5. 回帰分析による統計的推定

honocat

2025-12-28

回帰分析のシミュレーション

結果変数 Y と説明変数 X の**真の関係**（つまり、母集団における関係）が以下の式で表されるとする。

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

ここで、β_0 が y 切片、β_1 が回帰直線の傾きである。ε は誤差項と呼ばれるもので、$\epsilon \sim \text{Normal}(0, \sigma)$ であり、σ は正規分布（normal distribution）の標準偏差である。つまり、誤差 ε_i は平均 0、標準偏差 σ の正規分布に従う。

この関係は、次のように書くこともできる。

$$Y_i \sim \text{Normal}(\beta_0 + \beta_1 X_i, \sigma)$$

例として、$\beta_0 = 2, \beta_1 = 0.8$ の場合について考える。このとき、

$$Y_i = 2 + 0.8 X_i + \varepsilon_i$$

と表せる。

回帰分析では、観測された Y と X の値から、β_0 と β_1 の値を推定することになる。以下のシミュレーションでは、回帰分析による推定が、$\beta_0 = 2, \beta_1 = 0.8$ という値にどれだけ近い値を出せるかどうかを確かめる。

シミュレーションの方法

シミュレーションを行うために、データを生成する。私たちは真の関係を知っているので、その関係を利用する。

標本サイズ N は、試しに 5 とする。

```
N <- 5
```

次に X の値を決める。とりあえず、$[-5, 5]$ の一様分布から X の実現値（観測値）x をランダムに作ってみる。

```
x <- runif(N, min = -5, max = 5)
```

続いて、Yの実現値 y を生成する。真の関係は $Y = 2 + 0.8X + \varepsilon$ である。

まず、切片と傾きの値を設定する。

```
beta0 <- 2
beta1 <- 0.8
```

次に、ε を作る。$\varepsilon \sim \text{Normal}(0, \sigma)$ なので、誤差項の標準偏差 σ を決める必要がある。ここでは、$\sigma = 2$ とする。

```
sigma <- 2
```

この標準偏差を使って、ε をランダムに生成する。

```
epsilon <- rnorm(N, mean = 0, sd = sigma)
```

これで y が生成できる。

```
y <- beta0 + beta1 * x + epsilon
```

X の観測値 x と Yの観測値 y が手に入ったので、回帰分析を実行する。

```
df <- tibble(y = y, x = x)
fit <- lm(y ~ x, data = df)
broom::tidy(fit) |>
  mutate_if(is.double, round, digits = 2)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic p.value
  <chr>          <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)     3.34      0.44      7.59    0
2 x               0.91      0.13      6.76    0.01
```

β_0, β_1 の推定値をそれぞれ b_0, b_1 とすると、$b_0 = 3.34$, $b_1 = 0.91$ である。推定はどれくらい正確か？

複数回のシミュレーション

関数を作って、複数回実行する。

```
simple_reg <- function(n, beta0 = 0, beta1 = 1, sigma = 1) {
  ## 単回帰のシミュレーションを実行するための関数
  ## 引数: n      = 標本サイズ
  ##       beta0 = 真の y 切片
  ##       beta1 = 真の傾き
  ##       sigma = 誤差項の標準偏差

  # x を一様分布 Uniform(-5, 5) から作る
  x <- runif(n, min = -5, max = 5)

  # epsilon を正規分布 N(, sigma ^ 2) から作る
  epsilon <- rnorm(n, mean = 0, sd = sigma)

  # 真のモデルから y を作る
  y <- beta0 + beta1 * x + epsilon

  # 回帰分析を実行する
  fit <- lm(y ~ x)

  # beta の推定値を関数の出力として返す
  return(coef(fit))
}
```

試しに、この関数を使ってみる。私たちが実行したいのは $N = 5, \beta_0 = 2, \beta_1 = 0.8, \sigma = 2$ の場合なので、次のようにする。

```
simple_reg(n = 5,
           beta0 = 2, beta1 = 0.8,
           sigma = 2)
```

