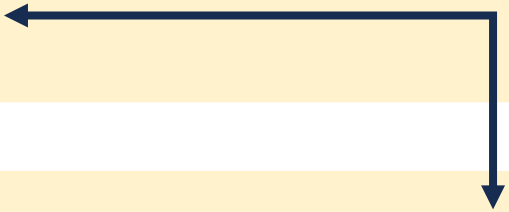
$$SD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

回帰モデル regression

```
df_res <-
  df_xy %>%
  mutate(y_hat = alpha0 + alpha1 * x,
         res = y - y_hat)

df_res %>% summarise(SE = sd(res) / sqrt(nrow(.)))

#> # A tibble: 1 × 1
#>       SE
#>    <dbl>
#> 1 0.277
```

回帰モデル regression

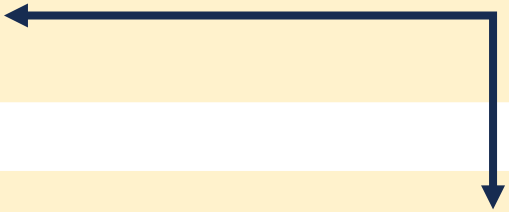
```
df_res <-
  df_xy %>%
  mutate(y_hat = alpha0 + alpha1 * x,
         res = y - y_hat)

df_res %>% summarise(SE = sd(res) / sqrt(nrow(.)))

#> # A tibble: 1 × 1
#>      SE
#>   <dbl>
#> 1 0.277
```



```
Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

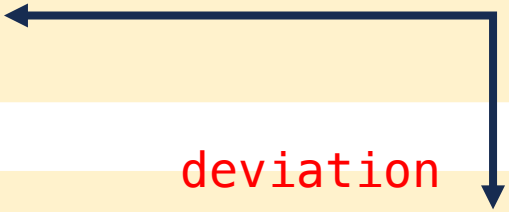
```
df_res <-
  df_xy %>%
  mutate(y_hat = alpha0 + alpha1 * x,
         res = y - y_hat)

df_res %>% summarise(SE = sqrt((sum(res^2) / (nrow(.) - 2))))

#> # A tibble: 1 × 1
#>      SE
#>   <dbl>
#> 1 0.928
```



```
Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

```
df_res <-
  df_xy %>%
  mutate(y_hat = alpha0 + alpha1 * x,
         res = y - y_hat)

df_res %>% summarise(SE = sqrt((sum(res^2) / (nrow(.) - 2))))

#> # A tibble: 1 × 1
#>      SE
#>   <dbl>
#> 1 0.928
```



deviation

```
Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

残差標準偏差
Residual standard deviation

$$RSD = \sqrt{\frac{1}{n-k-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

説明変数の次元（今は$k = 1$）

誤差uの標準偏差の期待値

残差標準偏差
Residual standard deviation

```
> ?stats::sigma
```

Description

Extract the estimated standard deviation of the errors, the “residual standard deviation” (misnamed also “residual standard error”, e.g., in summary.lm()'s output, from a fitted model).

回帰モデル regression

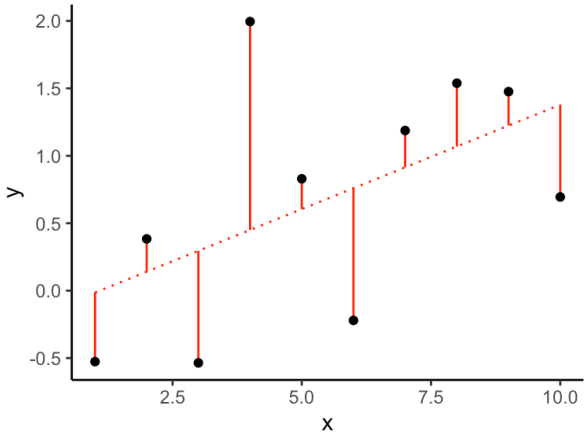
```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

                    deviation
Residual standard ~~error~~: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

                    deviation
Residual standard ~~error~~: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

3つの変動

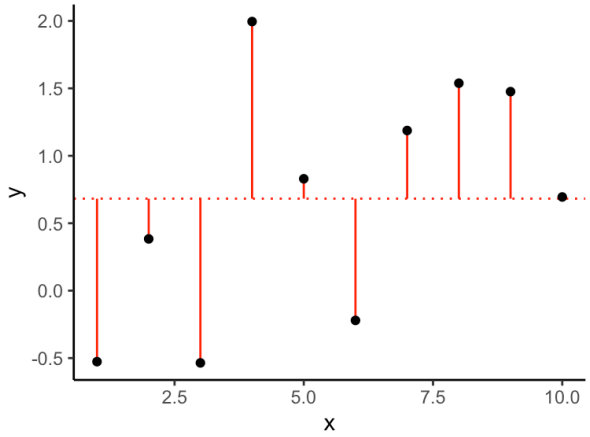
残差変動
RSS



$$RSS = \sum_{i=1}^{n} (y_i - \hat{y})^2$$

データ　推定値

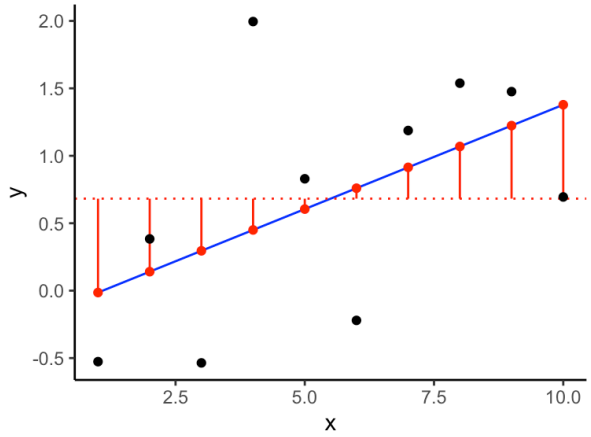
全変動
TSS



$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

データ　平均

回帰変動
ESS

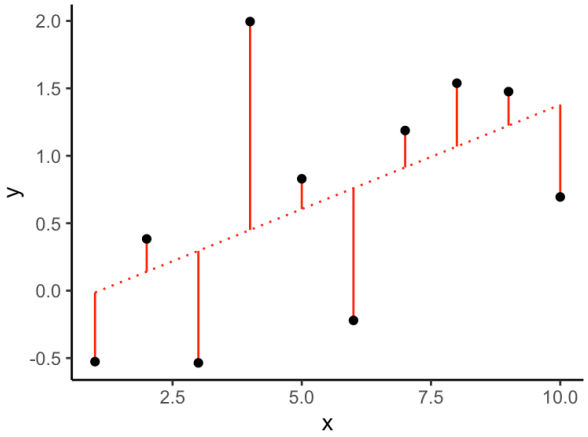


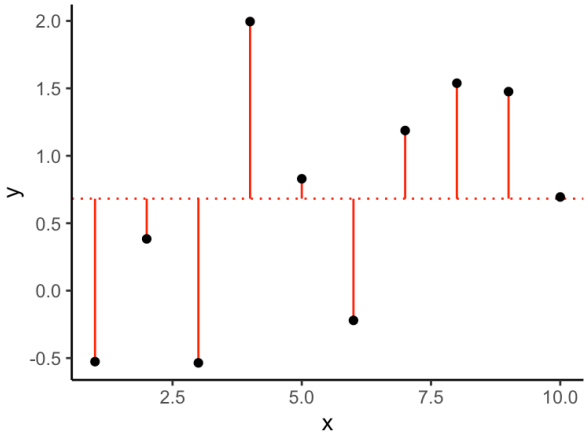
$$ESS = \sum_{i=1}^{n} (\hat{y} - \bar{y})^2$$

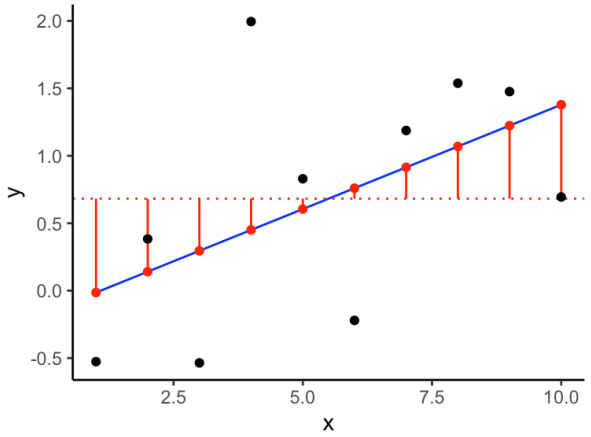
推定値　平均

3つの変動

残差変動 RSS = 全変動 TSS − 回帰変動 ESS







$$RSS = \sum_{i=1}^{n} (y_i - \hat{y})^2$$

データ　推定値

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

データ　平均

$$ESS = \sum_{i=1}^{n} (\hat{y} - \bar{y})^2$$

推定値　平均

決定係数 R-squared

回帰変動　　残差変動

$$\underset{\text{決定係数}}{R2} = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

全変動　　全変動

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y})^2$$

データ　推定値

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

データ　平均

$$ESS = \sum_{i=1}^{n} (\hat{y} - \bar{y})^2$$

推定値　平均

回帰モデル regression

```
df_res <-
  df_xy %>%
  mutate(y_hat = alpha0 + alpha1 * x,
         res = y - y_hat)

dat_dev <-
  df_res %>%
  mutate(yhat_mean = mean(y_hat),
         RSS = y - y_hat,
         ESS = y_hat - yhat_mean,
         TSS = y - yhat_mean) %>%
  summarize_at(c("RSS", "ESS", "TSS"),
               ~ sum(. ^ 2))
#> # A tibble: 1 × 3
#>     RSS   ESS   TSS
#>   <dbl> <dbl> <dbl>
#> 1  6.89  4.99  11.9
```

