2. セレクションバイアス

honocat

2026-01-02

セレクションバイアスのシミュレーション

メールによる販促の効果

セレクションの例として、メールによる販売促進キャンペーンについて考える。ある商品を売りたい企業は、その商品の販促メールを特定の人たちに送り、そのメール(原因)がその商品の売れ行き(結果)に影響を与えたかどうかを知りたいとする。消費者についてみると、メールを受け取るかどうか(処置)によって、商品を買うかどうか(結果)に影響があるかどうかという因果効果に関心があるとする。

まず、対象全体の人数を決める。

```
N <- 1000
```

次に、N 人をメールがないときに商品を買いやすいグループ A と、商品をあまり買わないグループ B に分ける。ここでは、同じ人数に分けることにする。

```
d1 <- tibble(group = rep(c('A', 'B'), each = N / 2))
```

グループ A と B が、メールを受け取らなかったときに商品を買う確率 p_A (pA)と p_B (pB)を決める。ただし、pA>pB である。

```
pA <- 0.6
pB <- 0.3
```

次に、誰にメールを送るかを決めよう。ここでは、企業側が元々商品を買いそうな人にメールを送りやすいというサンプルセレクションが存在すると考えて、各グループのメンバーがメールを受け取る確率をそれぞれ $\frac{p_A}{(p_A+p_B)}, \frac{p_B}{(p_A+p_B)}$ としよう。また、メールは全部で $\frac{N}{2}$ 人に送ることにする。メールを受け取れば 1、受け取らなければ 0 をとる mail という変数を作る。

```
n_mail <- N / 2
n_receive <- round(n_mail * c(pA, pB) / sum(pA, pB))
d1$mail <- rep(c(1, 0, 1, 0),
               times = c(n_receive[1], N / 2 - n_receive[1],
                         n_receive[2], N / 2 - n_receive[2]))
```

メールを送ることによって商品を購入する確率がどれくらい上がるか、つまり、メールの効果である tau（τ）の値を決める。ここでは、0.05 ポイント上昇することに使用。ここで設定する値が本当の因果効果である。

```
tau <- 0.05
```

メールの送信が終わったあとの購買行動を決める。まず、メール送信後の各人の購買確率を計算する。A の購買確率はメールを受け取らなければ pA、受け取れば pA + tau である。同様に、B の購買確率はメールを受け取らなければ pB、受け取れば pB + tau である。

```
d1 <- d1 |>
  mutate(p_after = ifelse(group == 'A',
                          pA + tau * mail,
                          pB + tau * mail))
```

購買確率によって、商品を買うかどうかをベルヌーイ（Bernoulli）試行で決める。つまり、確率 p で表が出るコインを投げて、表が出たら購入し、裏が出たら購入しないと考える。Bernoulli(p) = Binomial$(1, p)$ なので、R でベルヌーイ試行を実施するには、二項分布（binomial distribution）から乱数を生成する `rbinom()` を `size = 1` にして使えばいい。

```
set.seed(2026)
d1 <- d1 |>
  mutate(purchase = rbinom(N, size = 1, prob = p_after))
```