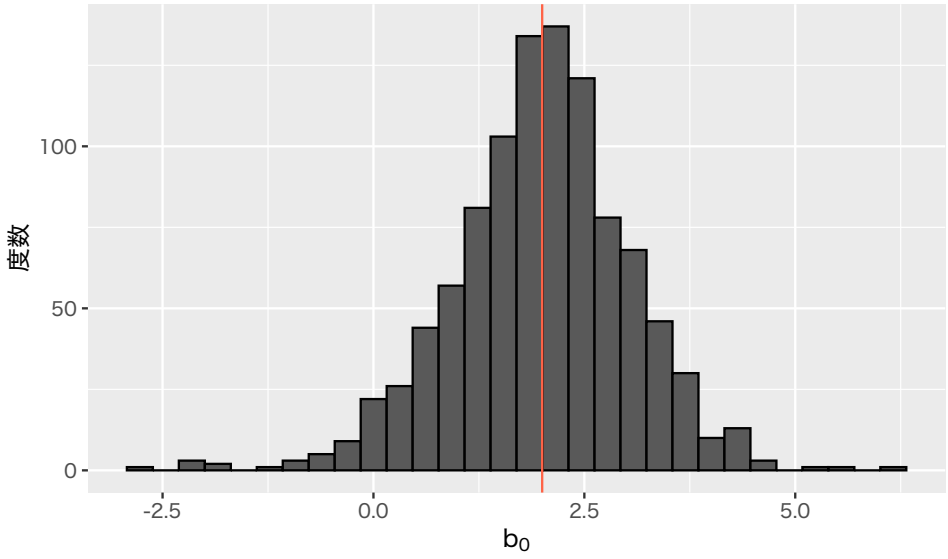
```
(Intercept)           x
   1.282906    1.161380
```

これを繰り返し実行すれば、最小二乗法がどれくらい正確に推定を行えるか理解することができるはず。

```
n_sims <- 1000
result <- matrix(NA, nrow = n_sims, ncol = 2)
colnames(result) <- c('b0', 'b1')
for (i in 1 : n_sims) {
  result[i,] <- simple_reg(n = 5,
                           beta0 = 2, beta1 = 0.8,
                           sigma = 2)
}
```

まず、β_0 の推定値である、b_0（b0)をヒストグラムにする。

```
res_data <- as_tibble(result)
hist_b0 <- ggplot(data = res_data, aes(x = b0)) +
  geom_histogram(color = 'black') +
  labs(x = expression(b[0]), y = '度数') +
  geom_vline(xintercept = 2, color = 'tomato')
plot(hist_b0)
```



分布の形に注目すると、分布の中心は真の値付近にあり、**平均すると**推定がうまく行っているように見える。

実際、1,000 個得られた b0 の平均値を求めると、

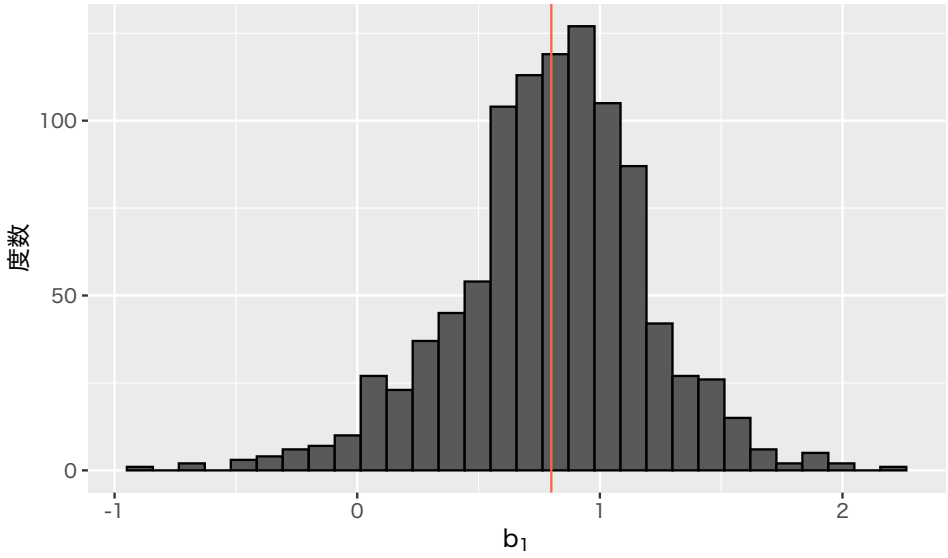
```
mean(res_data$b0)
```

```
[1] 1.987918
```

であり、真の値に近い。

同様に、β_1 の推定値である、b_1（b1）をヒストグラムにする。

```
hist_b1 <- ggplot(data = res_data, aes(x = b1)) +
  geom_histogram(color = 'black') +
  labs(x = expression(b[1]), y = '度数') +
  geom_vline(xintercept = 0.8, color = 'tomato')
plot(hist_b1)
```