回帰モデル regression

```
df_res <-
  df_xy %>%
  mutate(y_hat = alpha0 + alpha1 * x,
         res = y - y_hat)

dat_dev <-
  df_res %>%
  mutate(yhat_mean = mean(y_hat),
         RSS = y - y_hat,
         ESS = y_hat - yhat_mean,
         TSS = y - yhat_mean) %>%
  summarize_at(c("RSS", "ESS", "TSS"),
               ~ sum(. ^ 2))
#> # A tibble: 1 × 3
#>     RSS   ESS   TSS
#>   <dbl> <dbl> <dbl>
#> 1  6.89  4.99  11.9

dat_dev %$% { ESS / TSS }
#> [1] 0.4202279
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

                    deviation
Residual standard ~~error~~: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

deviation

F分布

カイ2乗分布

互いに独立な$X_1 \sim \chi^2(m)$、$X_2 \sim \chi^2(n)$について、

$$Y = \frac{X_1/m}{X_2/n}$$

なるYの確率分布を**F分布**と呼び、

$$Y \sim F(m, n)$$

と表す。
（mを第1自由度，nを第2自由度と呼ぶ）

χ^2分布

標準正規分布$\mathcal{N}(0,1)$に従う独立なk個の確率変数$X_1, X_2, \ldots, X_k$について、

$$Z = \sum_{i=1}^{k} {X_i}^2$$

なる Z の確率分布を χ^2 分布と呼び、

$$Z \sim \chi^2(k)$$

と表す。(kを自由度と呼ぶ)

χ^2分布

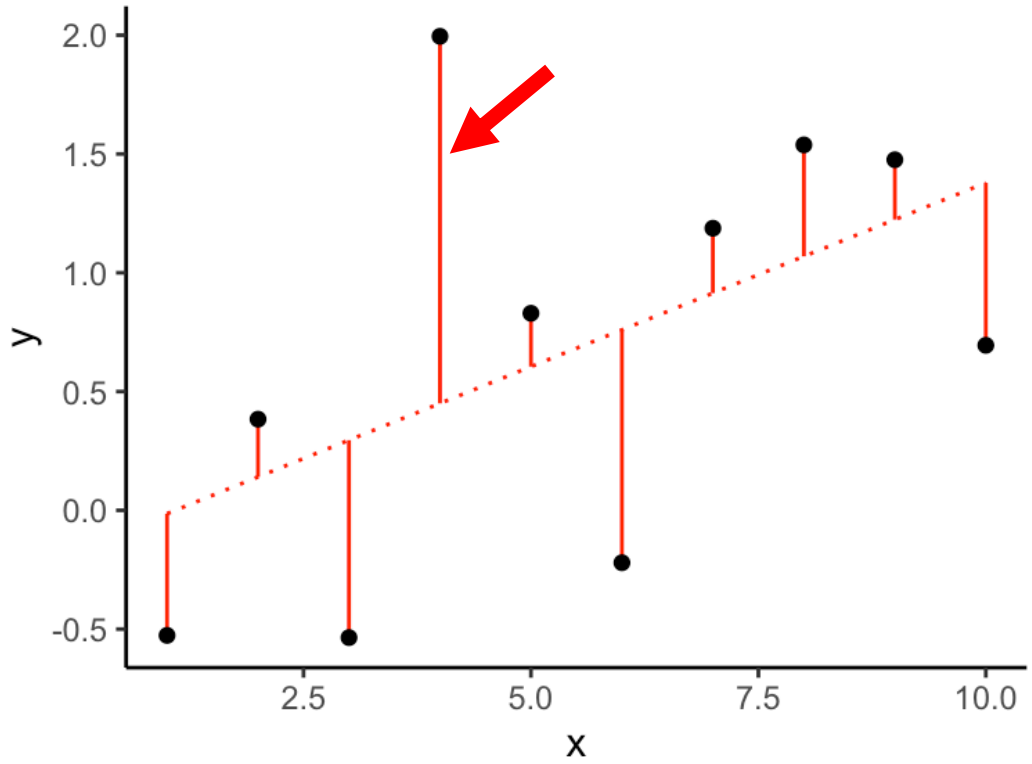
正規分布$\mathcal{N}(u, \sigma^2)$に従う独立なk個の確率変数$X_1, X_2, \ldots, X_k$について、

$$Z = \sum_{i=1}^{k} \frac{(X_i - u)^2}{\sigma^2}$$

なるZは自由度$k-1$のχ^2分布に従う

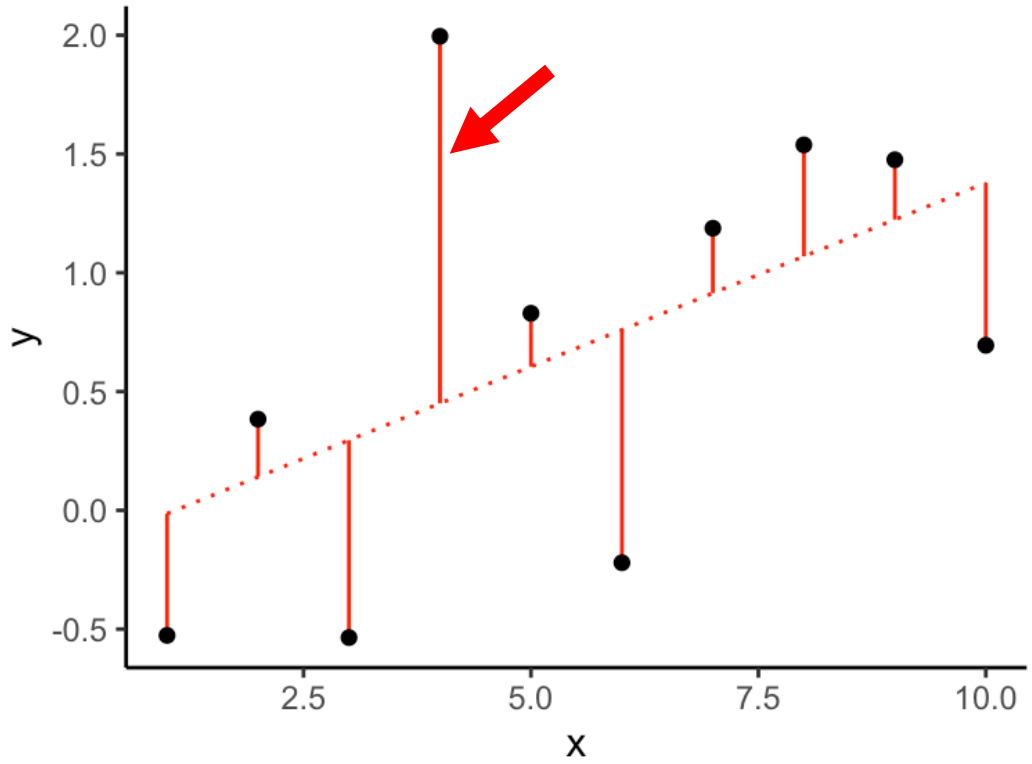
$$Z \sim \chi^2(k-1)$$

正規分布に従う独立なk個の確率変数
$X_1, X_2, \ldots, X_k$の2乗和、すなわちコレ！？



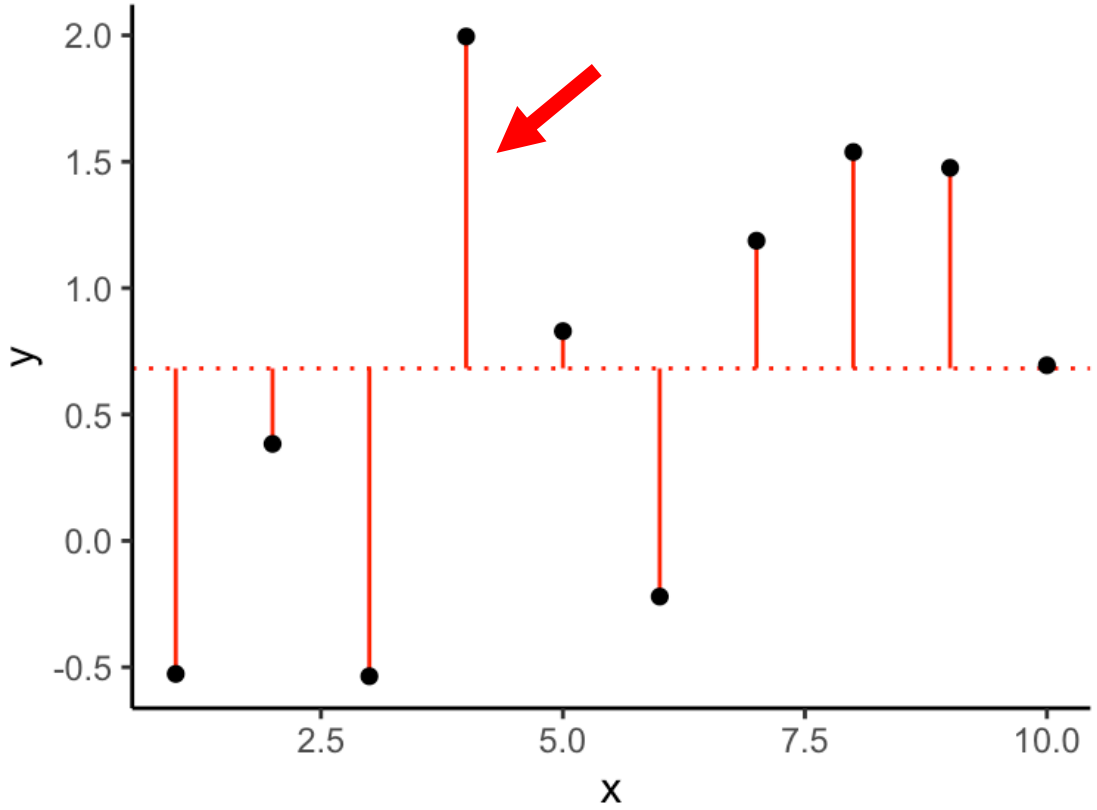
データ数kの時、残差二乗和は
自由度$k-1$のχ2分布に従う？

正規分布に従う独立なk個の確率変数
$X_1, X_2, \ldots, X_k$の2乗和、すなわちコレ！？



データ数kの時、残差二乗和は
自由度$k-1$のχ2分布に従う？　→　**従わない**

正規分布に従う独立なk個の確率変数
$X_1, X_2, \ldots, X_k$の2乗和、すなわちコレ！？



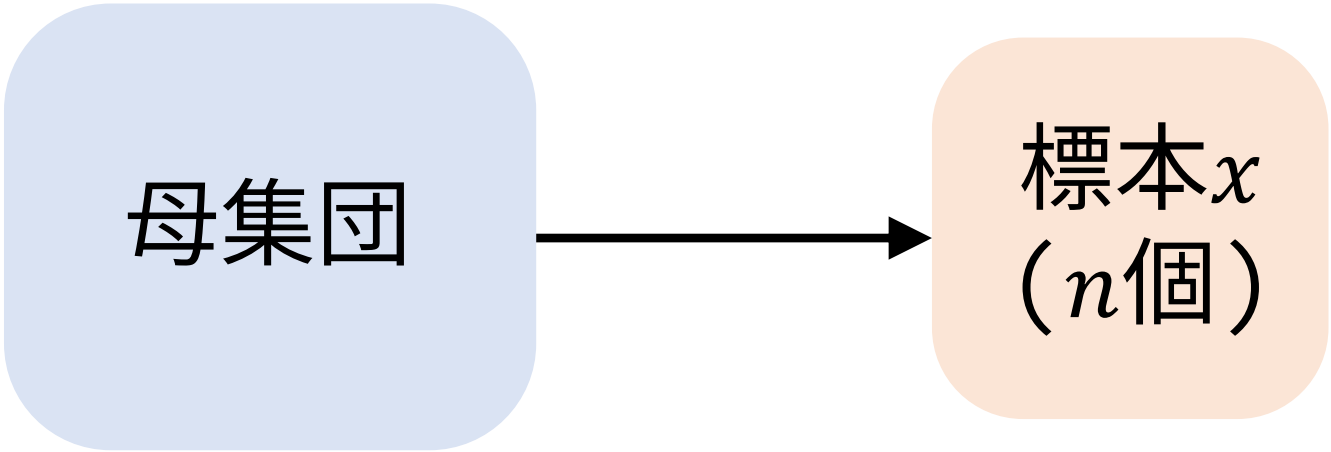
コレは自由度$k-1$のχ2分布に従う。

自由度 degrees of freedom

変数のうち自由な値をとりうるものの数

自由度 degrees of freedom

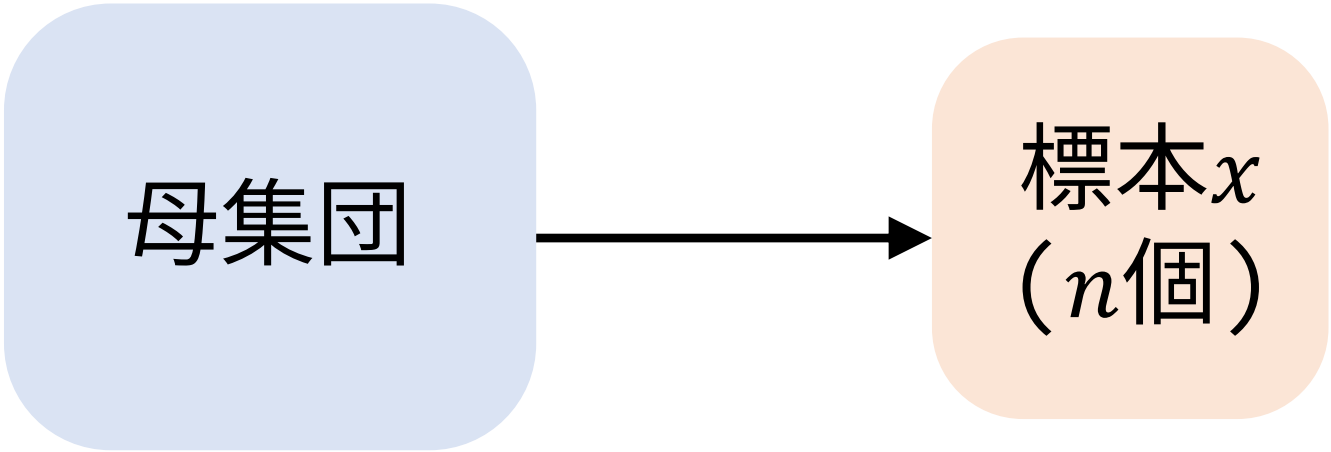
変数のうち自由な値をとりうるものの数



自由に選ばれたn個の標本の自由度は$\boldsymbol{n}$で、
標本の平均値は標本平均$\bar{x}$。

自由度 degrees of freedom

変数のうち自由な値をとりうるものの数

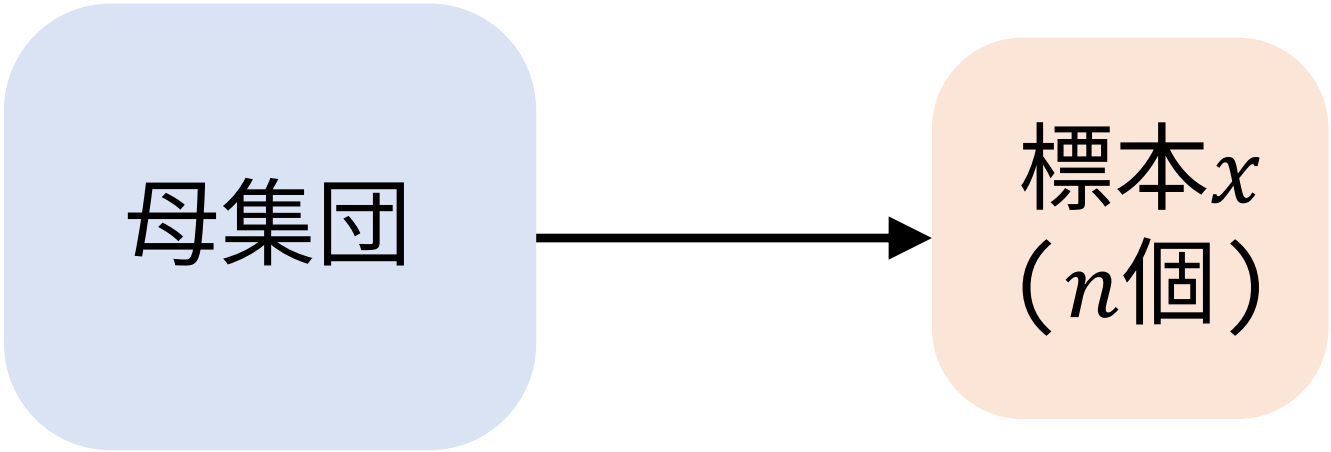


自由に選ばれたn個の標本の自由度は$\boldsymbol{n}$で、
標本の平均値は標本平均$\bar{x}$。

では、標本平均が既知の値$\bar{x}$になるように
選ばれたn個の標本の自由度は？

自由度 degrees of freedom

変数のうち自由な値をとりうるものの数



自由に選ばれたn個の標本の自由度は$\boldsymbol{n}$で、
標本の平均値は標本平均$\bar{x}$。

では、標本平均が既知の値$\bar{x}$になるように
選ばれたn個の標本の自由度は？　→ $\boldsymbol{n-1}$

自由度 degrees of freedom

変数のうち自由な値をとりうるものの数

標本平均が既知の値$\bar{x}$になるように
選ばれたn個の標本の自由度は $\boldsymbol{n-1}$

$$\bar{x} = \frac{1}{n}(x_1 + x_2 + \cdots + x_{n-1} + x_n)$$

$$x_n = n\bar{x} - (\underbrace{x_1 + x_2 + \cdots + x_{n-1}}_{\text{自由に値をとる}\boldsymbol{n-1}\text{項}})$$

既知の
標本平均

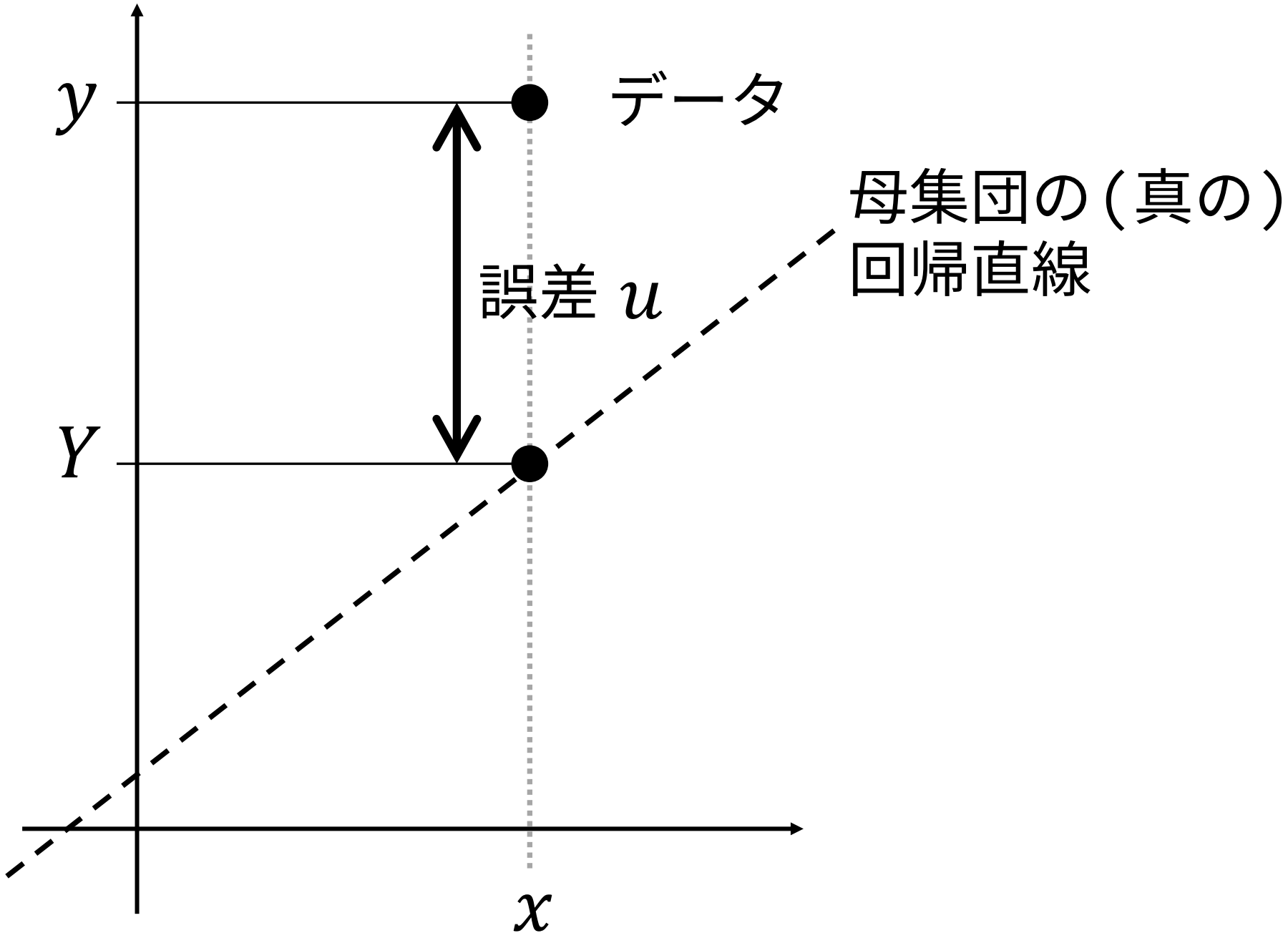
ここまでおさえておいて、

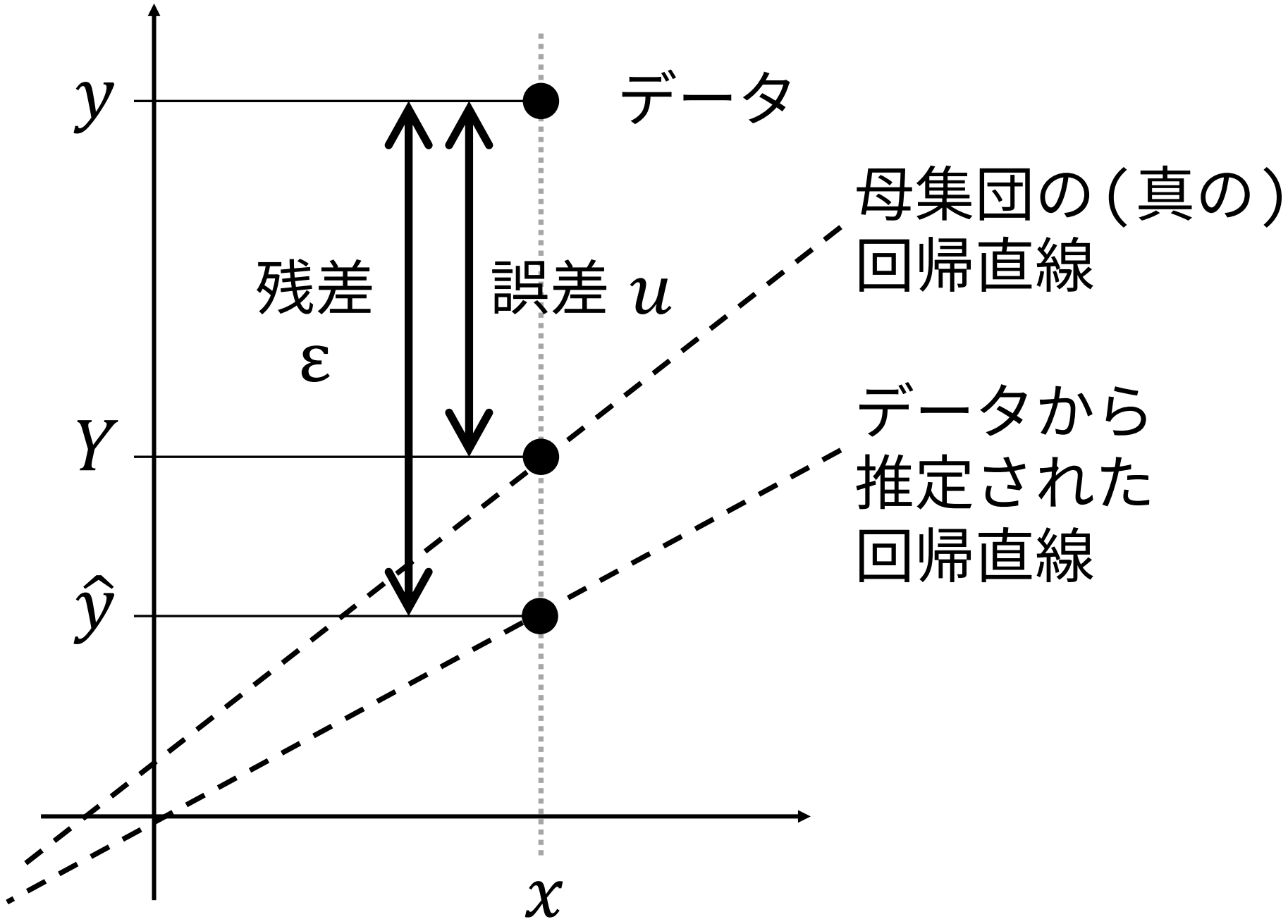
回帰モデルの考え方を再確認。

(単)回帰モデル

$$y = a_0 + a_1 x + u$$

$$u \sim_{i.i.d.} \mathcal{N}(0, \sigma^2)$$


y
データ
母集団の(真の)
回帰直線
誤差 u
Y
x


y
データ
残差
ε
誤差 u
母集団の(真の)
回帰直線
Y
データから
推定された
回帰直線
$\hat{y}$
x

真の回帰モデル

誤差

$$y = \alpha_0 + \alpha_1 x + u$$

$$u \sim_{i.i.d.} \mathcal{N}(0, \sigma^2)$$

最小二乗法で推定された回帰式

$$y = \hat{a}_0 + \hat{a}_1 x + \varepsilon$$

残差

$$\varepsilon = y - \hat{y}$$

(正規)回帰モデル5つの仮定

1. **誤差**の期待値はゼロ
   $E[u] = 0$
2. **誤差**の分散は定数
   $Var[u] = \sigma^2$
3. **誤差**は互いに無相関
   $E[u_i u_j] = 0 \quad (i \neq j)$
4. **誤差**と説明変数は無相関
   $E[u_i(x_i - E[x_i])] = 0$
5. **誤差**は正規分布に従う
   $u \sim_{i.i.d.} \mathcal{N}(0, \sigma^2)$

(正規)回帰モデル5つの仮定

1．誤差の期待値はゼロ
2．誤差の分散は定数
3．誤差は互いに無相関
4．誤差と説明変数は無相関
5．誤差は正規分布に従う

ガウス=マルコフの定理
仮定1-4が満たされる場合、最小二乗推定量は線型不偏推定量の中で最小分散を持つ。

+ 仮定5
最小二乗推定量は最小分散を持つ。

真の回帰モデル

誤差

$$y = \alpha_0 + \alpha_1 x + u$$

$$u \sim_{i.i.d.} \mathcal{N}(0, \sigma^2)$$

最小二乗法OLSでモデルの
パラメータを推定する
(OLS, Ordinary Last Squares)

→ **残差二乗和RSSの最小化**
(RSS, Residual Sum of Squares)

OLSが要請する残差の性質

１．残差の合計値はゼロ

$$\sum_{i=1}^{n} \varepsilon_i = 0$$

２．残差と説明変数は無相関

$$\sum_{i=1}^{n} x_i \varepsilon_i = 0$$

OLSが要請する残差の性質

１．残差の合計値はゼロ

$$\sum_{i=1}^{n} \varepsilon_i = 0$$

２．残差と説明変数は無相関

$$\sum_{i=1}^{n} x_{ki}\varepsilon_i = 0 \quad (k = 1, 2, \ldots)$$

説明変数の次元数

OLSで推定された(重)回帰式

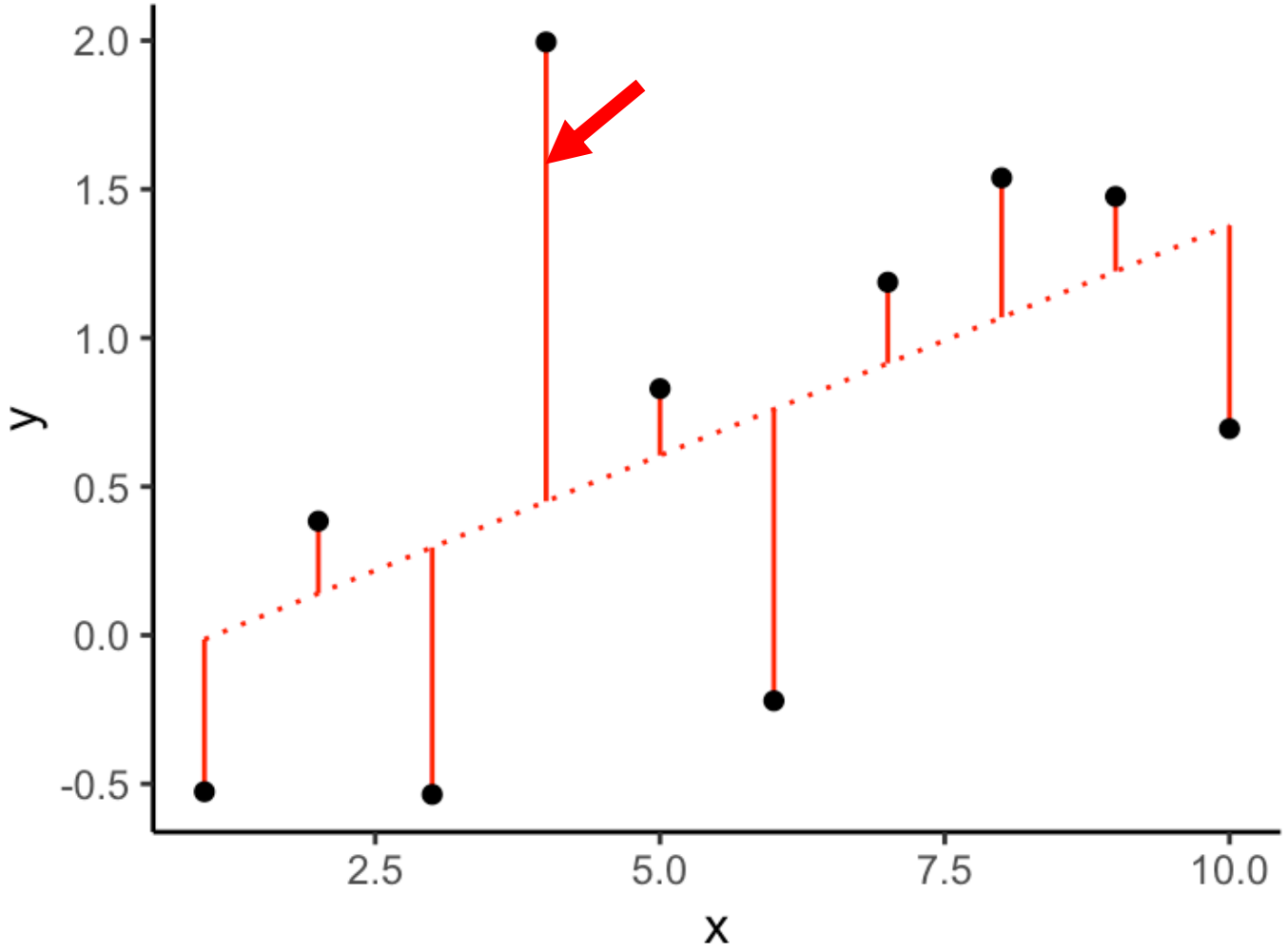
$$y = \hat{a}_0 + \underbrace{\hat{a}_1 x_1 + \cdots + \hat{a}_k x_k}_{k\text{項}} + \varepsilon$$

残差

$$\sum_{i=1}^{n} \varepsilon_i = 0 \quad \text{かつ} \quad \sum_{i=1}^{n} x_{ki}\varepsilon_i = 0 \quad (k = 1, 2, \ldots)$$

n個の残差εは、
$k+1$個の式を満たさなければならない。

残差εの自由度は$n - k - 1$となる。



残差εの自由度は$n - k - 1$となる。

残差標準偏差
Residual standard deviation

$$RSD = \sqrt{\frac{1}{n-k-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

説明変数の次元（今は$k = 1$）

$$y = a_0 + a_1 x + u$$

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

deviation

n - k - 1

OLSが要請する残差の性質

１．残差の合計値はゼロ

$$\sum_{i=1}^{n} \varepsilon_i = 0$$

※ 定数項$\hat{a}_0$が無い回帰式では
この性質は満たされない。
→ 残差εの自由度は$n - k$。

２．残差と説明変数は無相関

$$\sum_{i=1}^{n} x_{ki}\varepsilon_i = 0 \quad (k = 1, 2, \ldots)$$

説明変数の次元数

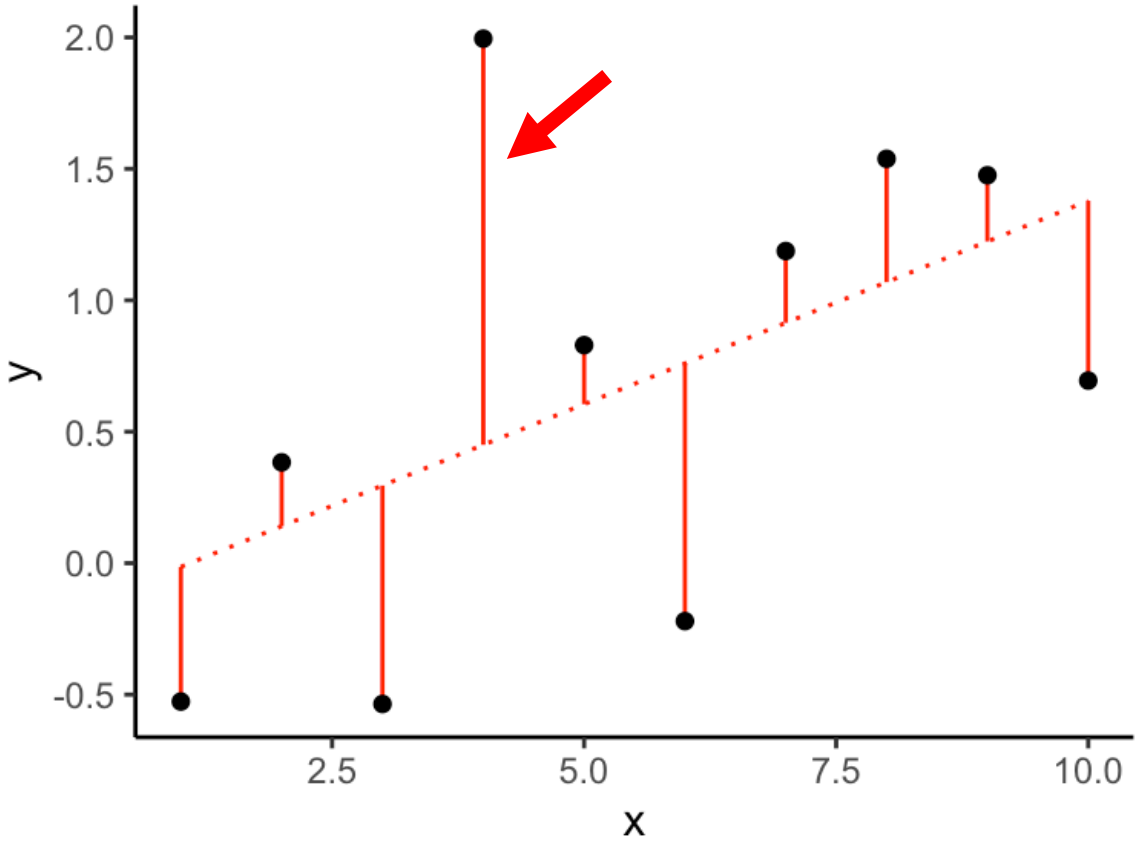
χ^2分布

正規分布に従う独立なk個の確率変数
$X_1, X_2, \ldots, X_k$について、

$$Z = \sum_{i=1}^{k} \frac{(X_i - \bar{X})^2}{\sigma^2}$$

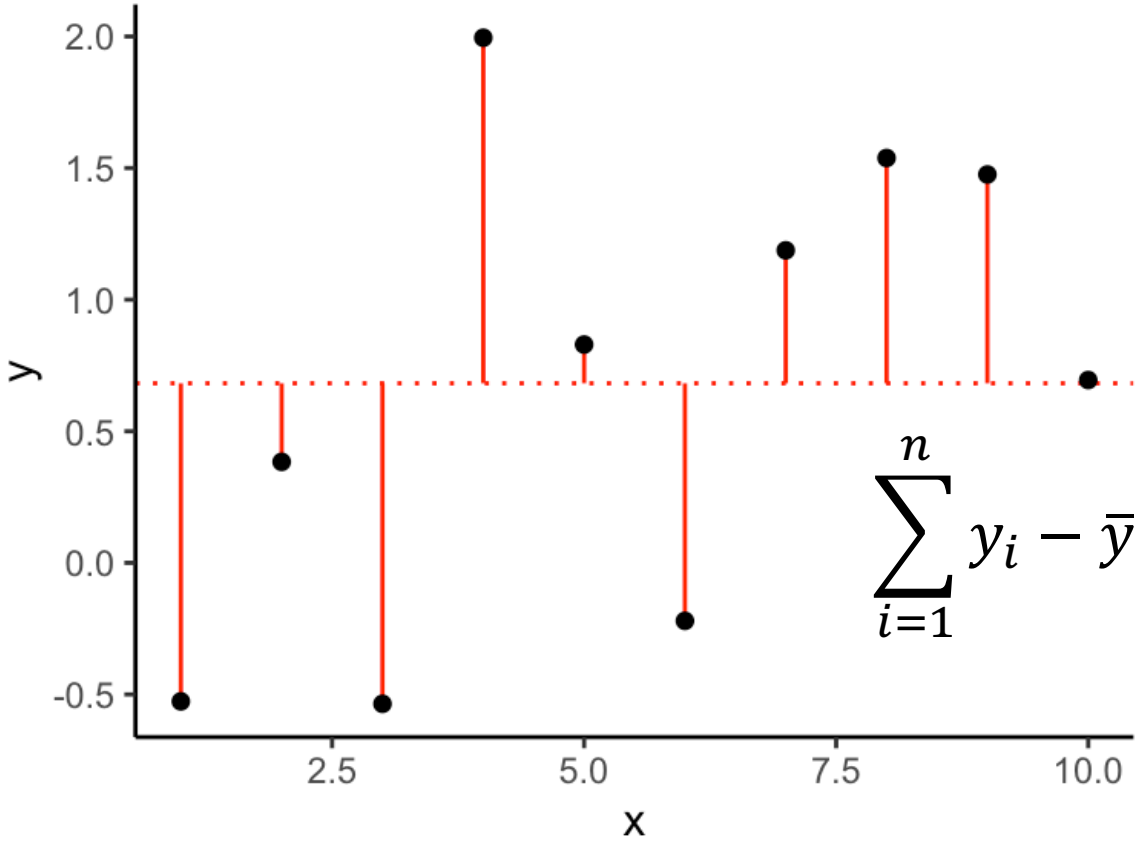
なるZは自由度$k-1$のχ^2分布に従う

$$Z \sim \chi^2(k-1)$$



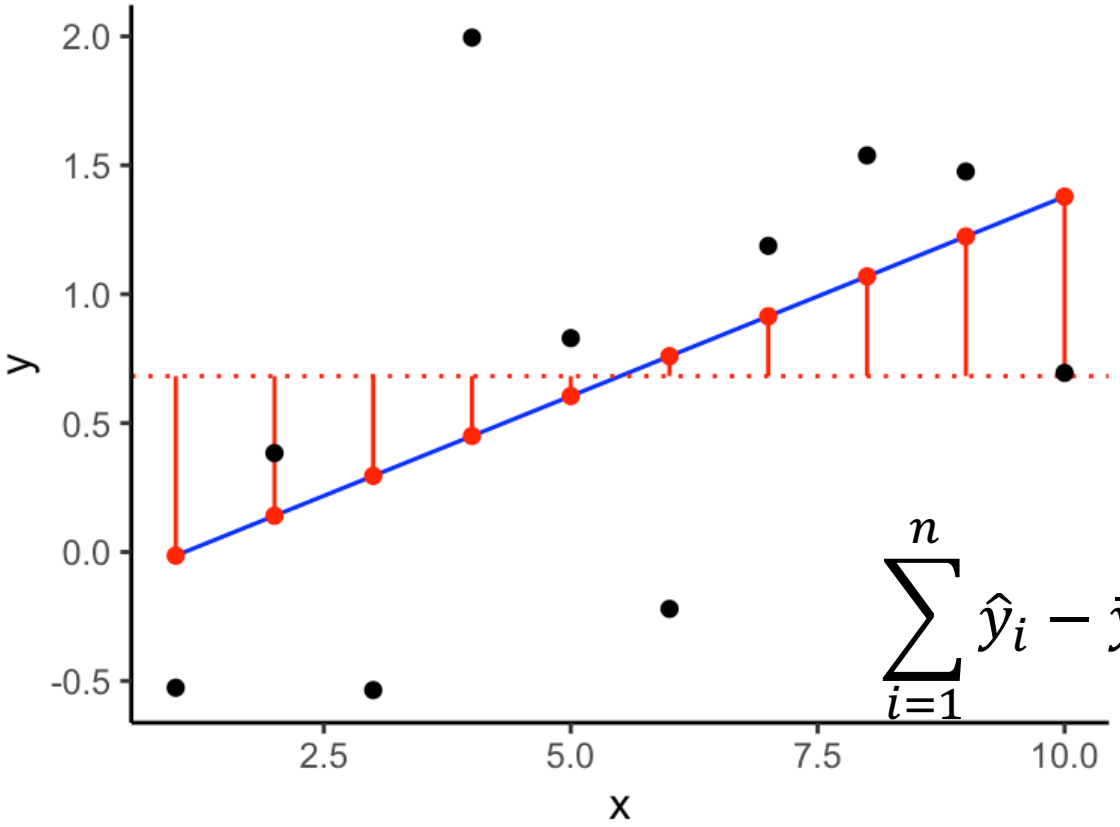
データ数nの時、残差平方和は(理想的には)自由度$n-k-1$のχ^2分布に従う。

$$RSS \sim \chi^2(n-k-1)$$



データ数nの時、全変動は(理想的には)自由度$n-1$のχ^2分布に従う。

$$TSS \sim \chi^2(n-1)$$



$$\sum_{i=1}^{n} \hat{y}_i - \bar{y} = 0$$

回帰変動は(理想的には)
自由度kのχ^2分布に従う。

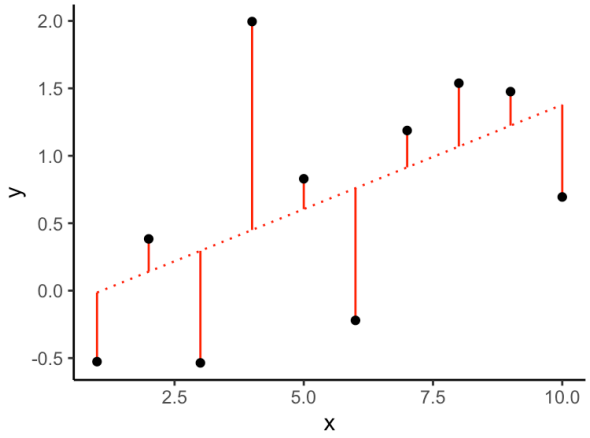
$$ESS = TSS - RSS$$

$\chi^2(k)$　　$\chi^2(n-1)$　　$\chi^2(n-k-1)$

これらが互いに独立なので、

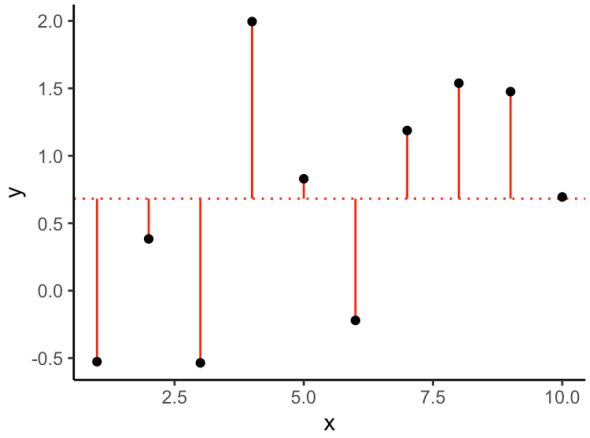
OLSの変動の確率分布

残差変動



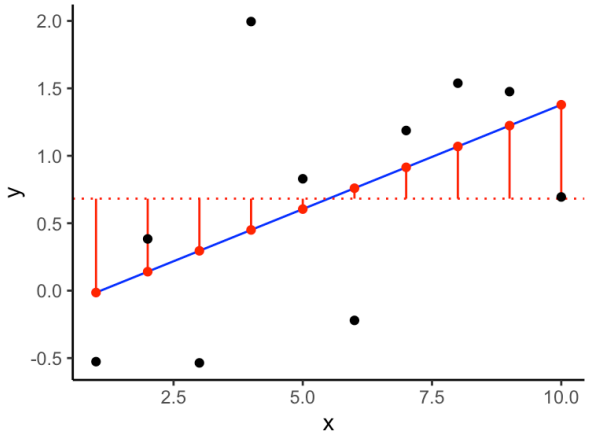
$RSS \sim \chi^2(n-k-1)$

全変動



$TSS \sim \chi^2(n-1)$

回帰変動



$ESS \sim \chi^2(k)$

回帰モデルの適合性

決定係数

$$R2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

自由度調整済み決定係数

$$\overline{R2} = 1 - \frac{RSS/(n-k-1)}{TSS/(n-1)}$$

回帰モデル regression