メール受信状況別の購入割合は、

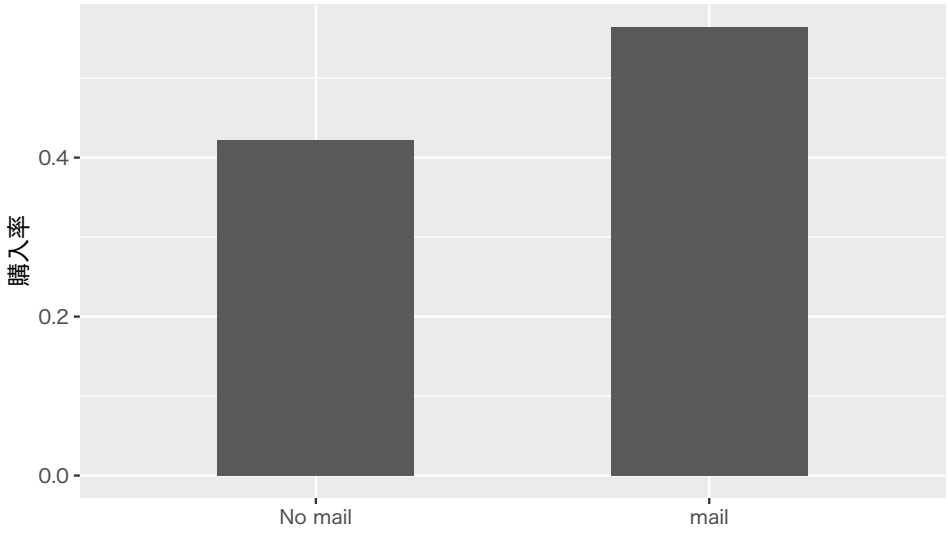
```
d1_gr <- d1 |>
  mutate(mail = factor(mail, labels = c('No mail', 'mail'))) |>
  group_by(mail) |>
  summarize(purchase_rate = mean(purchase),
            .groups       = 'drop') |>
  print()
```

```
# A tibble: 2 x 2
  mail    purchase_rate
```

```
  <fct>           <dbl>
1 No mail         0.422
2 mail            0.564
```

である。図にすると、

```
p1 <- ggplot(d1_gr, aes(x = mail, weight = purchase_rate)) +
  geom_bar(width = 0.5) +
  labs(x = '', y = '購入率')
plot(p1)
```



となる。単純に比較すると、

```
d1_gr |>
  pull(purchase_rate) |>
  diff()
```

```
[1] 0.142
```

がメールの効果のように見えてしまう。しかし、実際のメールの効果は、先程設定した通り 0.05 である。

ちなみに、この「バイアスを含む効果」は単回帰によっても得られる。

```
fit <- lm(purchase ~ mail, data = d1)
broom::tidy(fit)
```

```
# A tibble: 2 x 5
```

```
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    0.422    0.0222     19.0  3.05e-69
2 mail           0.142    0.0313      4.53 6.54e- 6
```

mail の効果の推定値は 0.14 である。また、その p 値が $6.54 \times 10^6 < 0.001$ なので、この効果は有意水準 0.001 (0.1%)で統計的に有意である。**単に「統計的に有意かどうか」を調べても、因果効果はわからない**ということがわかる。

1 回のシミュレーションでは、偶然そうなっただけかもしれないので、これを複数回繰り返す。

```
sim_mail <- function(tau = 0.05, N = 1000, n_mail = N / 2,
                     pA = 0.6, pB = 0.3) {
  if (pA <= pB) stop('pA must be larger than pB.')
  if (N < n_mail) stop('N must be larger than n_mail.')
  if (pA > 1 | pB < 0) stop('pA and pB must be in the range [0, 1].')
  if (tau + pA > 1) stop('beta + pA must be equal to or smaller than 1.')
  if (tau + pB < 0) stop('beta + pB must be equal to or greater than 0.')

  group <- rep(c('A', 'B'), each = N / 2)
  n_receive <- round(n_mail * c(pA, pB) / sum(pA, pB))
  mail <- rep(c(1, 0, 1, 0),
              times = c(n_receive[1], N / 2 - n_receive[1],
                        n_receive[2], N / 2 - n_receive[2]))
  p_after <- ifelse(group == 'A', pA + tau * mail, pB + tau * mail)
  purchase <- rbinom(N, size = 1, prob = p_after)
  fit <- lm(purchase ~ mail)
  return(coef(fit)[2])
}
```

