

9. 分析結果の提示方法

honocat

2025-12-31

```
HR09 <- read_csv('data/hr-data.csv') |>
  mutate(experience = as.numeric(status == '現職' | status == '元職')) |>
  filter(year == 2009,
        !is.na(expm))
```

4つの回帰モデルを推定しておく。

```
fit1 <- lm(voteshare ~ experience, data = HR09)
fit2 <- lm(voteshare ~ expm, data = HR09)
fit3 <- lm(voteshare ~ experience + expm, data = HR09)
fit4 <- lm(voteshare ~ experience * expm, data = HR09)
```

分析結果の提示

式を書く

説明変数の数がそれほど多くない場合は、式で結果を示しても良い。

例えば、fit1 の推定結果は、次の式にまとめられる。

```
tidy(fit1)
```

```
# A tibble: 2 x 5
  term      estimate std.error statistic   p.value
  <chr>     <dbl>    <dbl>     <dbl>    <dbl>
1 (Intercept)  13.9     0.622    22.4  5.84e- 92
2 experience   31.2     0.977    31.9  1.61e-159
```

$$\widehat{\text{得票率}} = 13.91 + 31.18 \cdot \text{議員経験}$$
$$(0.62) \quad (0.98)$$

ただし、カッコ内は標準誤差である。

これに加え、サンプルサイズと決定係数(重回帰なら自由度調整済み決定係数)を表示する必要がある。サンプルサイズは `length(fit1$residuals)` で、決定係数は `summary(fit1)$r.squared`、自由度調整済み決定係数は `summary(fit1)$adj.r.squared` で表示することができる。

さらに、信頼区間も表示することができる。

```
confint(fit1, level = 0.95)
```

```
2.5 % 97.5 %
(Intercept) 12.68800 15.12755
experience 29.26034 33.09466
```

よって、議員経験の回帰係数の 95% 信頼区間は [29.26, 33.09] である。

表を作る

■(1) modelsummary

回帰分析の結果を表にするには、`modelsummary` パッケージの `msummary()`（または、`modelsummary()`）関数を使う。

基本的な使い方は、

```
msummary(fit1,
          title = '得票率を結果変数とする単回帰の推定結果')
```

1. 表のキャプションは上に置きたい
2. 知らない統計量は書かない

そこで、コードを書き直す。

```
msummary(fit1,
          title = '得票率を結果変数とする単回帰の推定結果',
          gof_map = tibble(raw = c('nobs', 'r.squared'),
                           clean = c('N', 'R2')),
          fmt = c(0, 3))) |>
style_tt(bootstrap_class = 'table caption-top')
```

日本語で論文・レポートを書くなら、表の中身も日本語にすべき。

表 1: 得票率を結果変数とする単回帰の推定結果

	(1)
(Intercept)	13.908 (0.622)
experience	31.178 (0.977)
Num.Obs.	1124
R2	0.476
R2 Adj.	0.475
AIC	9437.7
BIC	9452.8
Log.Lik.	-4715.857
RMSE	16.07

表 2: 得票率を結果変数とする単回帰の推定結果

	(1)
(Intercept)	13.908 (0.622)
experience	31.178 (0.977)
N	1124
R ²	0.476

```
msummary(fit1,
  title = '得票率を結果変数とする単回帰の推定結果',
  coef_rename = c('切片', '議員経験'),
  gof_map = tibble(raw = c('nobs', 'r.squared'),
    clean = c('R^2', '観測数'),
    fmt = c(3, 0))) |>
style_tt(bootstrap_class = 'table caption-top')
```