```
dat_dev
#> # A tibble: 1 × 3
#>     RSS   ESS   TSS
#>   <dbl> <dbl> <dbl>
#> 1  6.89  4.99  11.9

dat_dev %$% { ESS / TSS }
#> [1] 0.4202279

dat_dev %$%
  {1 - RSS * (N - 1) / TSS / (N - 1 - 1)}
#> [1] 0.3477564
```

```
Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202, Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

deviation

F分布

カイ2乗分布

互いに独立な$X_1 \sim \chi^2(m)$、$X_2 \sim \chi^2(n)$について、

$$Y = \frac{X_1/m}{X_2/n}$$

なるYの確率分布を**F分布**と呼び、

$$Y \sim F(m, n)$$

と表す。
（mを第1自由度，nを第2自由度と呼ぶ）

自由度とf検定

無相関(傾き=0)なら0

F値

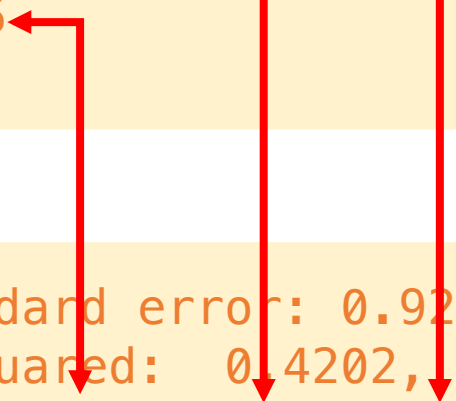
$$f = \frac{ESS/k}{RSS/(n-k-1)}$$

分散0なら0

$$f \sim F(k, n-k-1)$$

回帰モデル regression

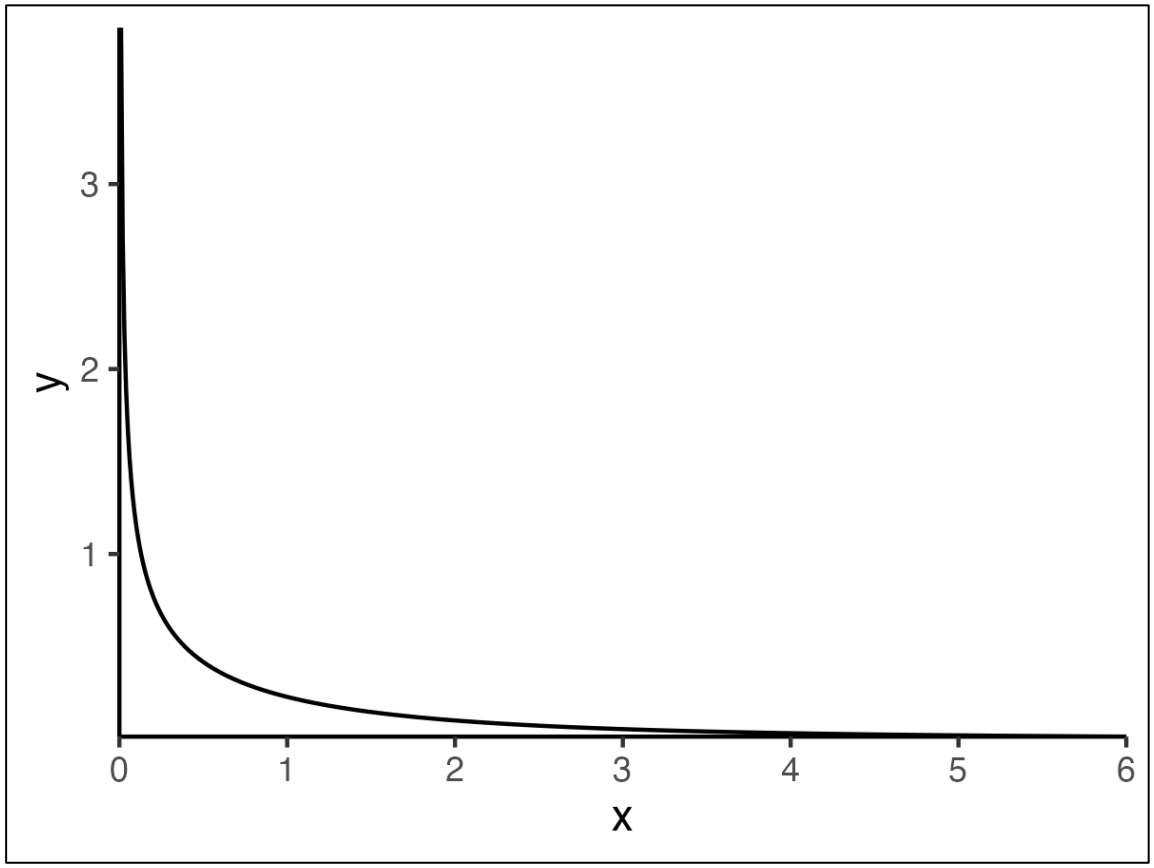
```
dat_dev
#> # A tibble: 1 × 3
#>     RSS   ESS   TSS
#>   <dbl> <dbl> <dbl>
#> 1  6.89  4.99  11.9

df1 <- 1
df2 <- N - df1 - 1

fval <- dat_dev %$% { ESS * df2 / RSS / df1 }
#> [1] 5.798525
```



```
Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,    Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

F分布

$Y \sim F(1,8)$の確率密度



確率分布を取り扱うための関数

	$f(x)$	$F(x)$	$F^{-1}(x)$	$\mathrm{f}(x)$
	確率分布に従う乱数	**累積確率**	**確率点**	**確率密度**
	r*()**	**p***()**	**q***()**	**d***()**
正規分布	rnorm()	pnorm()	qnorm()	dnorm()
一様分布	runif()	punif()	qunif()	dunif()
f分布	rf()	pf()	qf()	df()
t分布	rt()	pt()	qt()	dt()
カイ²分布	rchisq()	pchisq()	qchisq()	dchisq()
⋮				

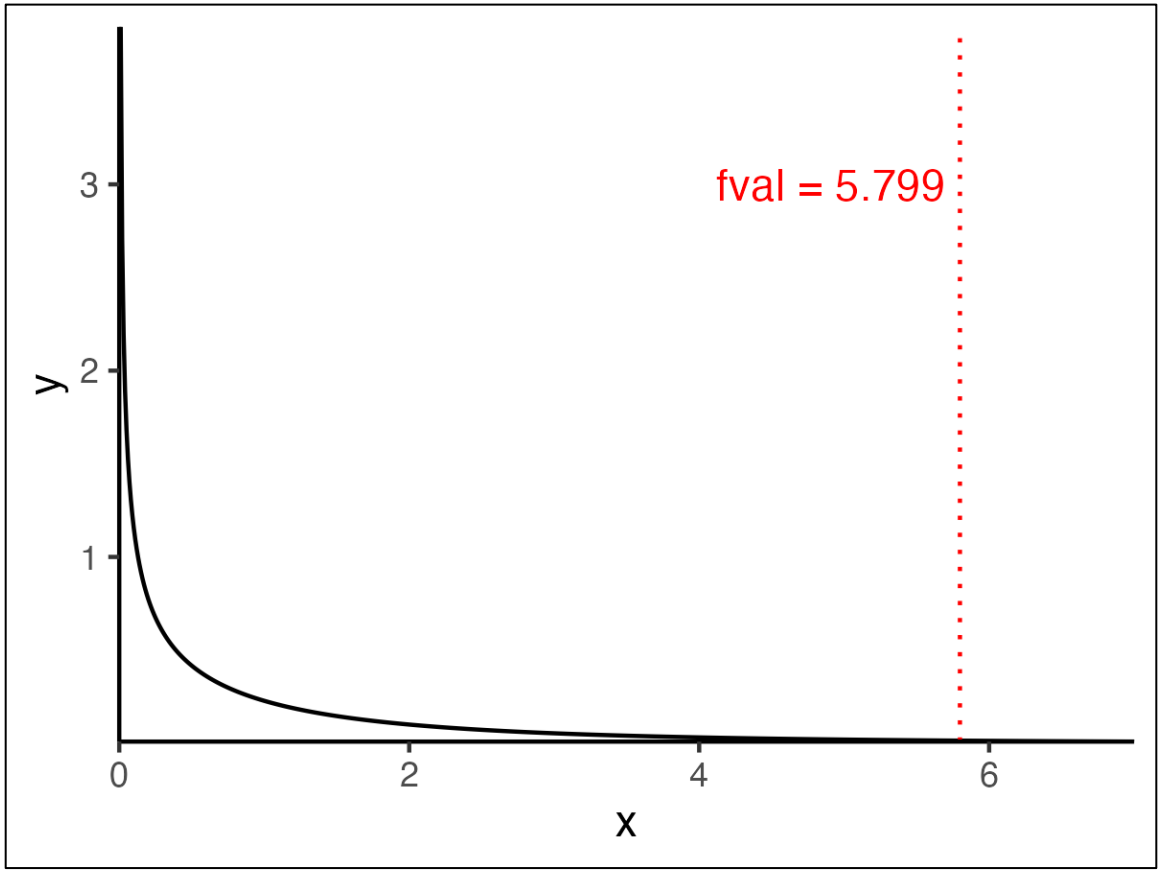
F分布

```
df_f <-
  tibble(
    x = seq(0, 6, by = 0.01),
    y = df(x, df1 = 1, df2 = 8)
  )

ggplot(data = df_f) +
  aes(x = x, y = y) +
  geom_path() +
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0))
```

F分布

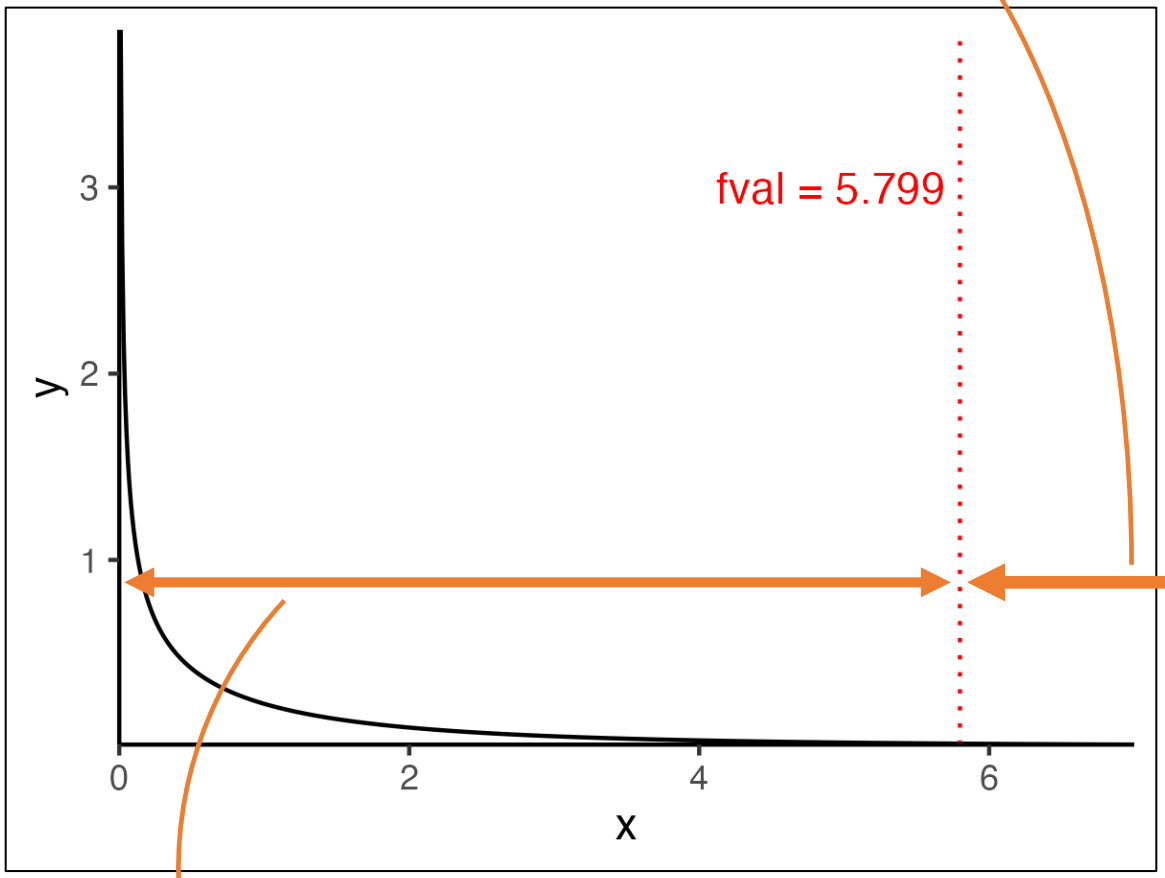
```
df_f <-
  tibble(
    x = seq(0, 6, by = 0.01),
    y = df(x, df1 = 1, df2 = 8)
  )

ggplot(data = df_f) +
  aes(x = x, y = y) +
  geom_path() +
  geom_vline(xintercept = fval,
             linetype = "dotted",
             color = "red") +
  geom_text(data = data.frame(x = fval - 0.1,
                              y = 3),
            label = str_c("fval = ", round(fval, 3)),
            hjust = 1,
            color = "red")+
  scale_x_continuous(expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0))
```

F分布



F分布

```
pf(q = fval, df1 = 1, df2 = 8,
   lower.tail = FALSE)
#> [1] 0.04264022
```



p値は
こっち側の
累積確率

```
pf(q = fval, df1 = 1, df2 = 8,
   lower.tail = FALSE)
#> [1] 0.9573598
```

回帰モデル regression

```
dat_dev
#> # A tibble: 1 × 3
#>     RSS   ESS   TSS
#>   <dbl> <dbl> <dbl>
#> 1  6.89  4.99  11.9

df1 <- 1
df2 <- N - df1 - 1

fval <- dat_dev %$% { ESS * df2 / RSS / df1 }
#> [1] 5.798525

pf(q = fval, df1 = df1, df2 = df2, lower.tail = FALSE)
#> [1] 0.04264022
```

```
Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% names()
#>  [1] "coefficients"  "residuals"     "effects"
#>  [4] "rank"          "fitted.values" "assign"
#>  [7] "qr"            "df.residual"   "xlevels"
#> [10] "call"          "terms"         "model”

lm(y ~ x, data = df_xy) %>% summary() %>% names()
#>  [1] "call"          "terms"         "residuals"
#>  [4] "coefficients"  "aliased"       "sigma"
#>  [7] "df"            "r.squared"     "adj.r.squared"
#> [10] "fstatistic"    "cov.unscaled"
```

```
Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

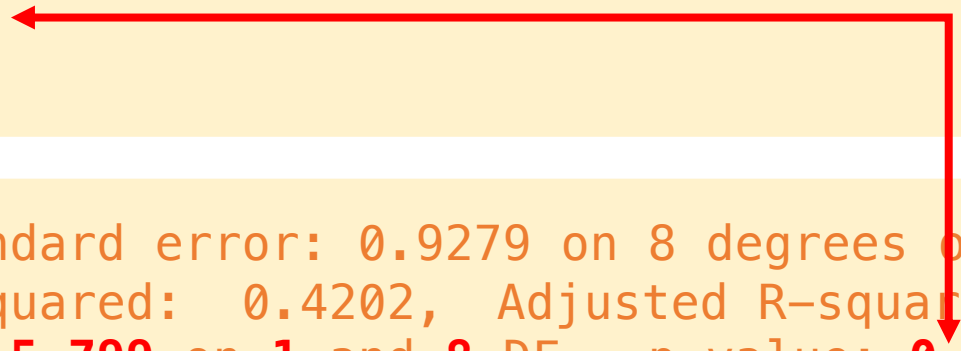
回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary() %>% names()
#>  [1] "call"          "terms"         "residuals"
#>  [4] "coefficients"  "aliased"       "sigma"
#>  [7] "df"            "r.squared"     "adj.r.squared"
#> [10] "fstatistic"    "cov.unscaled"

lm(y ~ x, data = df_xy) %>%
  summary() %>%
  .$fstatistics
#>    value    numdf    dendf
#> 5.798525 1.000000 8.000000
```