試しに、tau = 0.05 で一度使ってみる。

```
sim_mail(tau = 0.05)
```

```
mail
0.17
```

引数の値を変えてみる。

```
sim_mail(tau = 0.1, pA = 0.8, pB = 0.7)
```

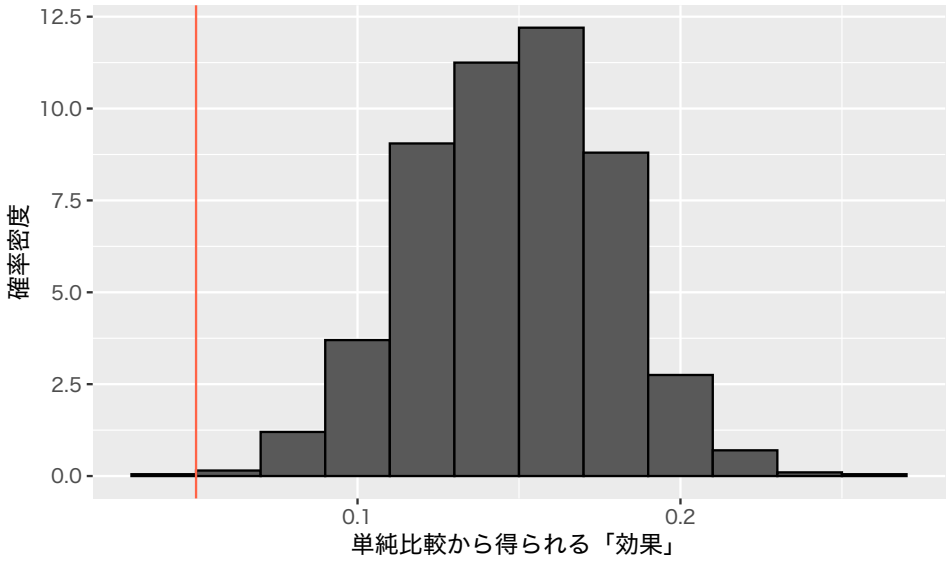
```
 mail
0.088
```

最初と同じ条件で、シミュレーションを 1,000 回繰り返す。

```
s1_1e3 <- replicate(1e3, sim_mail())
```

この結果を可視化する。

```
p_s1 <- tibble(tau = s1_1e3) |>
  ggplot(aes(x = tau, y = after_stat(density))) +
  geom_histogram(color = 'black', binwidth = 0.02) +
  geom_vline(xintercept = 0.05, color = 'tomato') +
  labs(x = '単純比較から得られる「効果」', y = '確率密度')
plot(p_s1)
```



本当の効果は 0.05 なのに、推定された効果の平均値は 0.15、中央値は 0.15 である。

■購買行動のモデリング：発展的内容

注意：以下の内容はセレクションバイアスとはあまり関係ない。

上の説明では、メールによる販促効果が一定であると考えたが、実際には、

- 全く買う気がない人にはあまり効果がない

- 元々買う予定の人にはあまり効果がない
- 買うか買わないか迷っている人には効果が大きい

ということが予想される。そのような効果をモデル化してみる。

個人 i が商品を買うかどうかを表す変数を Y_i とする。$Y_i = 1$ なら購入、$Y_i = 0$ なら非購入とする。Y_i は二値変数なので、個人 i が商品を購入する確率を θ_i として、ベルヌーイ分布で購買モデルを考える事ができる。すなわち、

$$Y_i \sim \text{Bernoulli}(\theta_i)$$

と考える。ここで、購買確率 θ_i は、メールがない場合に商品を買おうと思っていた度合い α と、メールの効果 τ によって決まると考える。α が 0 に近ければ買うかどうか迷っている状態、絶対値が大きい負の値ならほとんど買う気がない状態、大きい正の値ならほぼ買うつもりの状態を表す。商品を買いやすい集団 A と買いにくい集団 B がいるとすると、α も集団ごとに異なると考えられる。そこで、個人 i が属するグループを $G_i \in \{A, B\}$ とすると、この度合いは α_{G_i} と表すことができる。

M_i をメールを受け取ったことを表すダミー変数、つまり、メールを受け取れば $M_i = 1$、受け取らなければ $M_i = 0$ になる変数だとすると、個人 i が商品を買おうとする「度合い（確率ではない）」は、$\alpha_{G_i} + \tau M_i$ と表せる。

この「度合い」は確率ではなく、$(-\infty, \infty)$ の値を取る。確率は $[0, 1]$ でなければならないので、これを変換する必要がある。そのような変換を行うことができる関数の 1 つが、ロジスティック関数（ロジットの逆関数）である。