表 3: 得票率を結果変数
とする単回帰の推定結果

	(1)
切片	13.908
	(0.622)
議員経験	31.178
	(0.977)
R ²	1124.000
観測数	0

複数のモデルを 1 つの表にまとめたいときは以下のようにする。

```
msummary(list(`モデル 1` = fit1,
  `モデル 2` = fit2,
  `モデル 3` = fit3,
  `モデル 4` = fit4),
  title = '得票率を結果変数とする単回帰の推定結果',
  coef_rename = c('切片', '議員経験', '選挙費用(百万円)',
    '議員経験 x 選挙費用'),
  gof_map = tibble(raw = c('r.squared', 'adj.r.squared', 'nobs'),
    clean = c('R^2',
      '自由度調節済み R^2',
      '観測数'),
    fmt = c(3, 3, 0))) |>
style_tt(bootstrap_class = 'table caption-top')
```

■(2) texreg

図を作る

キャタピラプロットと呼ばれる図を使って、回帰分析の結果を図示しよう。

1 つのモデルを図示するときは、`coefplot::coefplot()` を使う。チャンクオプションに `fig.cap: 'caption'` を加える。

```
plt_fit3 <- coefplot(
  fit3,
```

表 4: 得票率を結果変数とする単回帰の推定結果

	モデル 1	モデル 2	モデル 3	モデル 4
切片	13.908 (0.622)	7.735 (0.757)	7.891 (0.691)	-2.096 (0.713)
議員経験	31.178 (0.977)		18.371 (1.226)	46.192 (1.568)
選挙費用(百万円)		3.072 (0.096)	1.831 (0.120)	4.870 (0.164)
議員経験 x 選挙費用				-4.774 (0.206)
R^2	0.476	0.478	0.565	0.706
自由度調節済み R^2	0.475	0.478	0.565	0.705
観測数	1124	1124	1124	1124

```

lwdOuter = 1,
intercept = FALSE,
title     = '係数の推定値:結果変数は得票率(%)',
xlab      = '係数の推定値',
ylab      = '説明変数',
newNames  = c(experience = '議員経験', expm = '選挙費用(百万円)')
)
plot(plt_fit3)