```
Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary() %>% names()
#>  [1] "call"          "terms"         "residuals"
#>  [4] "coefficients"  "aliased"       "sigma"
#>  [7] "df"            "r.squared"     "adj.r.squared"
#> [10] "fstatistic"    "cov.unscaled"

lm(y ~ x, data = df_xy) %>%
  summary() %>%
  .$fstatistic %>% {
  pf(q = .[1], df1 = .[2], df2 = .[3], lower.tail = FALSE)
  }
#> value
#> 0.04264022
```



```
Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

deviation

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

deviation

最小二乗法（線形回帰）

$$y = \hat{a}_0 + \hat{a}_1 x + \varepsilon$$

回帰係数$\hat{a}_0$と$\hat{a}_1$の仮説検定を行う。

すなわち例えば$\hat{a}_1$について、

帰無仮説：$\hat{a}_1 = 0$

対立仮説：$\hat{a}_1 \neq 0$

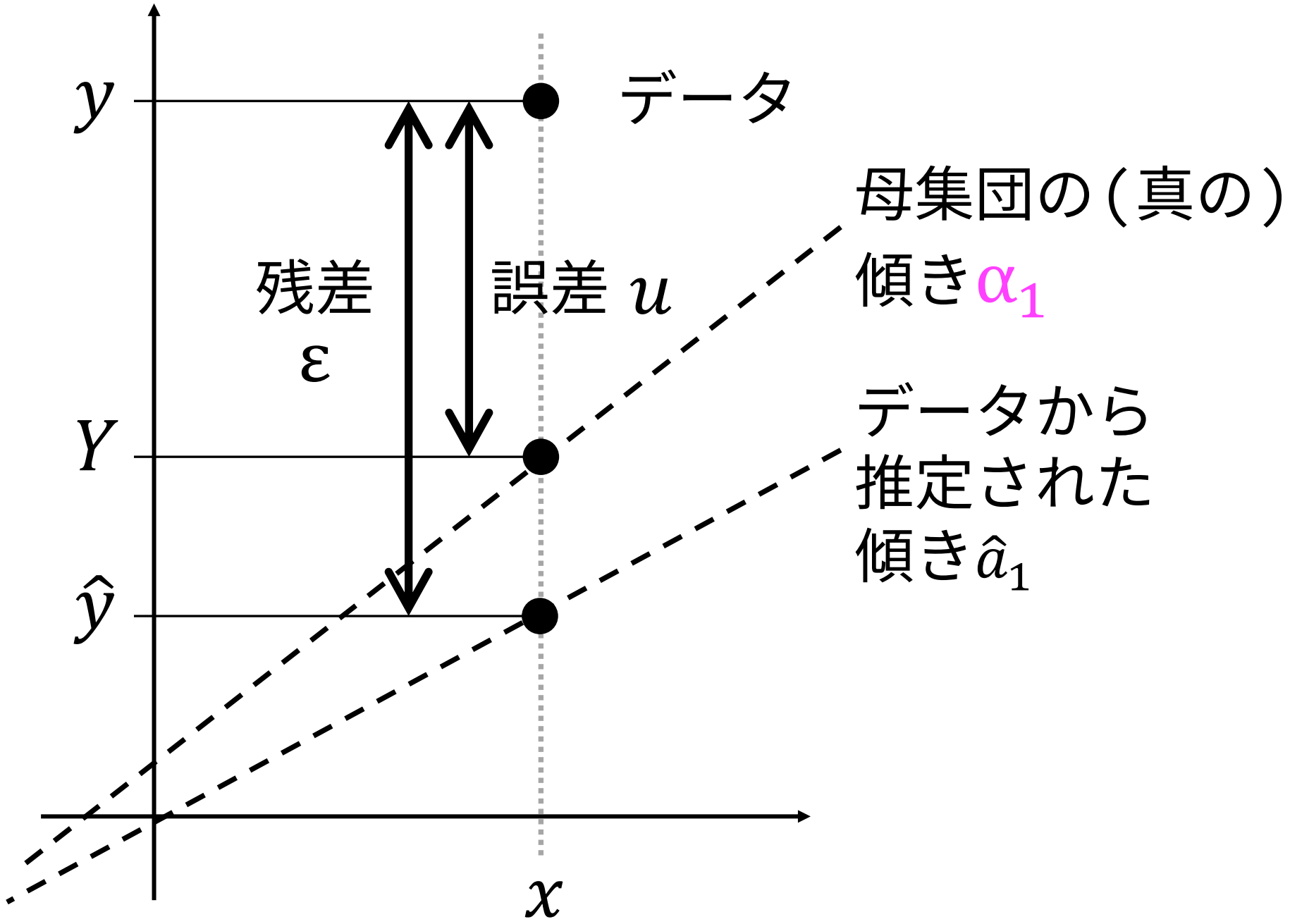
→ $\hat{a}_0$と$\hat{a}_1$の確率分布を知りたい。

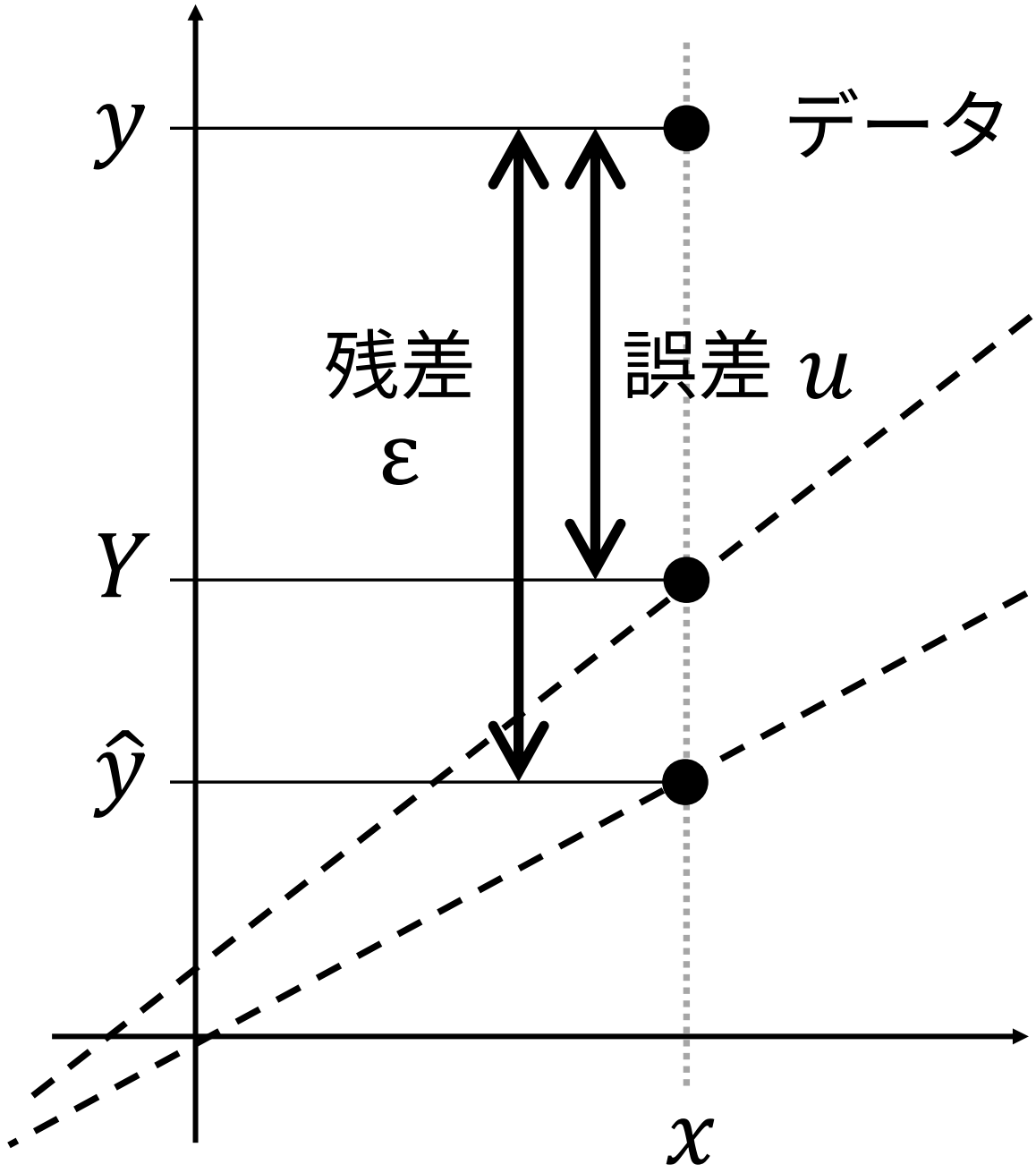
→ $\hat{a}_0$と$\hat{a}_1$の期待値と分散を調べる。

最小二乗法（線形回帰）

$$y = \hat{a}_0 + \hat{a}_1 x + \varepsilon$$

回帰係数$\hat{a}_0$と$\hat{a}_1$の仮説検定を行う。


y
データ
残差
ε
誤差 u
母集団の（真の）
傾きα_1
Y
データから
推定された
傾き$\hat{a}_1$
$\hat{y}$
x



母集団の（真の）
傾きα_1

データから
推定された
傾き$\hat{a}_1$

確率変数

最小二乗法（線形回帰）

$$y = \hat{a}_0 + \hat{a}_1 x + \varepsilon$$

回帰係数$\hat{a}_1$の仮説検定を行う。

真の傾きα_1がゼロだった時、
回帰係数$\hat{a}_1$が得られる確率。

最小二乗法（線形回帰）

$$y = \hat{a}_0 + \hat{a}_1 x + \varepsilon$$

回帰係数$\hat{a}_1$の仮説検定を行う。

真の傾きα_1がゼロだった時、
回帰係数$\hat{a}_1$が得られる確率。

確率密度？
累積確率？

回帰係数のt検定

$$\hat{a}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \frac{\sum_{i=1}^{n}(x_i - \bar{x})((\alpha_0 + \alpha_1 x_i + u_i) - (\alpha_0 + \alpha_1 \bar{x} + \bar{u}))}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

回帰係数のt検定

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \frac{\sum_{i=1}^{n}(x_i - \bar{x})((\alpha_0 + \alpha_1 x_i + u_i) - (\alpha_0 + \alpha_1 \bar{x} + \bar{u}))}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(\alpha_1 (x_i - \bar{x}) + u_i - \bar{u})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

回帰係数のt検定

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \frac{\sum_{i=1}^{n}(x_i - \bar{x})((\alpha_0 + \alpha_1 x_i + u_i) - (\alpha_0 + \alpha_1 \bar{x} + \bar{u}))}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(\alpha_1 (x_i - \bar{x}) + u_i - \bar{u})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \alpha_1 + \frac{\sum_{i=1}^{n}(x_i - \bar{x})u_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2} + \frac{\sum_{i=1}^{n}(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\bar{u} \quad \cdots \text{式(1)}$$

(正規)回帰モデル5つの仮定

1. **誤差**の期待値はゼロ
   $E[u] = 0$
2. **誤差**の分散は定数
   $Var[u] = \sigma^2$
3. **誤差**は互いに無相関
   $E[u_i u_j] = 0 \quad (i \neq j)$
4. **誤差**と説明変数は無相関
   $E[u_i(x_i - E[x_i])] = 0$
5. **誤差**は正規分布に従う
   $u \sim_{i.i.d.} \mathcal{N}(0, \sigma^2)$

回帰係数のt検定

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \frac{\sum_{i=1}^{n}(x_i - \bar{x})((\alpha_0 + \alpha_1 x_i + u_i) - (\alpha_0 + \alpha_1 \bar{x} + \bar{u}))}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(\alpha_1 (x_i - \bar{x}) + u_i - \bar{u})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$= \alpha_1 + \frac{\sum_{i=1}^{n}(x_i - \bar{x})u_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2} + \frac{\sum_{i=1}^{n}(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\bar{u} \quad \cdots \text{式(1)}$$

期待値ゼロ　　期待値ゼロ

従って、$E[\hat{a}_1] = \alpha_1$

最小二乗法

$$\hat{y}_i = a_0 + a_1 x_i,$$
$$\varepsilon_i = y_i - \hat{y}_i$$

$$\xrightarrow{\text{argmin}_{(a_0,a_1)} \sum_{i=1}^{n} \varepsilon^2}$$

$$\hat{a}_1 = \frac{S_{xy}}{S_{xx}},$$
$$\hat{a}_0 = \bar{y} - \frac{S_{xy}}{S_{xx}} \bar{x}$$

線形回帰モデル

$$Y_i = \alpha_0 + \alpha_1 x_i + u_i,$$
$$u_i \sim_{i.i.d.} N(0, \sigma^2)$$

$$\alpha_0 = E[\hat{a}_0] = \hat{a}_0,$$
$$\alpha_1 = E[\hat{a}_1] = \hat{a}_1$$

↑

期待値

あとでやります

回帰係数のt検定

$Var[\hat{a}_1]$

回帰係数のt検定

$Var[\hat{a}_1]$

恐怖の空白．．．

期待と分散の基本的性質

$$E[X+Y] = E[X] + E[Y]$$

$$E[cX] = cE[X]$$

XとYが独立の時、$E[XY] = E[X]E[Y]$

$$Var[X] = E[(X - E[X])^2] = E[X^2] - (E[X])^2$$

$$Var[X+Y] = Var[X] + Var[Y] + 2Cov[X,Y]$$

$$Var[cX] = c^2 Var[X]$$

$$Var[X+c] = Var[X]$$

期待と分散の基本的性質

$$E[X+Y] = E[X] + E[Y]$$

$$E[cX] = cE[X]$$

XとYが独立の時、$E[XY] = E[X]E[Y]$

$$\boldsymbol{Var}[\boldsymbol{X}] = \boldsymbol{E}\left[(\boldsymbol{X} - \boldsymbol{E}[\boldsymbol{X}])^{\boldsymbol{2}}\right] = E[X^2] - (E[X])^2$$

$$Var[X+Y] = Var[X] + Var[Y] + 2Cov[X,Y]$$

$$Var[cX] = c^2 Var[X]$$

$$Var[X+c] = Var[X]$$

回帰係数のt検定

$$Var[\hat{a}_1] = E[(\hat{a}_1 - E[\hat{a}_1])^2] = E[(\hat{a}_1 - \alpha_1)^2]$$

回帰係数のt検定

$$Var[\hat{a}_1] = E[(\hat{a}_1 - E[\hat{a}_1])^2] = E[(\hat{a}_1 - \alpha_1)^2]$$

$$= E\left[\left(\frac{\sum_{i=1}^{n}(x_i - \bar{x})u_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^2\right]$$

式(1)

$$\hat{a}_1 = \alpha_1 + \frac{\sum_{i=1}^{n}(x_i - \bar{x})u_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2} + \frac{\sum_{i=1}^{n}(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\bar{u}$$

初項は消えてくれる

誤差平均値$\bar{u}$の期待値はゼロ

回帰係数のt検定

$$Var[\hat{a}_1] = E[(\hat{a}_1 - E[\hat{a}_1])^2] = E[(\hat{a}_1 - \alpha_1)^2]$$

$$= E\left[\left(\frac{\sum_{i=1}^{n}(x_i - \bar{x})u_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^2\right]$$

$i \neq j$

$$= E\left[\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2 {u_i}^2}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2}\right] + E\left[\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - \bar{x})(x_j - \bar{x})u_i u_j}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2}\right]$$

回帰係数のt検定

$$Var[\hat{a}_1] = E[(\hat{a}_1 - E[\hat{a}_1])^2] = E[(\hat{a}_1 - \alpha_1)^2]$$

$$= E\left[\left(\frac{\sum_{i=1}^{n}(x_i - \bar{x})u_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^2\right]$$

$i \neq j$

$$= E\left[\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2 {u_i}^2}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2}\right] + E\left[\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - \bar{x})(x_j - \bar{x})u_i u_j}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2}\right]$$

$$= \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2 E[{u_i}^2]}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2} + \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - \bar{x})(x_j - \bar{x})E[u_i u_j]}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2}$$

回帰係数のt検定

$$Var[\hat{a}_1] = E[(\hat{a}_1 - E[\hat{a}_1])^2] = E[(\hat{a}_1 - \alpha_1)^2]$$

$$= E\left[\left(\frac{\sum_{i=1}^{n}(x_i - \bar{x})u_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^2\right]$$

$i \neq j$

$$= E\left[\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2 u_i{}^2}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2}\right] + E\left[\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - \bar{x})(x_j - \bar{x})u_i u_j}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2}\right]$$

$$= \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2 E[u_i{}^2]}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2} + \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - \bar{x})(x_j - \bar{x})E[u_i u_j]}{(\sum_{i=1}^{n}(x_i - \bar{x})^2)^2}$$

$E[u] = 0$　　$Var[a^2] = E[a^2] - (E[a])^2$

$$= \frac{E[u^2] - (E[u])^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{Var[u^2]}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

回帰係数のt検定

頑張って計算した結果、

$$E[\hat{a}_1] = \alpha_1$$

$$Var[\hat{a}_1] = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

回帰係数のt検定

頑張って計算した結果、

$$E[\hat{a}_1] = \alpha_1$$

$$Var[\hat{a}_1] = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$u \sim \mathcal{N}(0, \sigma^2)$

また式(1)

$$\hat{a}_1 = \alpha_1 + \frac{\sum_{i=1}^{n}(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\bar{u} + \frac{\sum_{i=1}^{n}(x_i - \bar{x})u_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

より $\hat{a}_1$ は正規分布に従い、

$$\hat{a}_1 \sim \mathcal{N}\left(\alpha_1, \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)$$

回帰係数のt検定

頑張って計算した結果、

$$E[\hat{a}_1] = \alpha_1$$

$$Var[\hat{a}_1] = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$u \sim \mathcal{N}(0, \sigma^2)$

また式(1)

$$\hat{a}_1 = \alpha_1 + \frac{\sum_{i=1}^{n}(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\bar{u} + \frac{\sum_{i=1}^{n}(x_i - \bar{x})u_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

より $\hat{a}_1$ は正規分布に従い、

$$\hat{a}_1 \sim \mathcal{N}\left(\alpha_1, \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right) \Leftrightarrow \frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

回帰係数のt検定

コレに関する帰無仮説を検証する

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

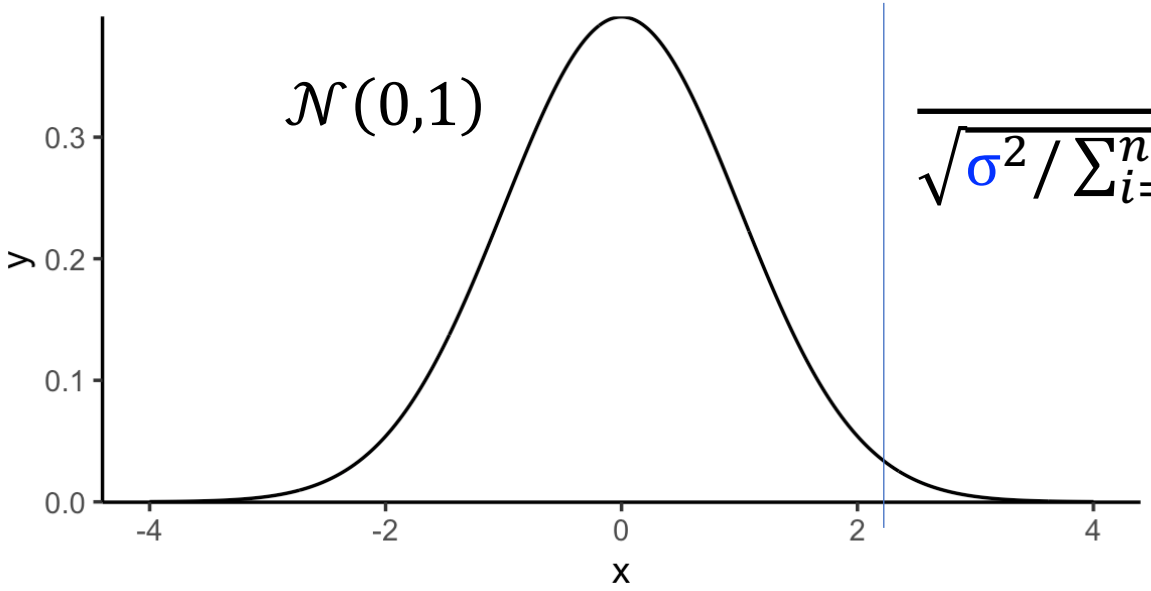
回帰係数のt検定

コレに関する帰無仮説を検証する

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

$\alpha_1 = 0$の時の左辺

$$\frac{\hat{a}_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}}$$



回帰係数のt検定

コレに関する帰無仮説を検証する

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

コレをどうにかしたい(データから推定したい)

回帰係数のt検定

コレに関する帰無仮説を検証する

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

コレをどうにかしたい(データから推定したい)

σは誤差uの標準偏差

$$u \sim_{i.i.d.} \mathcal{N}(0, \sigma^2)$$

残差標準偏差

Residual standard deviation

$$RSD = \sqrt{\frac{1}{n-k-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

誤差uの標準偏差の期待値

回帰係数のt検定

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

期待値で置き換える

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{RSD^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

回帰係数のt検定

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

期待値で置き換える

確率分布は？

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{RSD^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \ ?$$

回帰係数のt検定

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

期待値で置き換える

確率分布は？

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{RSD^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \ ?$$

$$\frac{RSD^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{1}{n-k-1} \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

回帰係数のt検定

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

期待値で置き換える

確率分布は？

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{RSD^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \ ?$$

$$\sim \chi^2(n-k-1)$$

$$\frac{RSD^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{1}{n-k-1} \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

t分布

互いに独立な、
標準正規分布$\mathcal{N}(0,1)$に従う確率変数Xと、
カイ二乗分布$\chi^2(n)$に従う確率変数Yについて

$$Z = \frac{X}{\sqrt{Y/n}}$$

なるZの確率分布をt分布と呼び、

$$Z \sim t(n)$$

と表す。(nを自由度と呼ぶ)

回帰係数のt検定

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

置き換える

確率分布も
置き換わる

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{RSD^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim t(n-k-1)$$

$$se(\hat{a}_1) := \sqrt{RSD^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}$$

回帰係数のt検定

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

置き換える

確率分布も
置き換わる

$$\frac{\hat{a}_1 - \alpha_1}{se(\hat{a}_1)} \sim t(n-k-1)$$

$$se(\hat{a}_1) := \sqrt{RSD^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}$$

回帰係数のt検定

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{\sigma^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} \sim \mathcal{N}(0,1)$$

置き換える

確率分布も
置き換わる

$$\frac{\hat{a}_1 - \alpha_1}{se(\hat{a}_1)} \sim t(n-k-1)$$

$\alpha_1 = 0$の時の左辺を計算し、分布$t(n-k-1)$に照らせば良い。

t値 (t-value)

回帰係数のt検定

$$\frac{\hat{a}_1 - \alpha_1}{\sqrt{RSD^2 / \sum_{i=1}^{n}(x_i - \bar{x})^2}} = \frac{\hat{a}_1 - \alpha_1}{se(\hat{a}_1)} \sim t(n-k-1)$$

```
a1_se <-
  df_res %>%
  summarize(RSD = sqrt(sum(res^2) / (N - 2)),
            vxx = sum((x - mean(x))^2)) %$%
  sqrt(RSD^2 / vxx)
#> [1] 0.1524416
t_value <- a1 - 0 / a1_se
#> [1] 2.408013
```