この関数を使うと、購買確率は、

$$\theta_i = \text{logit}^{-1}(\alpha_{G_i} + \tau M_i)$$

となる、ただし、

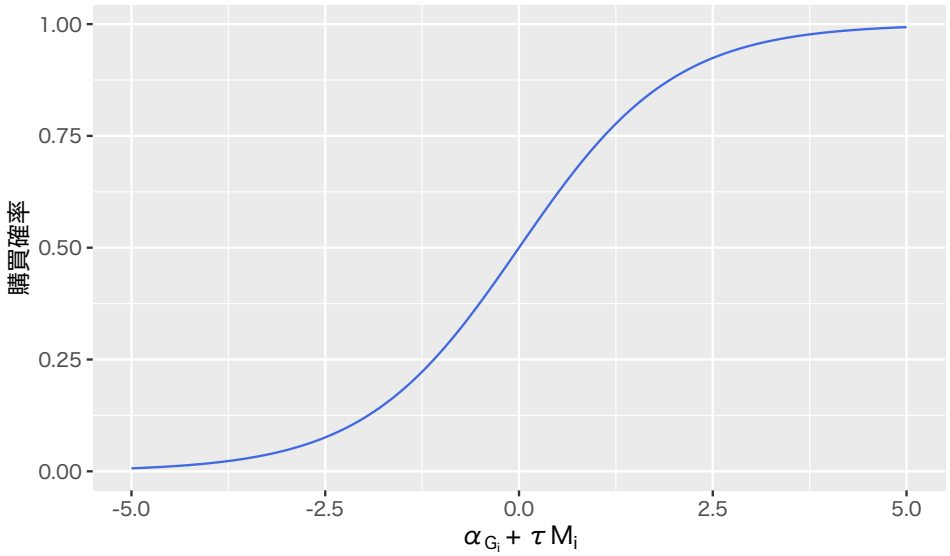
$$\text{logit}^{-1}(x) = \frac{\exp(x)}{1 + \exp(x)} = \frac{1}{1 + \exp(-x)}$$

である。

```
inv_logit <- function(x) {
  return(1 / (1 + exp(-x)))
}
```

これを使うと、τ を 1 つの値に決めても、それが購買確率 θ に与える影響は一定ではなく、α の大きさに依存して影響の大きさが変化することになる。グラフにすると、

```
myd <- tibble(x = seq(from = -5,
                      to   = 5,
                      length.out = 1000)) |>
  mutate(p = inv_logit(x))
p_logistic <- ggplot(myd, aes(x = x, y = p)) +
  geom_line(color = 'royalblue') +
  labs(x = expression(alpha[G[i]] + tau * M[i]),
       y = '購買確率')
plot(p_logistic)
```



グラフが直線ではなく曲線になっており、$\alpha_{G_i} + \tau M_i$ の増分が一定でも、横軸乗でどこにいるかによって変化の大きさが変わる。

このような効果を想定してシミュレーションを行うと、結果は変わるだろうか。

通院と健康状態：セルフセレクション

病院と健康状態の例(Angrist and Pischke, 2009)を使い、自己選択(セルフセレクション)によるセレクションバイアスをシミュレーションによって確かめてみる。