やはり、**平均すると**推定がうまくいっているように見える。

実際、b1 の平均値は、

```
mean(res_data$b1)
```

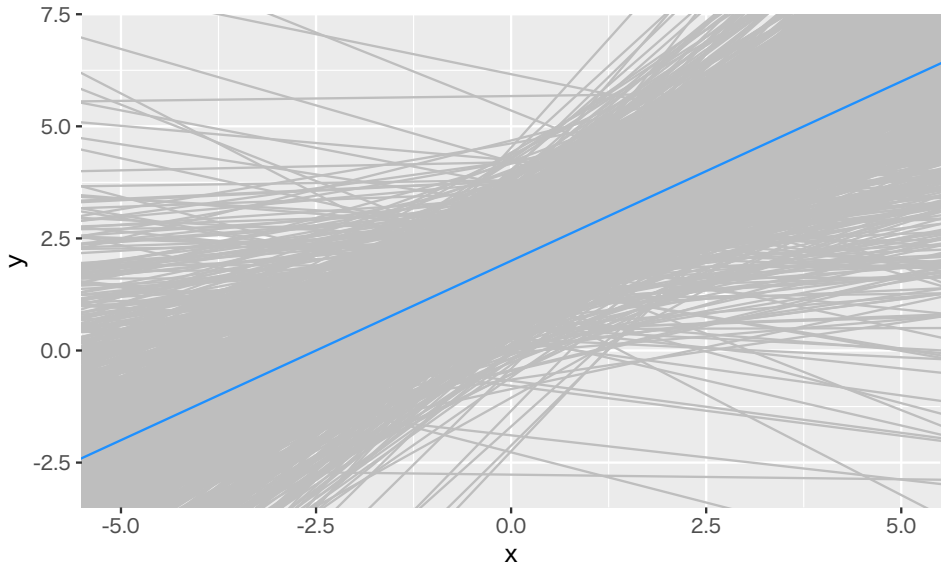
```
[1] 0.7978576
```

であり、真の値に近い。

最後に、得られた回帰直線を図示する。

```
plt <- ggplot(NULL) +
  geom_abline(intercept = res_data$b0,
              slope     = res_data$b1,
              color     = 'gray') +
  geom_abline(intercept = 2,
              slope     = 0.8,
```

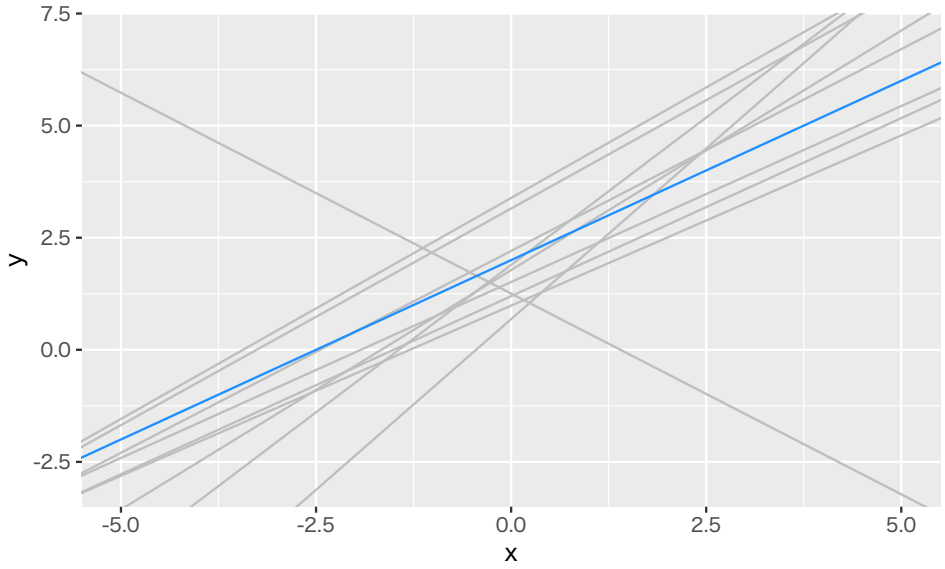
```
              color     = 'dodgerblue') +
  xlim(-5, 5) + ylim(-3, 7) +
  labs(x = 'x', y = 'y')
plot(plt)
```



平均的には、回帰分析はうまくいきそう。

回帰直線をランダムに 10 個だけ選んで描いてみる。

```
res_data_sub <- slice_sample(res_data, n = 10)
plt_sub10 <- ggplot(NULL) +
  geom_abline(intercept = res_data_sub$b0,
              slope     = res_data_sub$b1,
              color     = 'gray') +
  geom_abline(intercept = 2,
              slope     = 0.8,
              color     = 'dodgerblue') +
  xlim(-5, 5) + ylim(-3, 7) +
  labs(x = 'x', y = 'y')
plot(plt_sub10)
```



実際のデータ分析では、1 つ(または少数)のデータセットを対象に分析を行うことが多い。つまり、グレーの直線のうちどれか 1 つ(または少数)だけが得られることに成る。その直線は、X と Y の真の関係を捉えているとは言えないことが、今回のシミュレーションでわかった。私たちは、標本(サンプル)から得られた 1 つの直線を手がかりにして統計的検定や統計的推定を行い、母集団(真の関係)についての理解を深めることを目指すことになる。