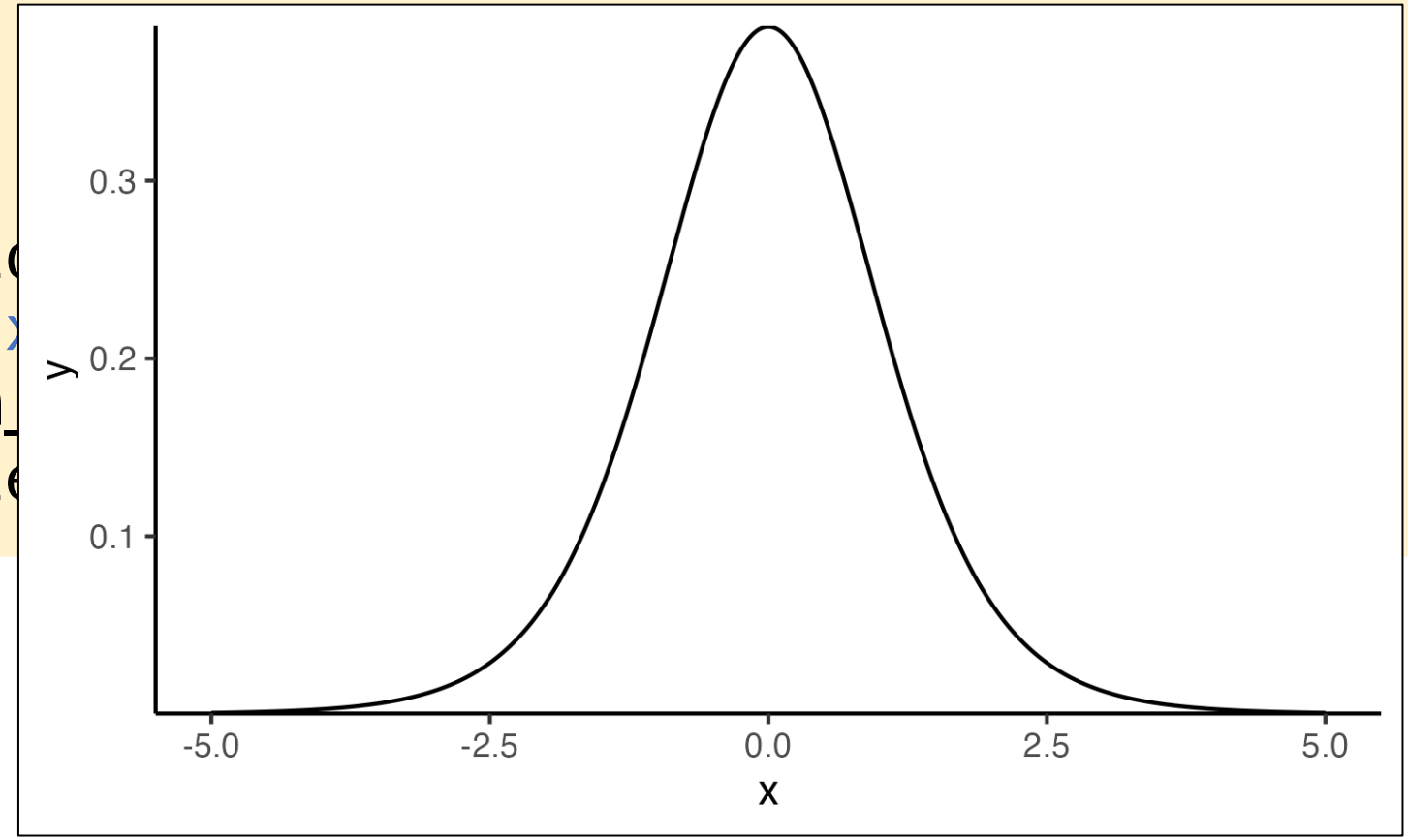
```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
```

回帰係数のt検定

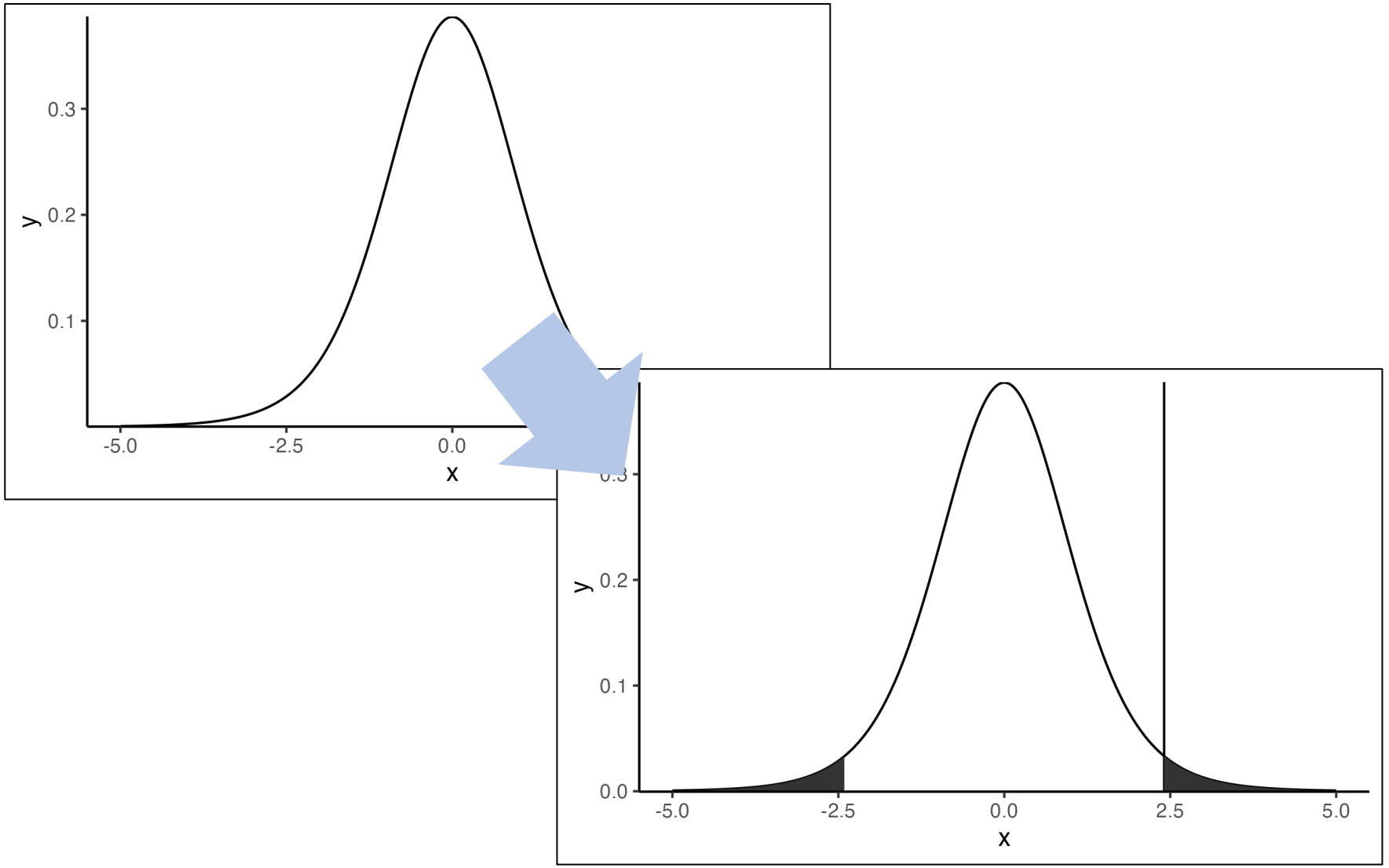
```
df_dt <-
  tibble(
    x = seq(-5, 5, by = 0.01),
    y = dt(x, df = 8)
  )


g <-
  ggplot(data = df_dt) +
  aes(x = x, y = y) +
  geom_path() +
  scale_y_continuous(expand = c(0.001, 0))
```

回帰係数のt検定

```
df_dt <-
  tibble(
    x = seq(-5, 5, by = 0.01),
    y = dt(x, df = 8)
  )

g <-
  ggplo
  aes(x
  geom_
  scale
```



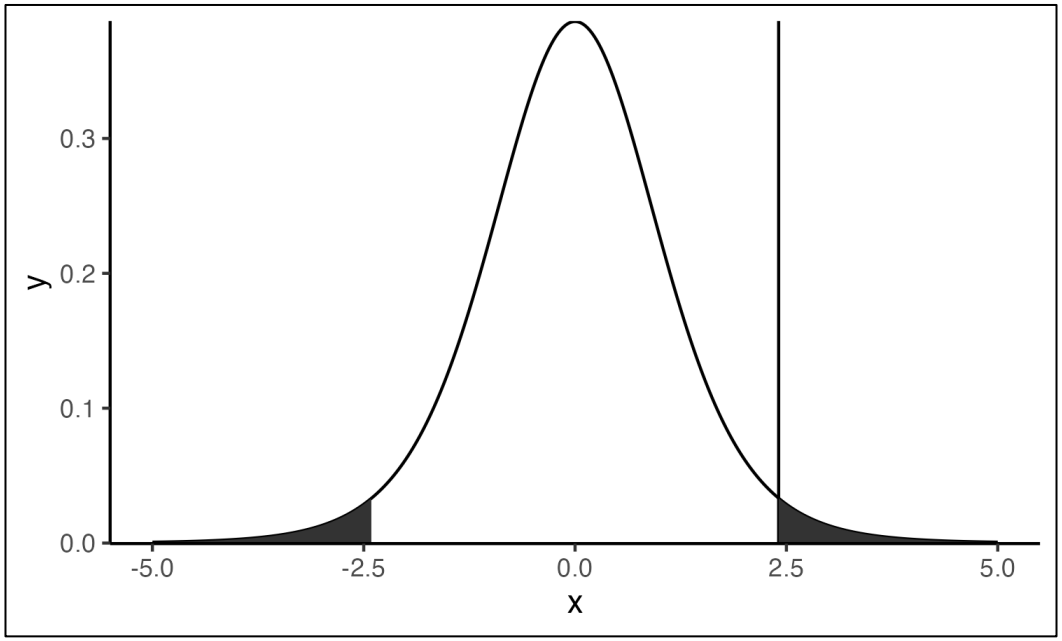
回帰係数のt検定



回帰係数のt検定

```
g +
  geom_vline(xintercept = t_value) +
  geom_ribbon(
    data = df_dt %>% filter(x >= t_value),
    mapping = aes(ymin = 0, ymax = y)
  ) +
  geom_ribbon(
    data = df_dt %>% filter(x <= -t_value),
    mapping = aes(ymin = 0, ymax = y)
  )
```

回帰係数のt検定




```
pt(q = t_value, df = 8, lower.tail = FALSE) * 2
#> [1] 0.04264022
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

deviation

最小二乗法（線形回帰）

$$y = \hat{a}_0 + \hat{a}_1 x + \varepsilon$$

回帰係数$\hat{a}_0$の仮説検定を行う。

真の切片α_0がゼロだった時、
回帰係数$\hat{a}_0$が得られる確率。

α_1と同じことをすれば良いですね

あとは各自、家でやってください

α_1と同じことをすれば良いですね

あとは各自、家でやってください

とは言いません。

回帰係数のt検定

【基本方針】

$$\frac{\hat{a}_0 - \alpha_0}{\sigma^2\text{の関数!?}} \sim \mathcal{N}(0,1)$$

σをRSDに置き換える

$$\frac{\hat{a}_0 - \alpha_0}{se(\hat{a}_0)} \sim t(n-k-1)$$

回帰係数のt検定

$$\hat{a}_0 = \bar{y} - \hat{a}_1 \bar{x}$$

$$= \alpha_0 + \alpha_1 \bar{x} - \bar{u} - \hat{a}_1 \bar{x}$$

$$E[\hat{a}_0] = E[\alpha_0 + \alpha_1 \bar{x} - \bar{u} - \hat{a}_1 \bar{x}]$$

$$= \alpha_0 + \alpha_1 \bar{x} - 0 - \alpha_1 \bar{x}$$

$$= \alpha_0$$

$$0 = E[\bar{u}]$$

$$\alpha_1 = E[\hat{a}_1]$$

回帰係数のt検定

$$\hat{a}_0 = \bar{y} - \hat{a}_1 \bar{x}$$

$$= \alpha_0 + \alpha_1 \bar{x} - \bar{u} - \hat{a}_1 \bar{x}$$

$$E[\hat{a}_0] = E[\alpha_0 + \alpha_1 \bar{x} - \bar{u} - \hat{a}_1 \bar{x}]$$

$$= \alpha_0 + \alpha_1 \bar{x} - 0 - \alpha_1 \bar{x}$$

$$= \alpha_0$$

$0 = E[\bar{u}]$

$\alpha_1 = E[\hat{a}_1]$

$$Var[\hat{a}_0] = E[(\hat{a}_0 - E[\hat{a}_0])^2] = E[(\hat{a}_0 - \alpha_0)^2]$$

$$= \ \ldots$$

$$= \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \right)$$

回帰係数のt検定

$$\hat{a}_0 \sim \mathcal{N}(E[\hat{a}_0], Var[\hat{a}_0])$$

$$\hat{a}_0 \sim \mathcal{N}\left(\alpha_0, \sigma^2\left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)\right)$$

$$\frac{\hat{a}_0 - \alpha_0}{\sqrt{\sigma^2\left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)}} \sim \mathcal{N}(0,1)$$

回帰係数のt検定

$$\frac{\hat{a}_0 - \alpha_0}{\sqrt{\sigma^2\left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)}} \sim \mathcal{N}(0,1)$$

σをRSDに置き換える

$$\frac{\hat{a}_0 - \alpha_0}{se(\hat{a}_0)} \sim t(n-k-1)$$

$$se(\hat{a}_0) := RSD\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

回帰係数のt検定

$$\frac{\hat{a}_0 - \alpha_0}{RSD\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}}$$

```
a0_se <-
  df_res %>%
  summarize(RSD = sqrt(sum(res^2) / (N - 2)),
            vxx = sum((x - mean(x))^2),
            barx2 = mean(x)^2) %$%
  {RSD * sqrt(1 / N + barx2 / vxx)}
#> [1] 7.534808
t_a0 <- a0 / a0_se
#> [1] 0.03581352
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
```

回帰係数のt検定

$$\frac{\hat{a}_0 - \alpha_0}{RSD\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}}$$

```
pt(q = t_a0, df = 8, lower.tail = FALSE) * 2
#> [1] 0.9721359
```

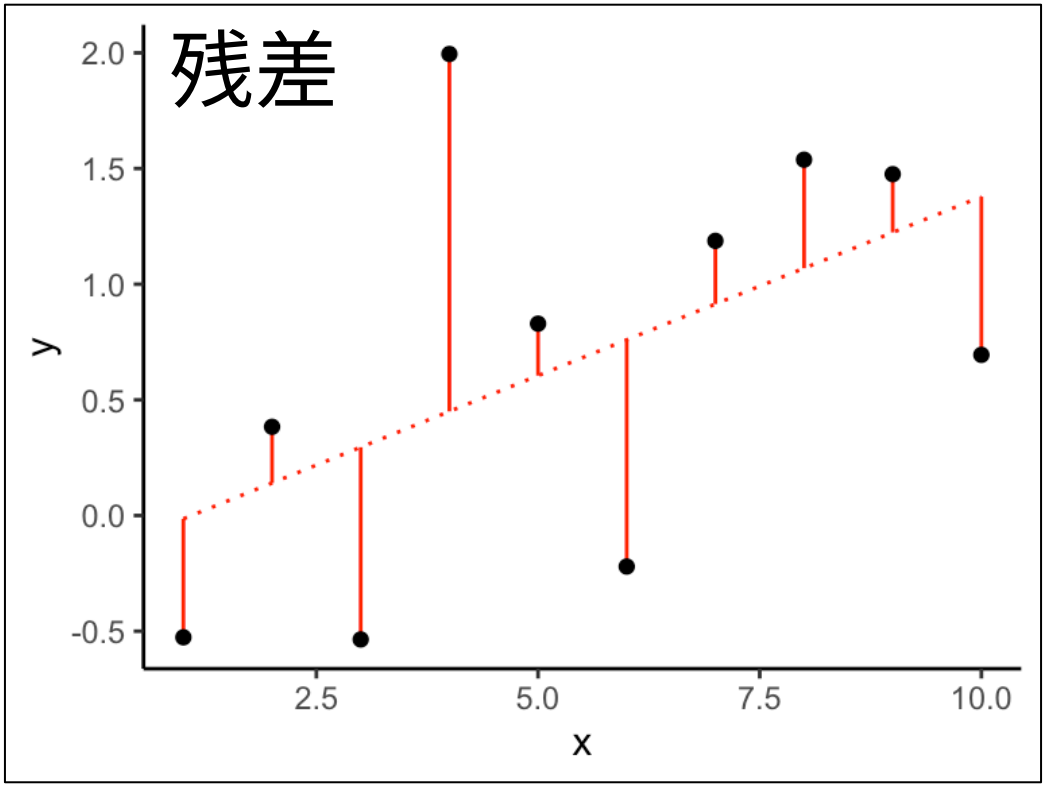
```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.2698     7.5348   0.036   0.9723
x              0.3671     0.1524   2.408   0.0426 *
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

                    deviation
Residual standard ~~error~~: 0.9279 on 8 degrees of freedom
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Residuals:
     Min       1Q   Median       3Q      Max
-1.25050 -0.75400 -0.06001  0.56055  1.34316
```



回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
```

$$\frac{\hat{a}_0 - E[\hat{a}_0]}{Var[\hat{a}_0]} \sim N(0,1)$$

$$\frac{\hat{a}_1 - E[\hat{a}_1]}{Var[\hat{a}_1]} \sim N(0,1)$$

$$E[\hat{a}_0] = \alpha_0$$

$$Var[\hat{a}_1] = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \right)$$

$$E[\hat{a}_1] = \alpha_1$$

$$Var[\hat{a}_1] = \frac{\sigma^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2698     7.5348   0.036   0.9723
x             0.3671     0.1524   2.408   0.0426 *
```

$$\frac{\hat{a}_0 - E[\hat{a}_0]}{Var[\hat{a}_0]} \sim N(0,1) \qquad \frac{\hat{a}_1 - E[\hat{a}_1]}{Var[\hat{a}_1]} \sim N(0,1)$$

σをRSDに置き換える

$$\frac{\hat{a}_0 - \alpha_0}{se(\hat{a}_0)} \sim t(n-k-1) \qquad \frac{\hat{a}_1 - \alpha_1}{se(\hat{a}_1)} \sim t(n-k-1)$$

回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()
```

```
Residual standard error: 0.9279 on 8 degrees of freedom
```

deviation

$$RSD = \sqrt{\frac{1}{n-k-1}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

誤差uの標準偏差の期待値

回帰モデル regression

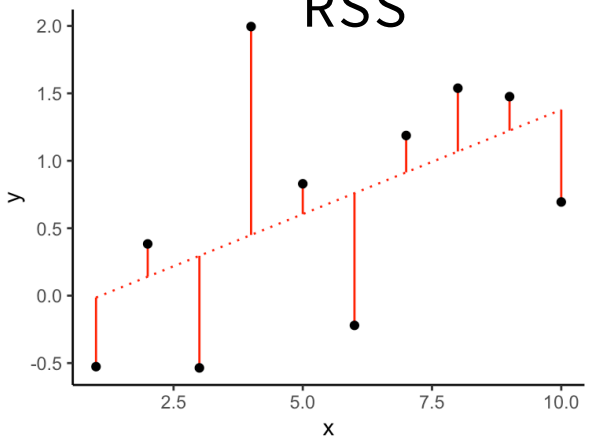
```
lm(y ~ x, data = df_xy) %>% summary()
```

```
Multiple R-squared:  0.4202,  Adjusted R-squared:  0.3478
```

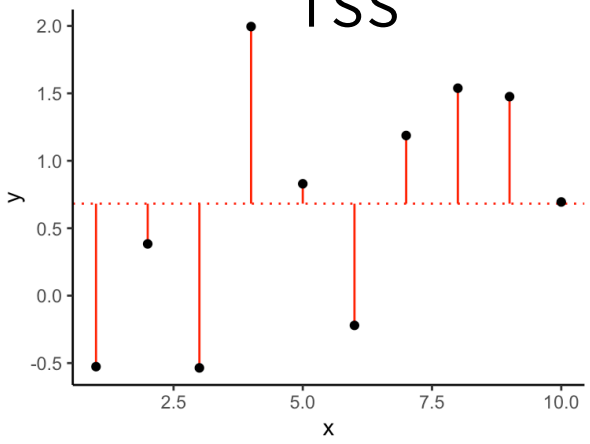
決定係数

$$R2 = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$

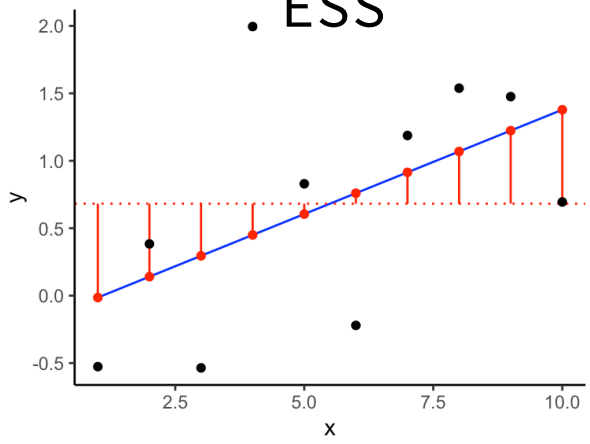
残差変動
RSS



全変動
TSS



回帰変動
ESS



回帰モデル regression

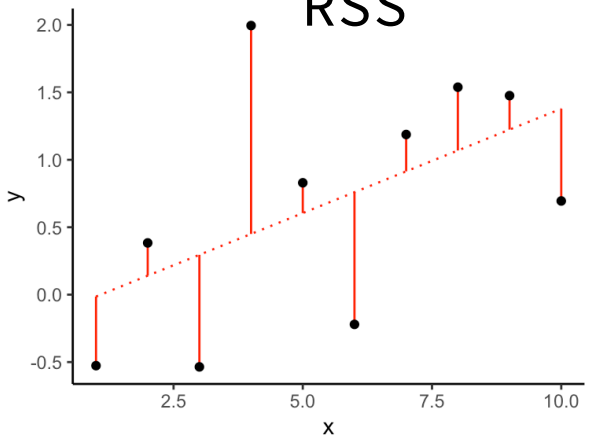
```
lm(y ~ x, data = df_xy) %>% summary()
```

```
F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

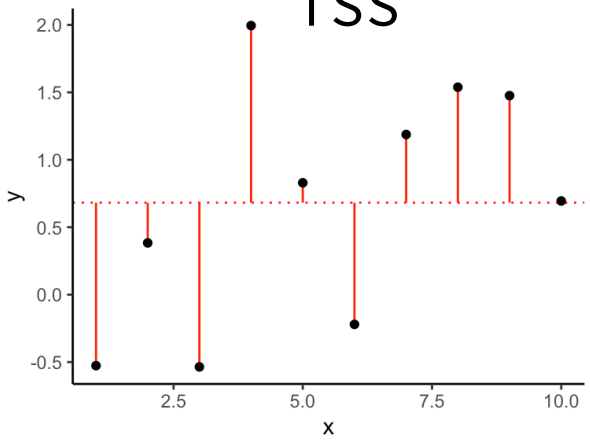
f値

$$f = \frac{ESS/k}{RSS/(n-k-1)} \sim F(k, n-k-1)$$

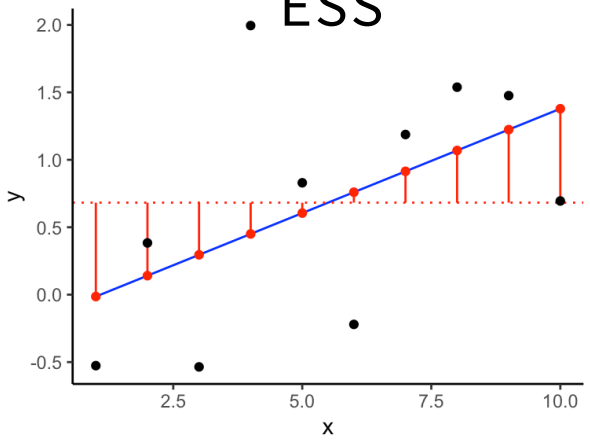
残差変動
RSS



全変動
TSS



回帰変動
ESS



回帰モデル regression