まず、全体の人数 N を決める。

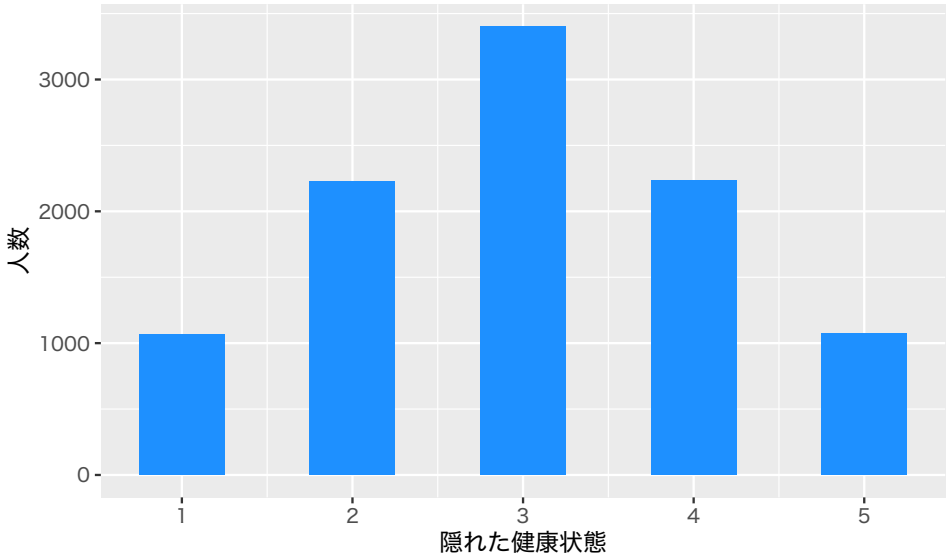
```
N <- 1e4
```

次に、元々の健康状態をランダムに決める。1 が最悪の健康状態、5 が最善の状態とする。中程度の健康状態の人が多いことにする。これは、病院に行く前の「隠れた」健康状態であり、観測されない変数であることに注意。

```
set.seed(2026-1-2)
d2 <- tibble(h_hidden = sample(1 : 5,
                               size    = N,
                               replace = TRUE,
                               prob    = c(1, 2, 3, 2, 1)))
```

分布を確認する。

```
hist_h_hidden <- ggplot(d2, aes(x = h_hidden)) +
  geom_bar(fill = 'dodgerblue', width = 0.5) +
  labs(x = '隠れた健康状態', y = '人数')
plot(hist_h_hidden)
```



この隠れた健康状態に基づき、各個人が病院に行くかどうか決められることにしよう。つまり、各個人が処置である「病院へ行くこと」を、結果である「観測される健康状態」に密接に関連する「隠れた健康状態」という変数に基づいて自己選択(セルフセレクション)するという状況をシミュレートする。

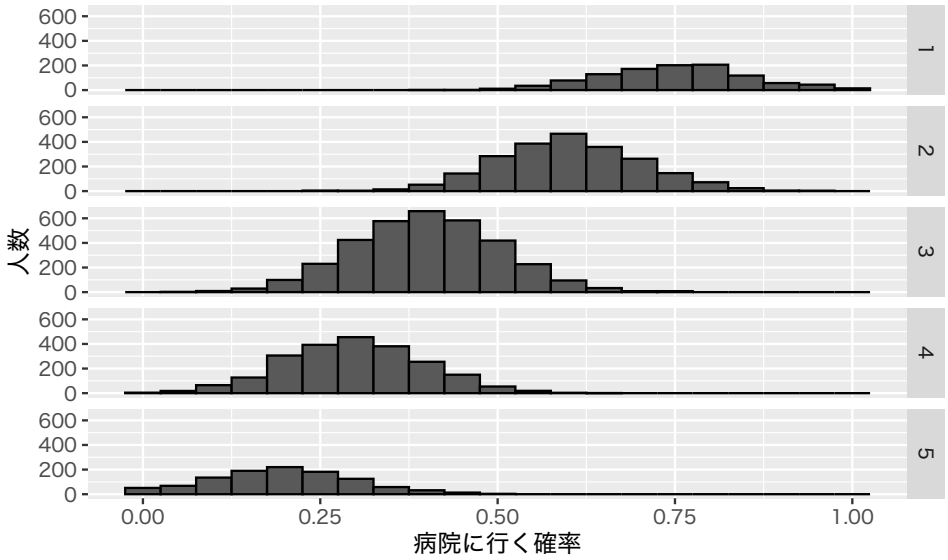
他の条件が等しければ(ceteris paribus)、健康状態が悪いほど病院に行きやすいはずだ。ここではまず、健康状態ごとに病院に行く確率の平均値 μ が異なると考える。そして、各個人が病院に行く確率は、平均値 μ の正規分布からランダムに生成されると考える。話を単純にするため、標準偏差は一定だと仮定する(発展的内容のようにロジット関数を用いても良い)。つまり、健康状態が s の人が病院に行く確率 p_s は、$p_s \sim \text{Normal}(\theta_s, \sigma)$ によって決まる。ただし、$0 \leq p_s \leq 1$ になるように調整する。例として、健康状態ごとの μ_s を、$(\mu_1, \mu_2, \mu_3, \mu_4, \mu_5) = (0.75, 0.6, 0.4, 0.3, 0.2)$ としてみる。σ は 0.1 とする

```
mu <- c(0.75, 0.6, 0.4, 0.3, 0.2)
sigma <- 0.1
```

```
d2 <- d2 |>
  mutate(prob = rnorm(n(), mean = mu[h_hidden], sd = sigma),
         prob = case_when(
           prob > 1 ~ 1,
           prob < 0 ~ 0,
           TRUE ~ prob
         ))
```