```

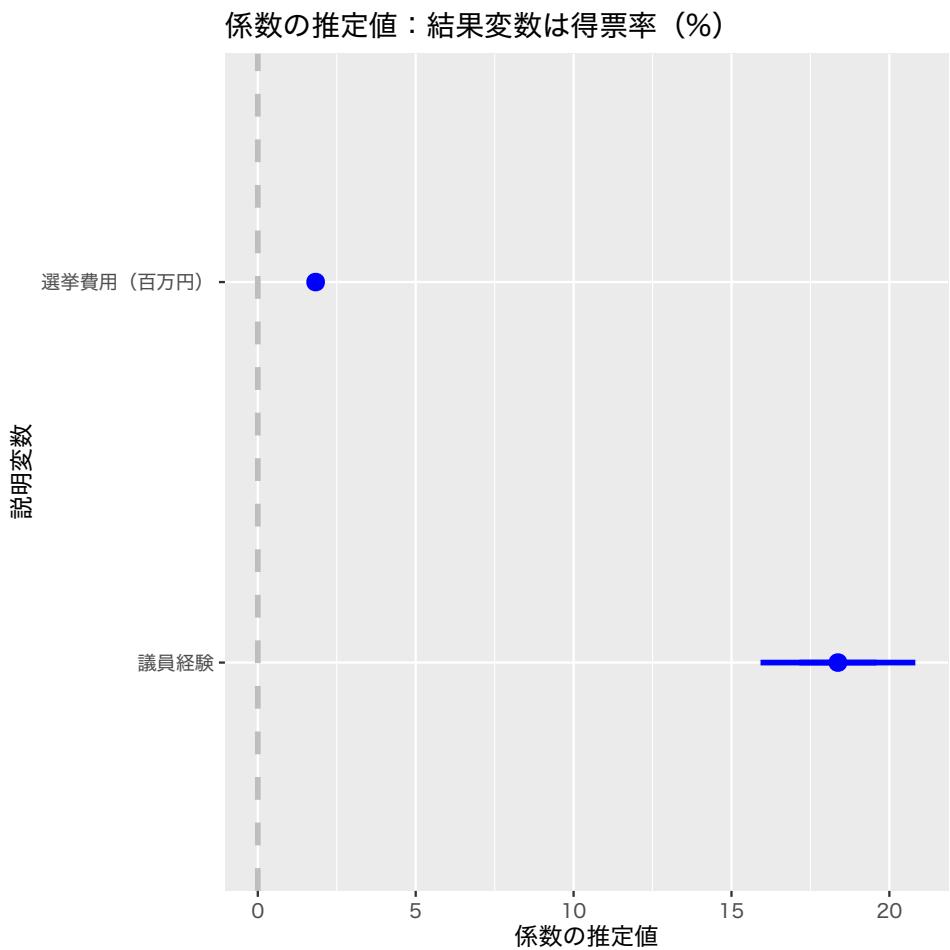
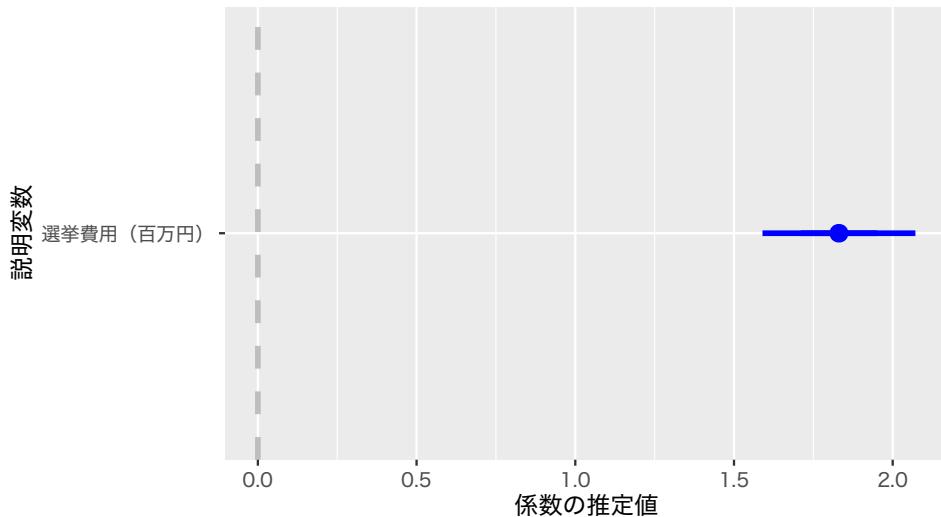


図 1: モデル 3 の推定結果

回帰係数の大きさがバラバラな場合、変数を非表示にできる。例えば、議員経験は交絡変数として含めたとしよう。

```
plt_fit3b <- coefplot(
  fit3,
  lwdOuter      = 1,
  coefficients = c('expm'),
  intercept    = FALSE,
  title         = '係数の推定値：結果変数は得票率(%)',
  xlab          = '係数の推定値',
  ylab          = '説明変数',
  newNames     = c(expm = '選挙費用(百万円)')
)
plot(plt_fit3b)
```

係数の推定値：結果変数は得票率 (%)



複数のモデルを 1 つの図に示すときは、`coefplot::multiplot()` を使う。

```
plt_multi <- multiplot(
  fit1, fit2, fit3, fit4,
  intercept = FALSE,
  numberAngle = 0,
  title      = ' 係数の推定値 : 結果変数は得票率(%) ',
  xlab       = ' 係数の推定値',
  ylab       = ' 説明変数',
  newNames   = c(experience      = '議員経験',
                 expm           = '選挙費用(百万円)',
                 'experience:expm' = '議員経験 x 選挙費用') +
  scale_color_brewer(palette = 'Set1',
                     name    = '',
                     labels  = paste(' モデル', 1 : 4))
)
plot(plt_multi)
```

係数の推定値：結果変数は得票率（%）