```
lm(y ~ x, data = df_xy) %>% summary()

F-statistic: 5.799 on 1 and 8 DF,  p-value: 0.04264
```

分散分析 ANOVA

```
aov(y ~ x, data = df_xy) %>% summary()

            Df Sum Sq Mean Sq F value Pr(>F)
x            1  4.992   4.992   5.799 0.0426 *
Residuals    8  6.888   0.861
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

分散分析 ANOVA

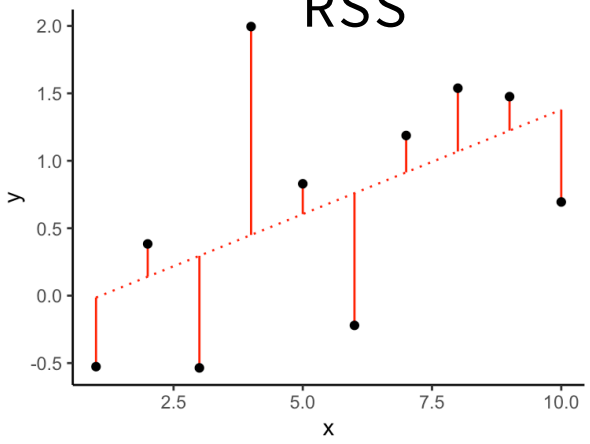
```
aov(y ~ x, data = df_xy) %>% summary()
```

```
            Df Sum Sq Mean Sq F value Pr(>F)
x            1  4.992   4.992   5.799 0.0426 *
Residuals    8  6.888   0.861
```

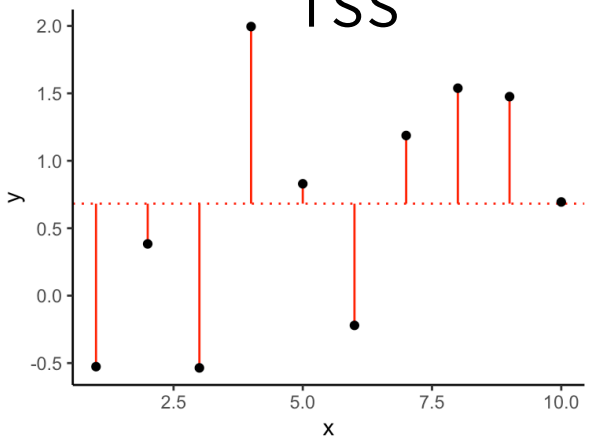
f値

$$f = \frac{ESS/k}{RSS/(n-k-1)} \sim F(k, n-k-1)$$

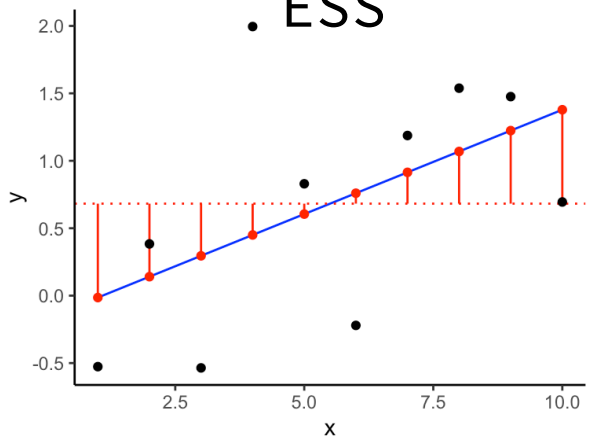
残差変動
RSS



全変動
TSS



回帰変動
ESS



分散分析 ANOVA

総平均　効果　誤差

$$y_{ij} = \mu_0 + \sigma_j + \varepsilon_{ij}$$

$$\varepsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$

回帰モデル regression

$$y_i = \alpha_0 + \alpha_1 x_i + u_i$$

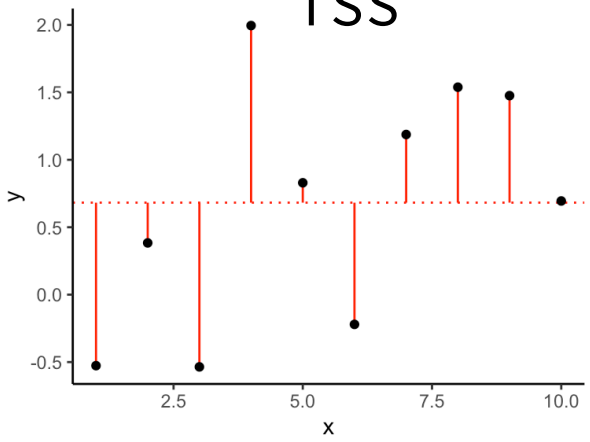
$$u_i \sim \mathcal{N}(0, \sigma^2)$$

分散分析 ANOVA

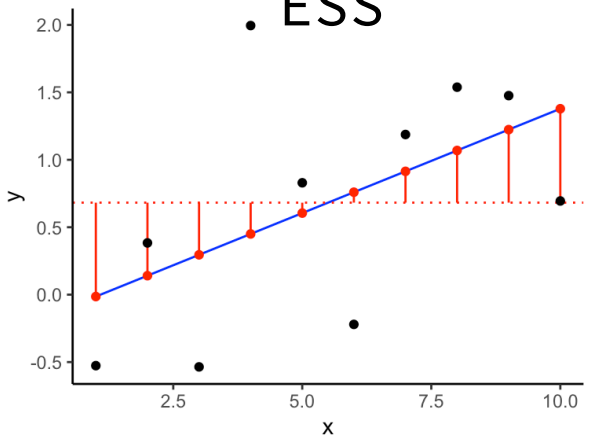
総平均　　効果　　誤差

$$y_{ij} - \mu_0 = \sigma_j + \varepsilon_{ij}$$

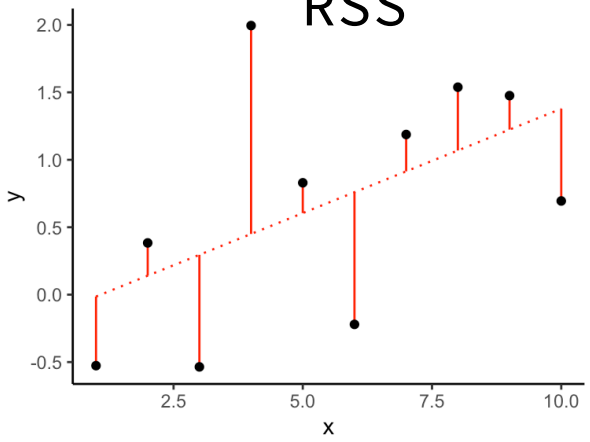
全変動
TSS



回帰変動
ESS



残差変動
RSS



$$f = \frac{ESS/k}{RSS/(n-k-1)} \sim F(k, n-k-1)$$

分散分析 ANOVA

```
df_for_aov <-
  penguins %>%
  na.omit() %>%
  filter(island == "Dream") %>%
  mutate(x = bill_length_mm,
         y = body_mass_g)
```

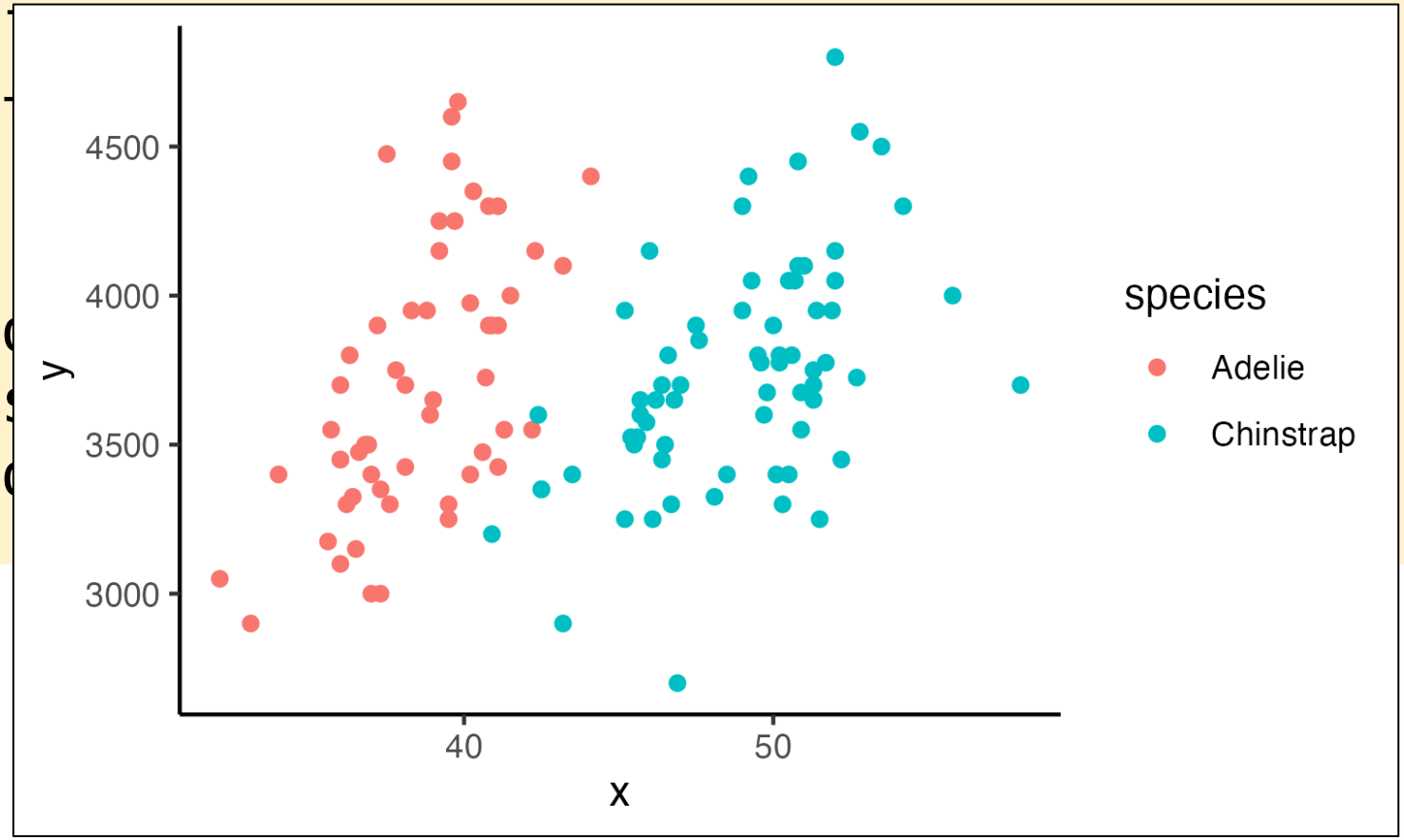
```
# A tibble: 123 × 10
   species island bill_length_mm bill_depth_mm flipper_length_mm
   <fct>   <fct>           <dbl>         <dbl>             <int>
 1 Adelie  Dream            39.5          16.7               178
 2 Adelie  Dream            37.2          18.1               178
 3 Adelie  Dream            39.5          17.8               188
 4 Adelie  Dream            40.9          18.9               184
 5 Adelie  Dream            36.4          17                 195
 6 Adelie  Dream            39.2          21.1               196
 7 Adelie  Dream            38.8          20                 190
 8 Adelie  Dream            42.2          18.5               180
 9 Adelie  Dream            37.6          19.3               181
10 Adelie  Dream            39.8          19.1               184
# ℹ 113 more rows
# ℹ 5 more variables: body_mass_g <int>, sex <fct>, year <int>,
```

回帰 regression

```
df_for_aov <-
  penguins %>%
  na.omit() %>%
  filter(island == "Dream") %>%
  mutate(x = bill_length_mm,
         y = body_mass_g)

ggplot(data = df_for_aov) +
  aes(x = x, y = y) +
  geom_point(mapping = aes(color = species))
```

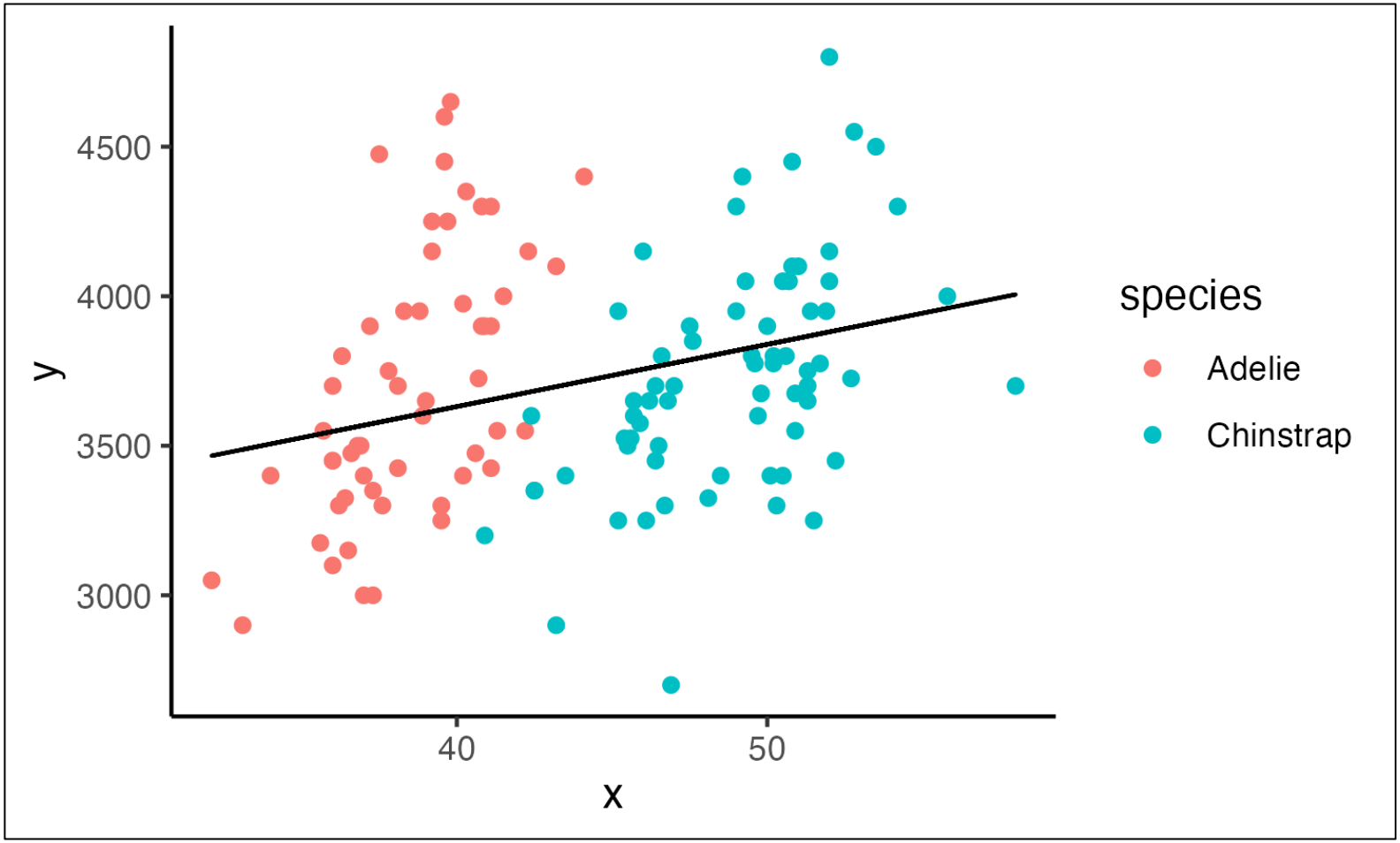
回帰 regression

```
df_for_aov <-
  penguins %>%
  na.omit() %>%
  fi
  mu

ggplo
  aes
  geo
```



回帰 regression

```
lm(y ~ x, data = dat_for_aov)
```



回帰 regression

```
lm(y ~ x, data = dat_for_aov)
```

```
df_pred <-
  df_for_aov %>%
  mutate(pred = df_for_aov %>%
           lm(y ~ x, data = .) %>%
           predict()
  )

ggplot(data = df_for_aov) +
  aes(x = x, y = y) +
  geom_point(mapping = aes(color = species)) +
  geom_path(data = df1,
            aes(y = pred))
```

回帰 regression

離散量

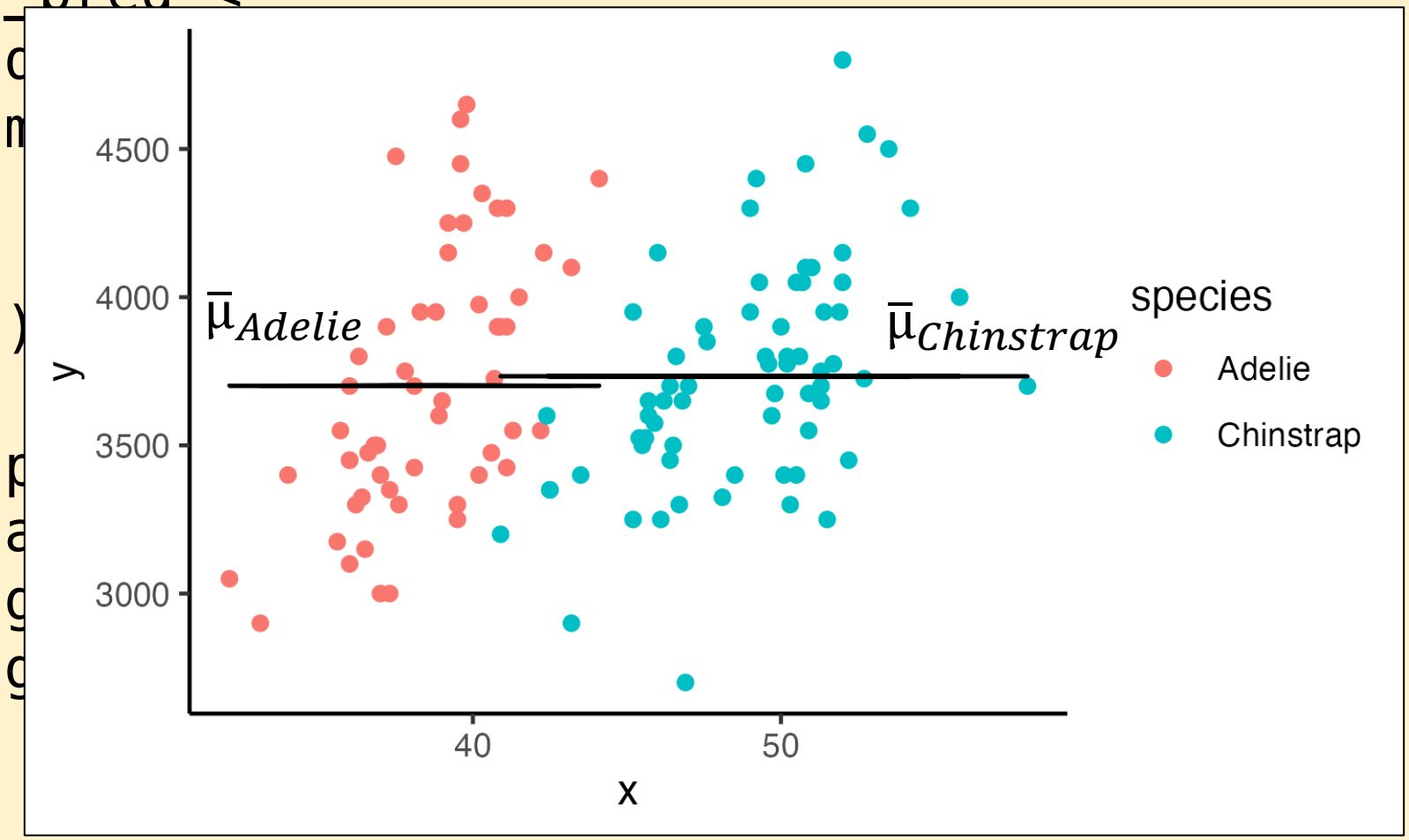
```
lm(y ~ species, data = dat_for_aov)
```

```
df_pred <-
  df_for_aov %>%
  mutate(pred = df_for_aov %>%
           lm(y ~ species, data = .) %>%
           predict()
  )

ggplot(data = df_for_aov) +
  aes(x = x, y = y) +
  geom_point(mapping = aes(color = species)) +
  geom_path(data = df1,
            aes(y = pred, group = species))
```

回帰 regression

```
lm(y ~ species, data = dat_for_aov)
```

```
df_pred <-
  d
  m

  )

ggp
  a
  g
  g
```




```
) +
```

重回帰 multiple regression

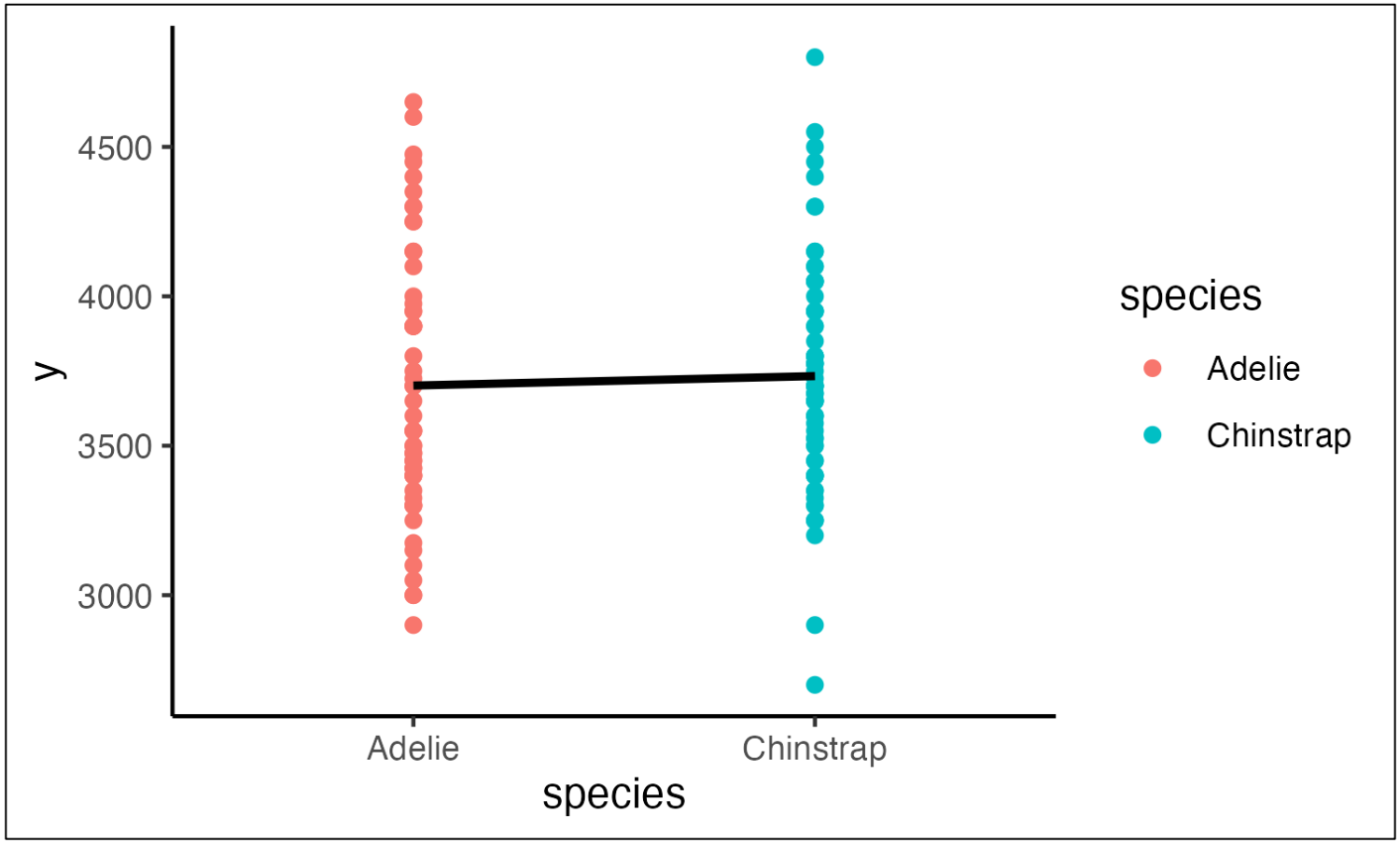
離散量

```
lm(y ~ species, data = dat_for_aov)
```

```
df_for_aov %>%
  lm(y ~ species, data = .)
#> Coefficients:
#>     (Intercept)  speciesChinstrap
#>         3701.36             31.72

df_for_aov %>%
  mutate(key = if_else(species == "Adelie", 0, 1)) %>%
  lm(y ~ key, data = .)
#> Coefficients:
#> (Intercept)          key
#>     3701.36        31.72
```