健康状態別の通院確率の分布を図示する。

```
hist_prob <- ggplot(d2, aes(x = prob)) +
  geom_histogram(binwidth = 0.05, color = 'black') +
  facet_grid(rows = vars(h_hidden)) +
  labs(x = '病院に行く確率', y = '人数')
plot(hist_prob)
```



健康状態が良い(5 の)場合は、ランダムに生成した「確率」が 0 より小さくなってしまったものを、あとから 0 に調整しているので、0 の度数が正規分布より大きくなってしまっている。したがって、このモデルは通院を説明するものとしてあまり望ましくない。しかし、ここでの目的はセルフセレクションによるセレクションバイアスを理解することなので、これで良しとする。

この確率に基づいて、ベルヌーイ試行で病院に行くかどうかを表す変数 $D_i \in \{0, 1\}$ の値を決める。これが観測される処置の値である。

```
d2 <- d2 |>
  mutate(D = rbinom(n(), size = 1, prob = prob))
```

病院に行く人の割合は、

```
mean(d2$D)
```

```
[1] 0.4365
```

である(現実よりかなり大きな値になっているが、とりあえずこれで進める)。

ここで、通院が健康状態に与える平均処置効果(ATE) beta を設定する。単純化のため、ATE=ITE とする。つまり、処置効果はどの個人にとっても同じだと仮定する。試しに、beta = 0.6 にしてみる。これは、隠れた献供お状態が 3 の人が病院に行くと、健康状態が 3.6 になるということである。実際には、健康状態は 1 から 5 の整数で観測される。隠れた健康状態が 5 の人が病院に行っても健康状態は 5 のままのはずであるが、ここでは 5.6 になることを許そう。どちらも、シミュレーションを単純化するための妥協である。

```
beta <- 0.6
```

この効果を、通院した人にのみ与え、健康状態 Y を観測する。

```
d2 <- d2 |>
  mutate(Y = h_hidden + beta * D)
```

これでデータが揃った。シミュレーションでなければ、観測されるのは D と Y のみである。

病院に行った人と行かなかった人の健康状態を単純に比較してみる。

```
d2_D <- d2 |>
  mutate(D = factor(D, label = c('病院に行かなかった',
                                 '病院に行った'))) |>
  group_by(D) |>
  summarize(health = mean(Y)) |>
  print()
```

```
# A tibble: 2 x 2
  D                  health
  <fct>               <dbl>
1 病院に行かなかった   3.32
2 病院に行った         3.20
```

健康状態は、病院に行った人のほうが、病院に行かなかった人よりも悪いことがわかる。このように、本当の ATE は 0.6 なのに、観測された値を単純比較すると、それが -0.14 のように見えてしまう。

これはシミュレーションなので、本来は計算できないはずのセレクションバイアスも計算することができる。セレクションバイアスは、

$$\mathbb{E}[Y(0)|D=1] - \mathbb{E}[Y(0)|D=0]$$

である。まず、$\mathbb{E}[Y(0)|D=1]$ は、

```
e1 <- d2 |>
  filter(D == 1) |>
  pull(h_hidden) |>
  mean() |>
  print()
```

```
[1] 2.59748
```

である。$\mathbb{E}[Y(0)|D=0]$ は、

```
e0 <- d2 |>
  filter(D == 0) |>
  pull(h_hidden) |>
  mean() |>
  print()
```

```
[1] 3.316415
```

である。よってこの場合のセレクションバイアスは、

```
(sb <- e1 - e0)
```

```
[1] -0.7189353
```

である。セルフセレクションの影響で、負のセレクションバイアスが生じていることがわかる。ATE とセレクションバイアスを足した値、

```
beta + sb
```

```
[1] -0.1189353
```