分散分析 ANOVA

```
lm(y ~ species, data = dat_for_aov)
```



重回帰 multiple regression

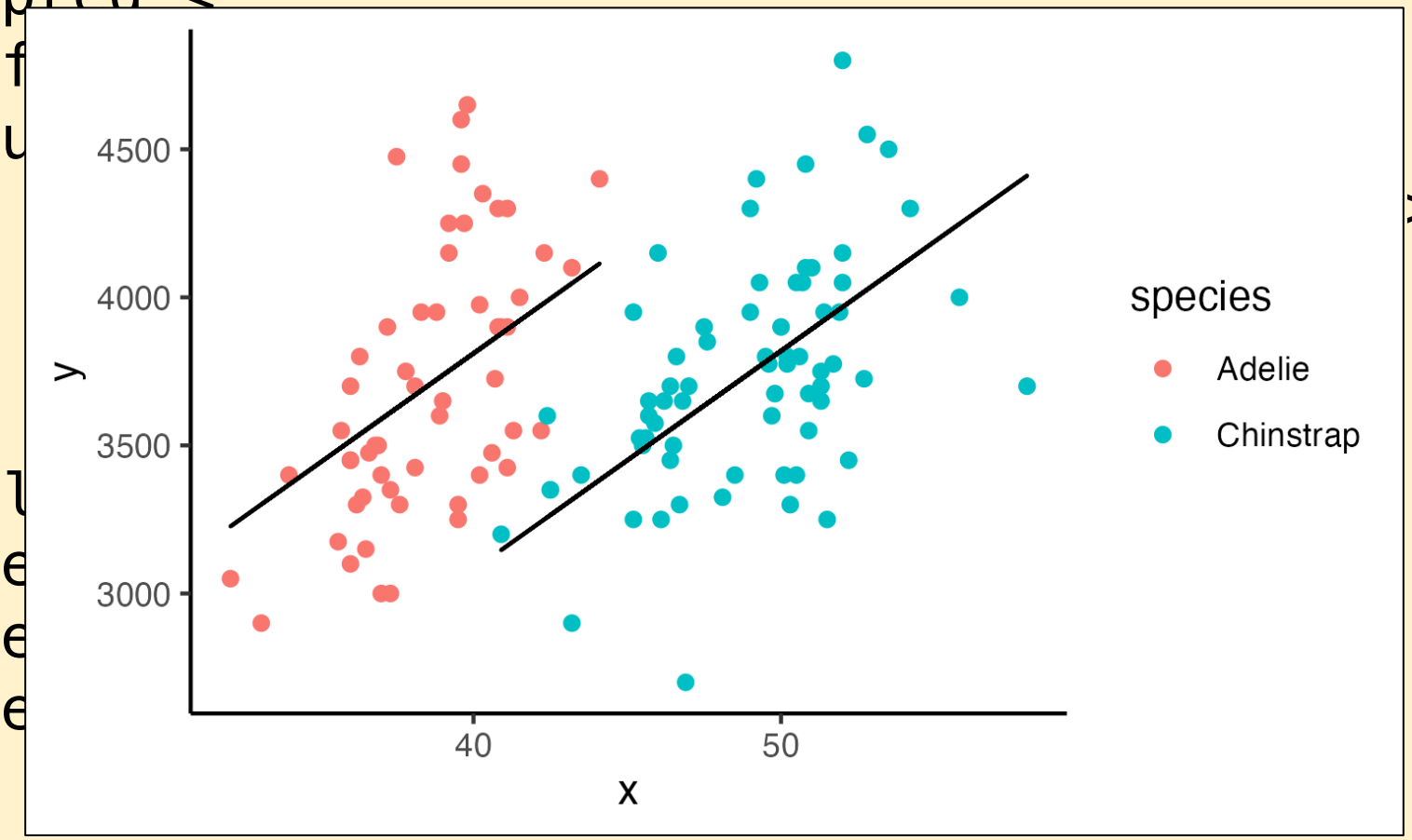
```
lm(y ~ x + species, data = dat_for_aov)
```

```
df_pred <-
  df_for_aov %>%
  mutate(pred = df_for_aov %>%
           lm(y ~ x + species, data = .) %>%
           predict()
  )

ggplot(data = df_for_aov) +
  aes(x = x, y = y) +
  geom_point(mapping = aes(color = species)) +
  geom_path(data = df1,
            aes(y = pred))
```

重回帰 multiple regression

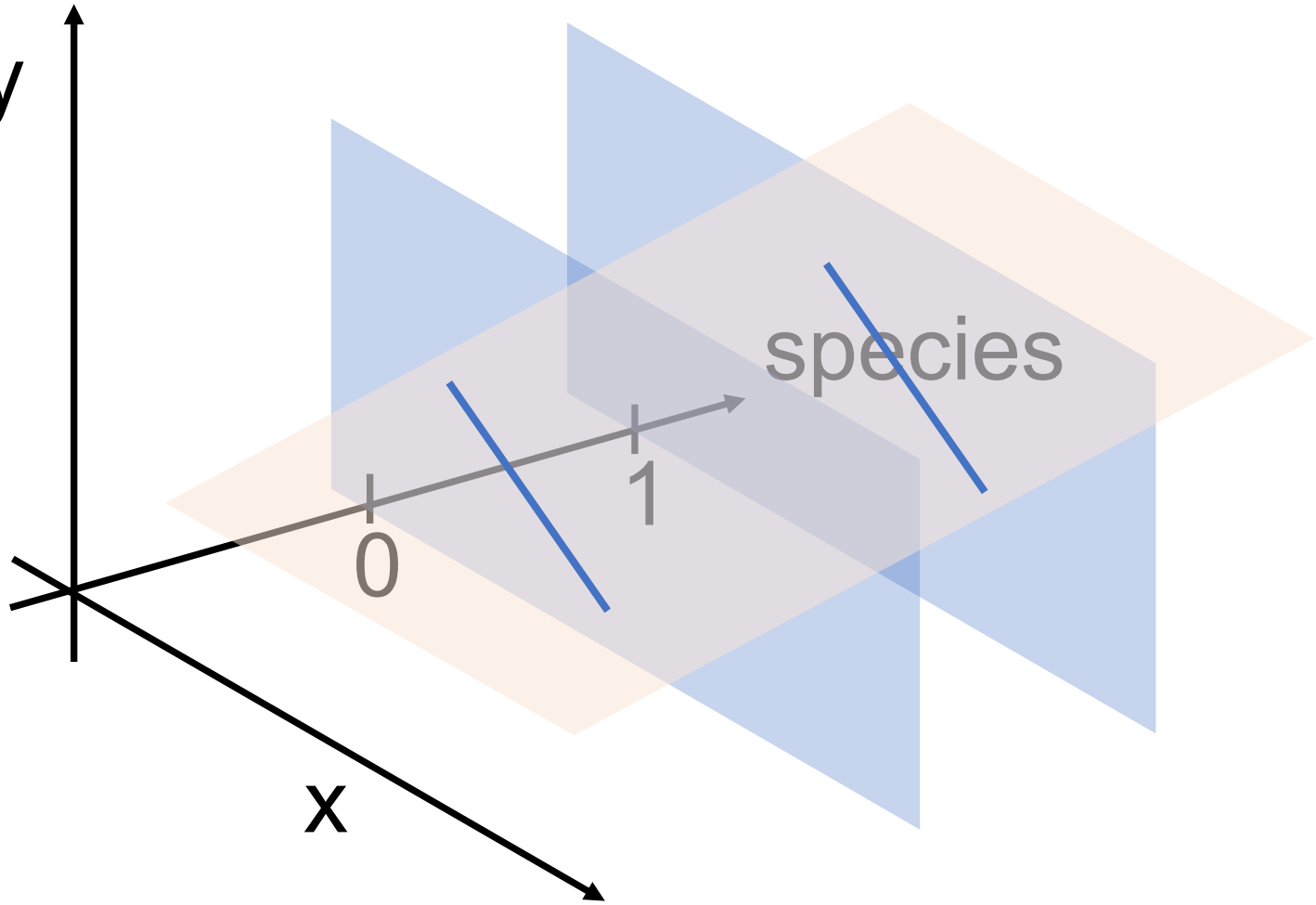
```
lm(y ~ x + species, data = dat_for_aov)
```

```
df_pred <-
  df
  mu
                                                    %>%
  )

ggpl
  ae
  ge                                                )) +
  ge
```



重回帰 multiple regression

```
lm(y ~ x + species, data = dat_for_aov)
```



重回帰 multiple regression

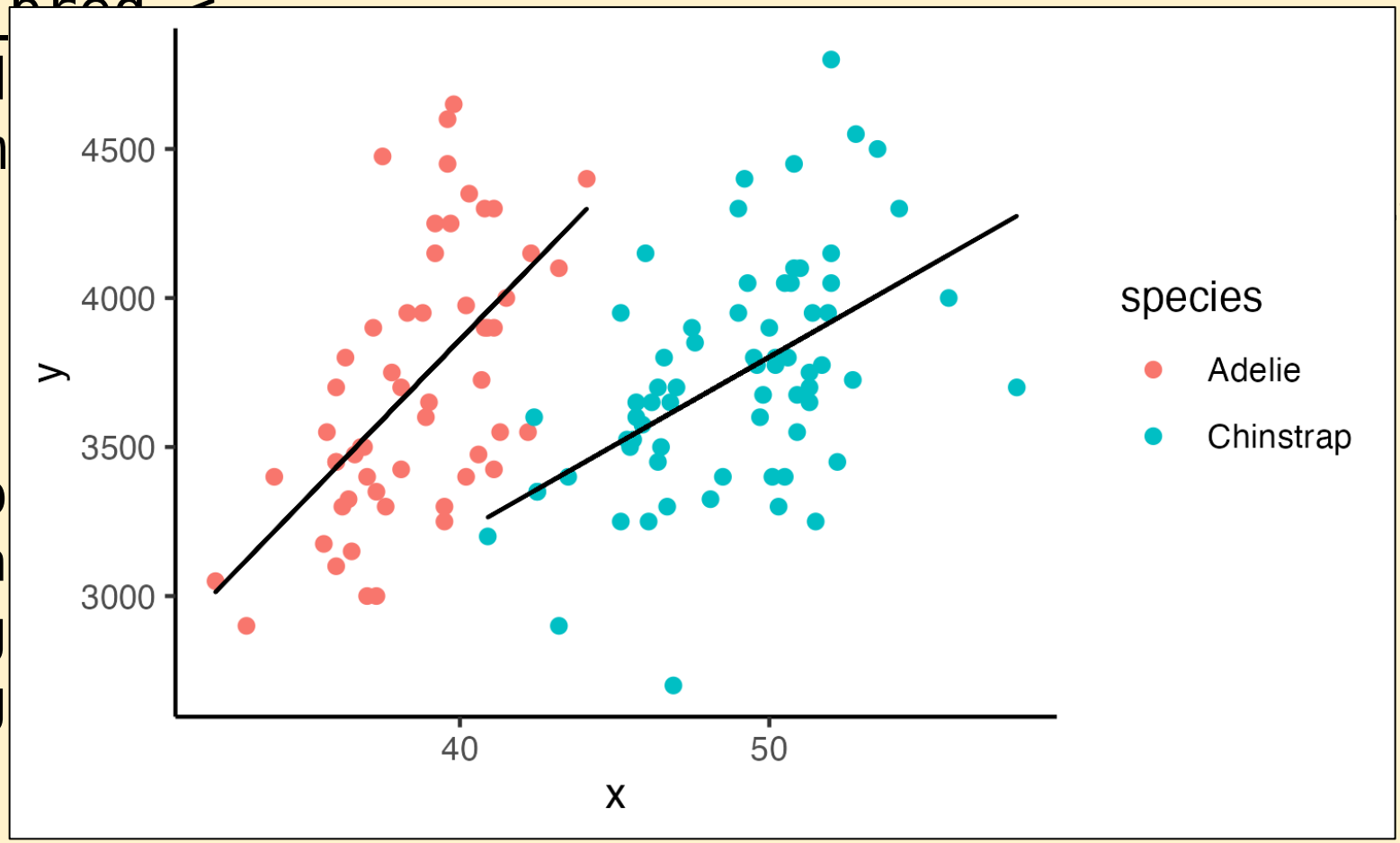
```
lm(y ~ x * species, data = dat_for_aov)
```

```
df_pred <-
  df_for_aov %>%
  mutate(pred = df_for_aov %>%
           lm(y ~ x * species, data = .) %>%
           predict()
  )

ggplot(data = df_for_aov) +
  aes(x = x, y = y) +
  geom_point(mapping = aes(color = species)) +
  geom_path(data = df1,
            aes(y = pred))
```

重回帰 multiple regression

```
lm(y ~ x * species, data = dat_for_aov)
```

```
df_
  d
  m
  )

ggp
  a
  g
  g
```




```
%

) +
```

重回帰 multiple regression

```
lm(y ~ x * species, data = dat_for_aov)
```

交互作用項

$$y = \alpha_0 + \alpha_1 x + \alpha_3 s + \alpha_3 xs + \varepsilon$$

speciesのダミー変数
(Adelie = 0, Chinstrap = 1)

```
> df_for_aov %>%
+   lm(y ~ x * species, data = .)

Coefficients:
       (Intercept)                   x    speciesChinstrap  x:speciesChinstrap
           -425.63                107.14             1271.77              -48.02
```

```
> df_for_aov %>%
+  mutate(tag = as.numeric(species) - 1,
+         tag2 = x * tag) %>%
+  lm(y ~ x + tag + tag2, data = .)

Coefficients:
(Intercept)            x          tag         tag2
    -425.63       107.14      1271.77       -48.02
```

分散分析 ANOVA

```
df_for_aov %>%
  aov(y ~ x, data = .) %>%
  summary()
```

```
            Df   Sum Sq Mean Sq F value   Pr(>F)
x            1  1869797 1869797   11.95 0.000755 ***
Residuals  121 18933130  156472
```

```
df_for_aov %>%
  lm(y ~ x, data = .) %>%
  summary()
```

```
F-statistic: 11.95 on 1 and 121 DF,  p-value: 0.0007548
```

分散分析 ANOVA

```
df_for_aov %>%
  aov(y ~ x, data = .) %>%
  summary()
```

```
            Df   Sum Sq Mean Sq F value   Pr(>F)
x            1  1869797 1869797   11.95 0.000755 ***
Residuals  121 18933130  156472
```

```
df_for_aov %>%
  lm(y ~ x, data = .) %>%
  summary()
```

```
F-statistic: 11.95 on 1 and 121 DF,  p-value: 0.0007548
```

分散分析 ANOVA

```
df_for_aov %>%
  aov(y ~ species, data = .) %>%
  summary()
```

```
             Df   Sum Sq Mean Sq F value Pr(>F)
species       1    30603   30603   0.178  0.674
Residuals   121 20772324  171672
```

```
df_for_aov %>%
  lm(y ~ species, data = .) %>%
  summary()
```

```
F-statistic: 0.1783 on 1 and 121 DF,  p-value: 0.6736
```

分散分析 ANOVA

```
df_for_aov %>%
  aov(y ~ x + species, data = .) %>%
  summary()
```

```
             Df   Sum Sq Mean Sq F value   Pr(>F)
x             1  1869797 1869797   15.09 0.000168 ***
species       1  4065873 4065873   32.82 7.65e-08 ***
Residuals   120 14867257  123894
```

```
df_for_aov %>%
  lm(y ~ x + species, data = .) %>%
  summary()
```

```
F-statistic: 23.95 on 2 and 120 DF,  p-value: 1.764e-09
```

分散分析 ANOVA

```
df_for_aov %>%
  aov(y ~ x * species, data = .) %>%
  summary()
```

```
            Df   Sum Sq Mean Sq F value   Pr(>F)
x            1  1869797 1869797  15.521 0.000138 ***
species      1  4065873 4065873  33.750 5.34e-08 ***
x:species    1   531455  531455   4.412 0.037808 *
Residuals  119 14335802  120469
```

```
df_for_aov %>%
  lm(y ~ x * species, data = .) %>%
  summary()
```

```
F-statistic: 17.89 on 3 and 119 DF,  p-value: 1.189e-09
```

分散分析 ANOVA

```
df_for_aov %>%
  aov(y ~ x * species, data = .) %>%
  summary()
```

```
             Df   Sum Sq Mean Sq F value   Pr(>F)
x             1  1869797 1869797  15.521 0.000138 ***
species       1  4065873 4065873  33.750 5.34e-08 ***
x:species     1   531455  531455   4.412 0.037808 *
Residuals   119 14335802  120469
```

```
df_for_aov %>%
  lm(y ~ x * species, data = .) %>%
  summary()
```

```
Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       -425.63     733.86  -0.580   0.5630
x                  107.14      19.01   5.635 1.19e-07 ***
speciesChinstrap  1271.77     961.70   1.322   0.1886
x:speciesChinstrap  -48.02      22.86  -2.100   0.0378 *
```

分散分析 ANOVA

```
df_for_aov %>%
  aov(y ~ x * species, data = .) %>%
  summary()
```

```
             Df   Sum Sq Mean Sq F value   Pr(>F)
x             1  1869797 1869797  15.521 0.000138 ***
species       1  4065873 4065873  33.750 5.34e-08 ***
x:species     1   531455  531455   4.412 0.037808 *
Residuals   119 14335802  120469
```

総平均　効果　誤差

$$y_{ij} - \mu_0 = \sigma_j + \varepsilon_{ij}$$

$$f = \frac{ESS/df_{ESS}}{RSS/df_{RSS}} \sim F(df_{ESS}, df_{RSS})$$

分散分析 ANOVA

```
df_for_aov %>%
  aov(y ~ x + Error(species), data = .) %>%
  summary()
Error: species
  Df Sum Sq Mean Sq
x  1  30603   30603

Error: Within
           Df   Sum Sq Mean Sq F value   Pr(>F)
x           1  5905067 5905067   47.66 2.57e-10 ***
Residuals 120 14867257  123894
```

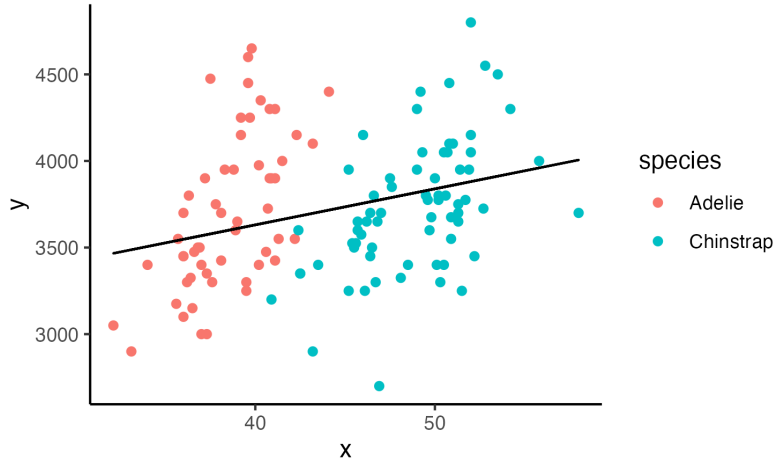
反復測定分散分析:

例えば「各種について同一個体を繰り返し捕獲して測定した」といった観察の評価に用いる。

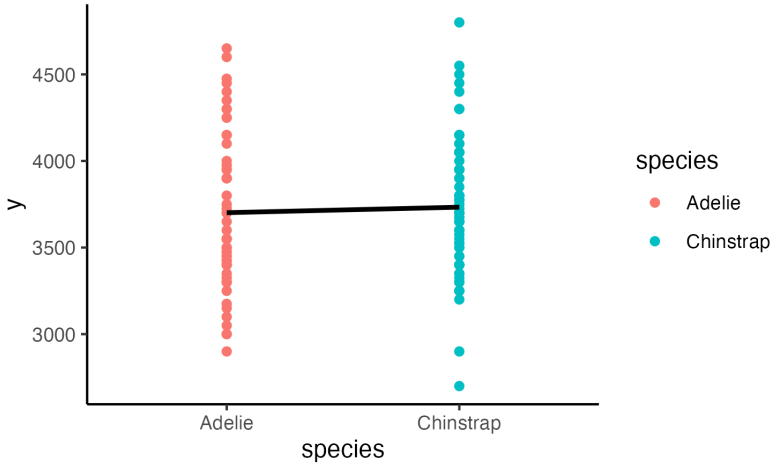
→ speciesの群内ではi.i.d.が**成り立たない**と仮定する

モデル選択 model selection

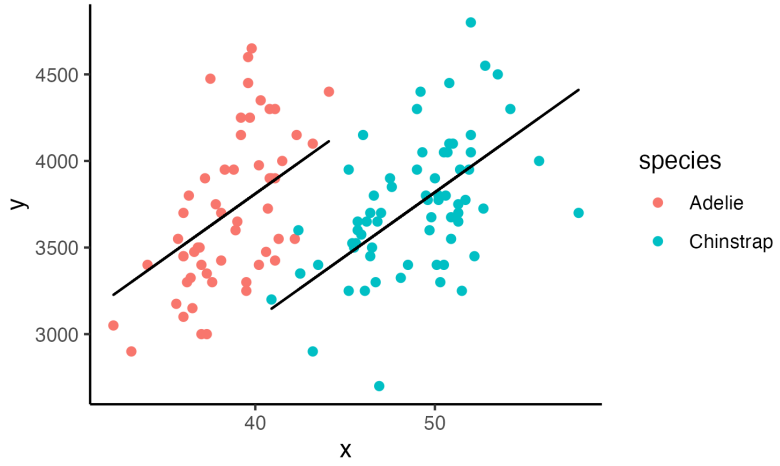
① `y ~ x`



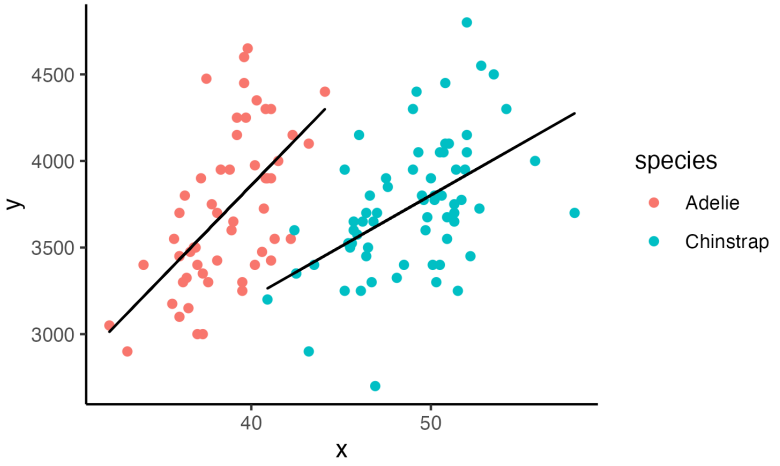
② `y ~ species`



③ `y ~ x + species`



④ `y ~ x * species`



モデル選択 model selection

```
# model1
df_for_aov %>%
  lm(y ~ x, data = .) %>% AIC()
#> [1] 1824.2

# model2
df_for_aov %>%
  lm(y ~ species, data = .) %>% AIC()
#> [1] 1835.603

# model3
df_for_aov %>%
  lm(y ~ x + species, data = .) %>% AIC()
#> [1] 1796.465

# model4
df_for_aov %>%
  lm(y ~ x * species, data = .) %>% AIC()
#> [1] 1793.987
```