は、単純比較による効果の推定値、

```
diff(d2_D$health)
```

```
[1] -0.1189353
```

に一致する。単純比較による効果の推定を回帰分析で行ってみる。

```
lm(Y ~ D, data = d2) |>
  broom::tidy()
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic      p.value
  <chr>          <dbl>     <dbl>     <dbl>        <dbl>
1 (Intercept)    3.32     0.0144    230.    0
2 D             -0.119    0.0219     -5.44 0.0000000551
```

(当たり前だが)上と同じ推定値が得られた。また、その効果は有意水準 0.001 で統計的に有意である。**「統計的に有意」な結果を見つけても、それ自体は因果効果があるかどうかを示していない**ことがわかる。

以上のシミュレーションを、繰り返し実行できるように関数にまとめる。返り値は、単純比較によって得られる「バイアスを含んだ因果効果」とする。

```
sim_hospital <- function(beta  = 0.6,
                         N     = 1e4,
                         mu    = c(0.75, 0.6, 0.4, 0.3, 0.2),
                         sigma = rep(0.1, 5)) {
  # sigma は健康状態ごとに変えてもいいことにする
  if (length(sigma == 1)) sigma <- rep(sigma, 5)

  h_hidden <- sample(1 : 5, size = N, replace = TRUE,
                     prob = c(1, 2, 3, 2, 1))
  prob <- rnorm(N, mean = mu[h_hidden], sd = sigma[h_hidden])
  prob <- case_when(
    prob > 1 ~ 1,
    prob < 0 ~ 0,
    TRUE ~ prob
  )
  D <- rbinom(N, size = 1, prob = prob)
  Y <- h_hidden + beta * D
  fit <- lm(Y ~ D)
  return(coef(fit)[2])
}
```

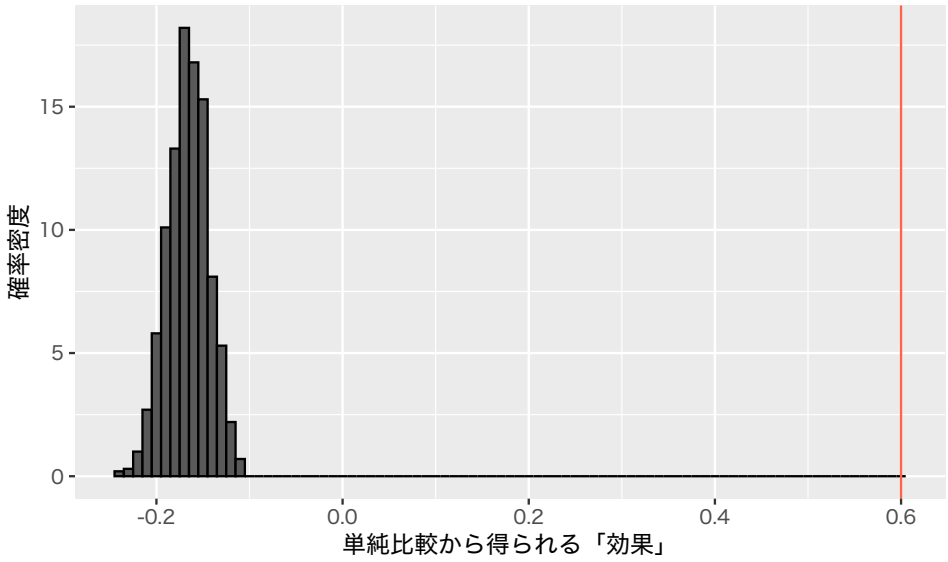
`beta = 0.6` でこの関数を一度だけ実行してみる。

```
sim_hospital(beta = 0.6)
```

```
          D
-0.1705834
```

シミュレーションを 1,000 回繰り返してみる。

```
s2_1e3 <- replicate(1e3, sim_hospital())
p_s2 <- tibble(beta = s2_1e3) |>
  ggplot(aes(x = beta, y = after_stat(density))) +
  geom_histogram(color = 'black', binwidth = 0.01) +
  geom_vline(xintercept = 0.6, color = 'tomato') +
  labs(x = '単純比較から得られる「効果」',
       y = '確率密度')
plot(p_s2)
```



効果を大幅に過小推定している。