モデル選択 model selection

```
# modelの関数をリストにしておく
models <-
  list(
    function(data){lm(y ~ x, data = data)},
    function(data){lm(y ~ species, data = data)},
    function(data){lm(y ~ x + species, data = data)},
    function(data){lm(y ~ x * species, data = data)}
  )
```

モデル選択 model selection

```
df_for_aov
# A tibble: 123 × 10
   species island bill_length_mm bill_depth_mm flipper_length_mm
   <fct>   <fct>           <dbl>         <dbl>             <int>
 1 Adelie  Dream            39.5          16.7               178
 2 Adelie  Dream            37.2          18.1               178
 3 Adelie  Dream            39.5          17.8               188
 4 Adelie  Dream            40.9          18.9               184
 5 Adelie  Dream            36.4          17                 195
 6 Adelie  Dream            39.2          21.1               196
 7 Adelie  Dream            38.8          20                 190
 8 Adelie  Dream            42.2          18.5               180
 9 Adelie  Dream            37.6          19.3               181
10 Adelie  Dream            39.8          19.1               184
# ℹ 113 more rows
# ℹ 5 more variables: body_mass_g <int>, sex <fct>, year <int>,
#   x <dbl>, y <int>
# ℹ Use `print(n = ...)` to see more rows
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest()

# A tibble: 1 × 1
  data
  <list>
1 <tibble [123 × 10]>
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models))

# A tibble: 1 × 2
  data                model
  <list>              <list>
1 <tibble [123 × 10]> <list [4]>
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model)

# A tibble: 4 × 2
  data                model
  <list>              <list>
1 <tibble [123 × 10]> <fn>
2 <tibble [123 × 10]> <fn>
3 <tibble [123 × 10]> <fn>
4 <tibble [123 × 10]> <fn>
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y)))
```

```
# A tibble: 4 × 3
  data                model  fit
  <list>              <list> <list>
1 <tibble [123 × 10]> <fn>   <lm>
2 <tibble [123 × 10]> <fn>   <lm>
3 <tibble [123 × 10]> <fn>   <lm>
4 <tibble [123 × 10]> <fn>   <lm>
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y))) %>%
  mutate(AIC = purrr::map_dbl(fit, ~ AIC(.)))
```

```
# A tibble: 4 × 4
  data                model  fit      AIC
  <list>              <list> <list> <dbl>
1 <tibble [123 × 10]> <fn>   <lm>   1824.
2 <tibble [123 × 10]> <fn>   <lm>   1836.
3 <tibble [123 × 10]> <fn>   <lm>   1796.
4 <tibble [123 × 10]> <fn>   <lm>   1794.
```

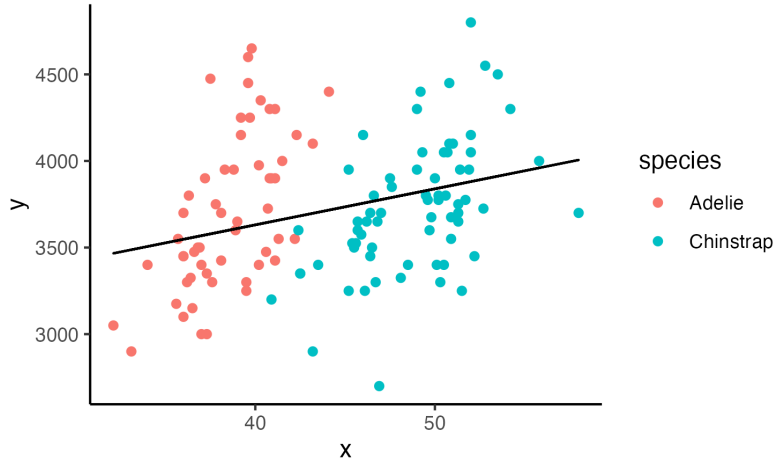
モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y))) %>%
  mutate(AIC = purrr::map_dbl(fit, ~ AIC(.))) %>%
  mutate(dAIC = AIC - min(AIC))
```

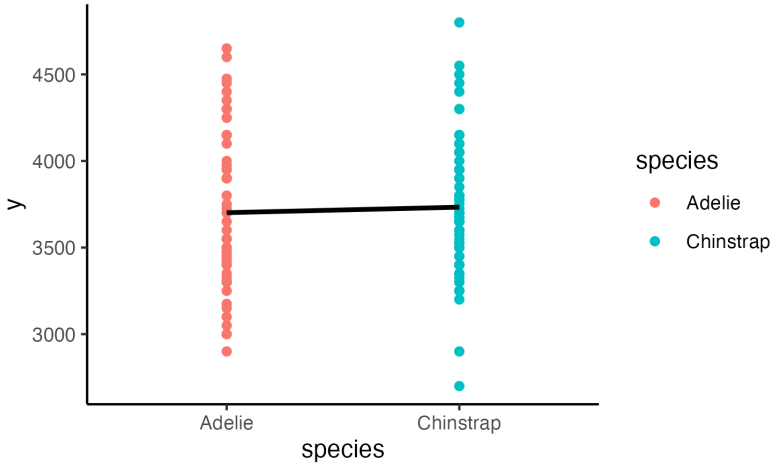
```
# A tibble: 4 × 5
  data                model  fit      AIC  dAIC
  <list>              <list> <list> <dbl> <dbl>
1 <tibble [123 × 10]> <fn>   <lm>   1824. 30.2
2 <tibble [123 × 10]> <fn>   <lm>   1836. 41.6
3 <tibble [123 × 10]> <fn>   <lm>   1796.  2.48
4 <tibble [123 × 10]> <fn>   <lm>   1794.  0
```

モデル選択 model selection

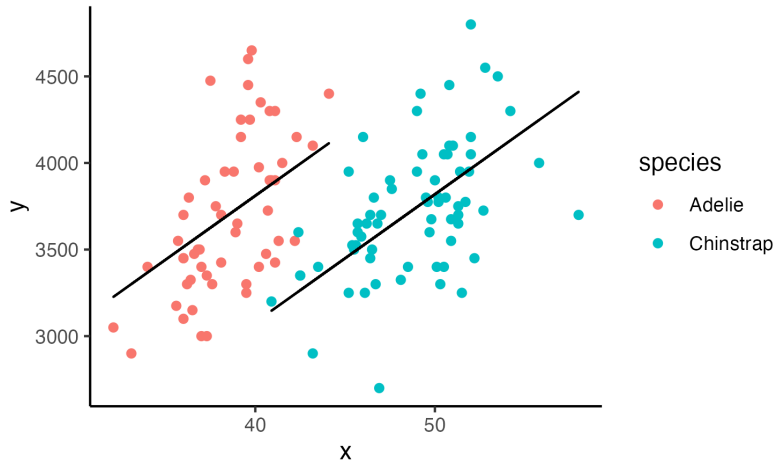
① `y ~ x`



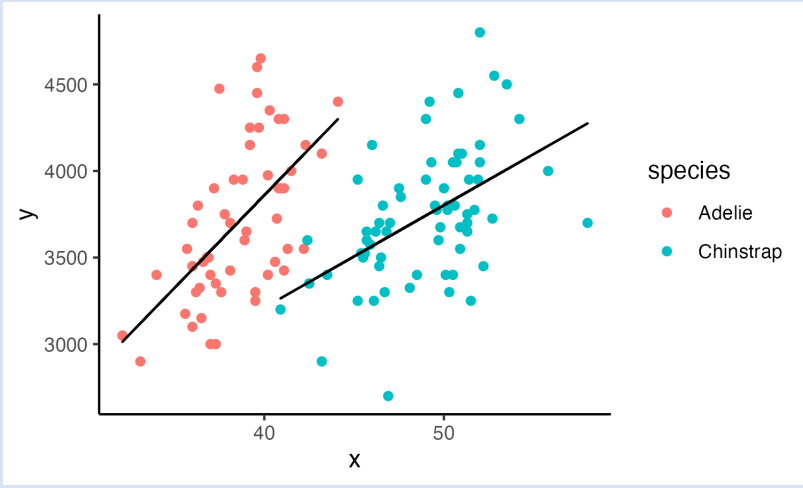
② `y ~ species`



③ `y ~ x + species`



④ `y ~ x * species`



赤池情報量基準 AIC

$$AIC = 2k - 2\ln L$$

パラメータ数　　最大尤度

同程度の誤差を持つモデル同士であれば、
パラメータ数のより少ないモデルが「良い」

同一のデータに対してAICが小さいモデル
→ 新たなデータに対する予測精度が高い

モデル選択 model selection

```
# modelの関数をリストにしておく
models <-
  list(
    function(data){lm(y ~ x, data = data)},
    function(data){lm(y ~ species, data = data)},
    function(data){lm(y ~ x + species, data = data)},
    function(data){lm(y ~ x * species, data = data)}
  )
```

モデル選択 model selection

```
library(lme4)

# modelの関数をリストにしておく
models <-
  list(
    function(data){lm(y ~ x, data = data)},
    function(data){lm(y ~ species, data = data)},
    function(data){lm(y ~ x + species, data = data)},
    function(data){lm(y ~ x * species, data = data)},
    function(data){lmer(y ~ x + (x|species), data = data)},
    function(data){lmer(y ~ x + (1|species), data = data)},
    function(data){lmer(y ~ x + (0 + x|species), data = data)}
  )
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y))) %>%
  mutate(AIC = purrr::map_dbl(fit, ~ AIC(.))) %>%
  mutate(dAIC = AIC - min(AIC))
```

```
# A tibble: 7 × 5
  data                model  fit          AIC  dAIC
  <list>              <list> <list>     <dbl> <dbl>
1 <tibble [123 × 10]> <fn>   <lm>       1824. 39.7
2 <tibble [123 × 10]> <fn>   <lm>       1836. 51.1
3 <tibble [123 × 10]> <fn>   <lm>       1796. 12.0
4 <tibble [123 × 10]> <fn>   <lm>       1794.  9.48
5 <tibble [123 × 10]> <fn>   <lmerMod>  1787.  2.30
6 <tibble [123 × 10]> <fn>   <lmerMod>  1787.  2.54
7 <tibble [123 × 10]> <fn>   <lmerMod>  1785.  0
```

線形混合モデル liner mixed model

線形モデル

$$y_i = \alpha_0 + \alpha_1 x_i + u_i$$

線形混合モデル

$$y_i = (\alpha_0 + \alpha_{0j}) + (\alpha_1 + \alpha_{1j}) x_i + u_i$$

$$\alpha_{0j} \sim \mathcal{N}(0, \sigma^2_{\alpha 0}) \qquad \alpha_{1j} \sim \mathcal{N}(0, \sigma^2_{\alpha 1})$$

線形混合モデル liner mixed model

固定効果

$$y_i = (\alpha_0 + \alpha_{0j}) + (\alpha_1 + \alpha_{1j})x_i + u_i$$

$$\alpha_{0j} \sim \mathcal{N}(0, \sigma^2_{\alpha 0}) \qquad \alpha_{1j} \sim \mathcal{N}(0, \sigma^2_{\alpha 1})$$

混合効果

線形混合モデル liner mixed model

固定効果

$$y_i = (\alpha_0 + \alpha_{0j}) + (\alpha_1 + \alpha_{1j})x_i + u_i$$

$$(\alpha_{0j}, \alpha_{1j}) \sim biN(0, \Sigma_\alpha^2)$$

混合効果

線形混合モデル liner mixed model

```
y ~ x + (x|species)
```

$$y_i = (\alpha_0 + \alpha_{0j}) + (\alpha_1 + \alpha_{1j})x_i + u_i$$

```
y ~ x + (1|species)
```

$$y_i = (\alpha_0 + \alpha_{0j}) + \alpha_1 x_i + u_i$$

```
y ~ x + (0 + x|species)
```

$$y_i = \alpha_0 + (\alpha_1 + \alpha_{1j})x_i + u_i$$

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y))) %>%
  mutate(AIC = purrr::map_dbl(fit, ~ AIC(.))) %>%
  mutate(dAIC = AIC - min(AIC))
```

```
# A tibble: 7 × 5
  data                model  fit          AIC  dAIC
  <list>              <list> <list>     <dbl> <dbl>
1 <tibble [123 × 10]> <fn>   <lm>       1824. 39.7
2 <tibble [123 × 10]> <fn>   <lm>       1836. 51.1
3 <tibble [123 × 10]> <fn>   <lm>       1796. 12.0
4 <tibble [123 × 10]> <fn>   <lm>       1794.  9.48
5 <tibble [123 × 10]> <fn>   <lmerMod>  1787.  2.30
6 <tibble [123 × 10]> <fn>   <lmerMod>  1787.  2.54
7 <tibble [123 × 10]> <fn>   <lmerMod>  1785.  0
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y))) %>%
  mutate(AIC = purrr::map_dbl(fit, ~ AIC(.))) %>%
  mutate(dAIC = AIC - min(AIC)) %>%
  mutate(pred = map(fit, predict))
```

```
# A tibble: 7 × 6
  data                 model  fit          AIC  dAIC pred
  <list>               <list> <list>     <dbl> <dbl> <list>
1 <tibble [123 × 10]> <fn>   <lm>       1824. 39.7  <dbl [123]>
2 <tibble [123 × 10]> <fn>   <lm>       1836. 51.1  <dbl [123]>
3 <tibble [123 × 10]> <fn>   <lm>       1796. 12.0  <dbl [123]>
4 <tibble [123 × 10]> <fn>   <lm>       1794.  9.48 <dbl [123]>
5 <tibble [123 × 10]> <fn>   <lmerMod>  1787.  2.30 <dbl [123]>
6 <tibble [123 × 10]> <fn>   <lmerMod>  1787.  2.54 <dbl [123]>
7 <tibble [123 × 10]> <fn>   <lmerMod>  1785.  0    <dbl [123]>
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y))) %>%
  mutate(AIC = purrr::map_dbl(fit, ~ AIC(.))) %>%
  mutate(dAIC = AIC - min(AIC)) %>%
  mutate(pred = map(fit, predict))
```

```
# A tibble: 7 × 6
  data                 model  fit          AIC  dAIC pred
  <list>               <list> <list>     <dbl> <dbl> <list>
1 <tibble [123 × 10]> <fn>   <lm>       1824. 39.7  <dbl [123]>
2 <tibble [123 × 10]> <fn>   <lm>       1836. 51.1  <dbl [123]>
3 <tibble [123 × 10]> <fn>   <lm>       1796. 12.0  <dbl [123]>
4 <tibble [123 × 10]> <fn>   <lm>       1794.  9.48 <dbl [123]>
5 <tibble [123 × 10]> <fn>   <lmerMod>  1787.  2.30 <dbl [123]>
6 <tibble [123 × 10]> <fn>   <lmerMod>  1787.  2.54 <dbl [123]>
7 <tibble [123 × 10]> <fn>   <lmerMod>  1785.  0    <dbl [123]>
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y))) %>%
  mutate(AIC = purrr::map_dbl(fit, ~ AIC(.))) %>%
  mutate(dAIC = AIC - min(AIC)) %>%
  mutate(pred = map(fit, predict)) %>%
  select(data, pred)

# A tibble: 7 × 2
  data                pred
  <list>              <list>
1 <tibble [123 × 10]> <dbl [123]>
2 <tibble [123 × 10]> <dbl [123]>
3 <tibble [123 × 10]> <dbl [123]>
4 <tibble [123 × 10]> <dbl [123]>
5 <tibble [123 × 10]> <dbl [123]>
6 <tibble [123 × 10]> <dbl [123]>
7 <tibble [123 × 10]> <dbl [123]>
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y))) %>%
  mutate(AIC = purrr::map_dbl(fit, ~ AIC(.))) %>%
  mutate(dAIC = AIC - min(AIC)) %>%
  mutate(pred = map(fit, predict)) %>%
  select(data, pred) %>%
  rowid_to_column("model")
```

```
# A tibble: 7 × 3
  model data                pred
  <int> <list>              <list>
1     1 <tibble [123 × 10]> <dbl [123]>
2     2 <tibble [123 × 10]> <dbl [123]>
3     3 <tibble [123 × 10]> <dbl [123]>
4     4 <tibble [123 × 10]> <dbl [123]>
5     5 <tibble [123 × 10]> <dbl [123]>
6     6 <tibble [123 × 10]> <dbl [123]>
7     7 <tibble [123 × 10]> <dbl [123]>
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y))) %>%
  mutate(AIC = purrr::map_dbl(fit, ~ AIC(.))) %>%
  mutate(dAIC = AIC - min(AIC)) %>%
  mutate(pred = map(fit, predict)) %>%
  select(data, pred) %>%
  rowid_to_column("model") %>%
  mutate(model = str_c("model", model))
```

```
# A tibble: 7 × 3
  model  data                pred
  <chr>  <list>              <list>
1 model1 <tibble [123 × 10]> <dbl [123]>
2 model2 <tibble [123 × 10]> <dbl [123]>
3 model3 <tibble [123 × 10]> <dbl [123]>
4 model4 <tibble [123 × 10]> <dbl [123]>
5 model5 <tibble [123 × 10]> <dbl [123]>
6 model6 <tibble [123 × 10]> <dbl [123]>
7 model7 <tibble [123 × 10]> <dbl [123]>
```

モデル選択 model selection

```
df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y))) %>%
  mutate(AIC = purrr::map_dbl(fit, ~ AIC(.))) %>%
  mutate(dAIC = AIC - min(AIC)) %>%
  mutate(pred = map(fit, predict)) %>%
  select(data, pred) %>%
  rowid_to_column("model") %>%
  mutate(model = str_c("model", model)) %>%
  unnest(everything())
```

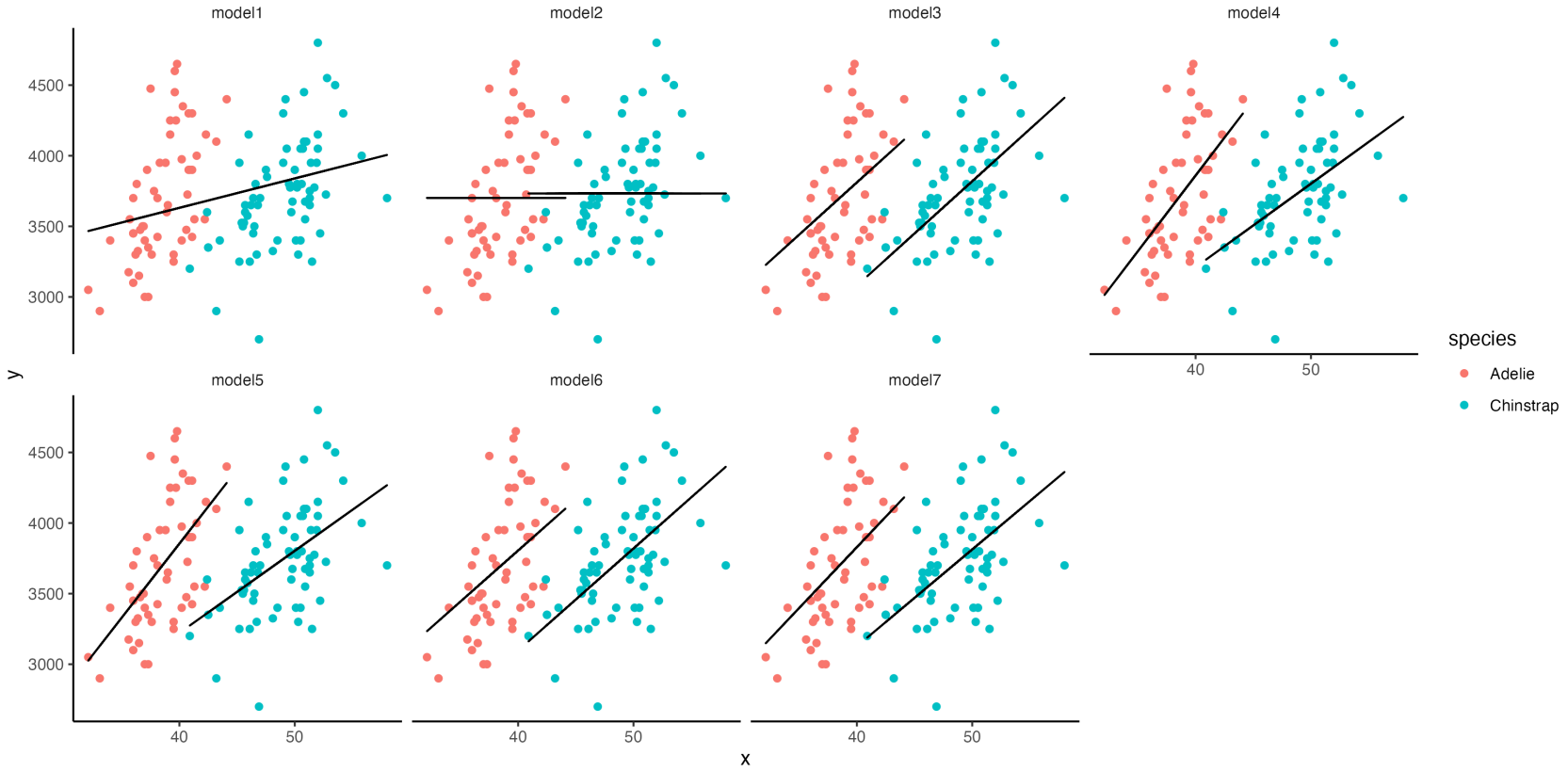
```
# A tibble: 861 × 12
   model  species island bill_length_mm bill_depth_mm flipper_length_mm
   <chr>  <fct>   <fct>           <dbl>         <dbl>             <int>
 1 model1 Adelie  Dream            39.5          16.7               178
 2 model1 Adelie  Dream            37.2          18.1               178
 3 model1 Adelie  Dream            39.5          17.8               188
 4 model1 Adelie  Dream            40.9          18.9               184
 5 model1 Adelie  Dream            36.4          17                 195
```

モデル選択 model selection

```
df_fit <-
  df_for_aov %>%
  tidyr::nest() %>%
  mutate(model = list(models)) %>%
  unnest(model) %>%
  mutate(fit = purrr::map2(model, data, ~.x(.y))) %>%
  mutate(AIC = purrr::map_dbl(fit, ~ AIC(.))) %>%
  mutate(dAIC = AIC - min(AIC))

df_pred <-
  df_fit %>%
  mutate(pred = map(fit, predict)) %>%
  select(data, pred) %>%
  rowid_to_column("model") %>%
  mutate(model = str_c("model", model)) %>%
  unnest(everything())
```

モデル選択 model selection

```
ggplot(data = df_pred) +
  aes(x, y) +
  geom_point(mapping = aes(color = species)) +
  geom_path(mapping = aes(y = pred, group = species)) +
  facet_wrap(~model, nrow = 2)
```



- 確率の話
- 回帰モデルの話
- 分散分析の話
- 線形混合モデルの話

Day1

- データ科学とは
- データ科学のツール
- Rを始めよう

  基礎知識、データの読み書き

  データの操作、データ可視化

Day2

- 確率の話
- 回帰モデルの話
- 分散分析の話
- 線形混合モデルの話

課題

penguinsデータの全てまたは**ペンギン3種**を含む一部を用い、任意の2つの連続量の相関関係をペンギン種について考慮しながら考察・可視化して下さい。

提出 タイトルに**[R研修]** 宛先: **kmimura@ism.ac.jp**

結果をまとめたスライド資料(15枚以内)

スライドを使って説明した動画(10分間以内)

※ 後日、研修参加者に限定して共有します。

締切

2024年1月15日17:00