

### 3. ランダム化比較試験(RCT)

honocat

2026-01-02

#### ランダム化のシミュレーション

##### 母集団の設定

ある処置  $D$  の平均処置効果が正であると仮定して、母集団のデータを作る。共変量として二値の性別と、年齢 (20 歳から 65 歳) を考える。

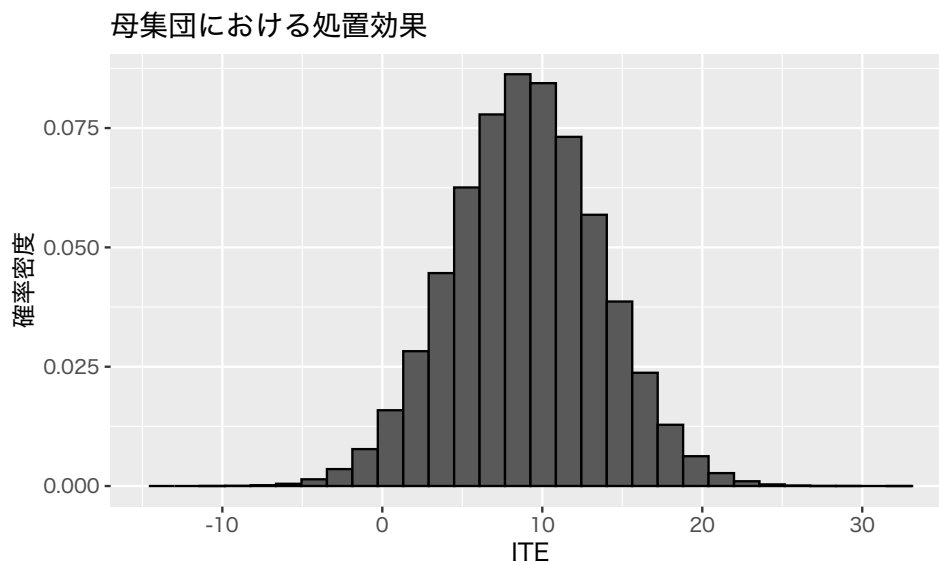
```
set.seed(2026-1-2)
N <- 1e6
y1_m <- rnorm(N, mean = 10, sd = 4)
y1_f <- rnorm(N, mean = 12, sd = 4)
y0_m <- rnorm(N, mean = 0, sd = 2)
y0_f <- rnorm(N, mean = 4, sd = 2)
age <- sample(20 : 65, size = N, replace = TRUE)
df_pop <- tibble(id = 1 : N,
                  age = age) |>
  mutate(male = rbinom(n(), size = 1, prob = 0.5),
         y1 = ifelse(male, y1_m, y1_f) + 0.05 * (age - mean(age)),
         y0 = ifelse(male, y0_m, y0_f) + 0.05 * (age - mean(age)),
         ITE = y1 - y0)
summary(df_pop)
```

	id	age	male	y1
Min.	: 1	Min. :20.00	Min. :0.0000	Min. : -9.577
1st Qu.:	250001	1st Qu.:31.00	1st Qu.:0.0000	1st Qu.: 8.184
Median :	500000	Median :43.00	Median :1.0000	Median :11.002
Mean :	500000	Mean :42.49	Mean :0.5005	Mean :11.002
3rd Qu.:	750000	3rd Qu.:54.00	3rd Qu.:1.0000	3rd Qu.:13.821
Max.	:1000000	Max. :65.00	Max. :1.0000	Max. :32.650
	y0	ITE		

Min.	:-9.7347	Min.	:-13.801
1st Qu.:	-0.1335	1st Qu.:	5.911
Median	: 1.9989	Median	: 8.998
Mean	: 1.9996	Mean	: 9.002
3rd Qu.:	4.1334	3rd Qu.:	12.090
Max.	:13.4596	Max.	: 32.321

ITE（個体処置効果）の分布を確認する。

```
pop_ite <- ggplot(df_pop, aes(x = ITE,
                              y = after_stat(density))) +
  geom_histogram(color = 'black') +
  labs(y = ' 確率密度',
       title = ' 母集団における処置効果')
plot(pop_ite)
```



母集団における平均処置効果を計算しておく。

```
pop_ATE <- mean(df_pop$ITE) |>
print()
```

```
[1] 9.00203
```

```
pop_ATEm <- df_pop |>
  filter(male == 1) |>
  with(mean(ITE)) |>
```

```
print()
```

```
[1] 10.00583
```

```
pop_ATEf <- df_pop |>
  filter(male == 0) |>
  with(mean(ITE)) |>
  print()
```

```
[1] 7.996259
```

以下では、この母集団に対して異なる方法で RCT を実施した場合にどのような結果が得られるか、シミュレーションで確かめる。

### ベルヌーイ実験

ベルヌーイ実験(Bernoulli experiments)を R で実行しよう。まず、シミュレーションを繰り返す行うために、実験を実行する関数を作る。標本サイズ(1 回の実験における被験者の数)を `sample_size` で、各ベルヌーイ試行で処置に割り付ける確率を `prob` で指定できるようにする。

```
bern_exp <- function(population,
                      sample_size = 100,
                      prob = 0.5,
                      info = FALSE) {
  df <- slice_sample(.data = population, n = sample_size)
  df <- df |>
    mutate(D = rbinom(sample_size, size = 1, prob = prob))
  ATE <- with(df, mean(y1[D == 1]) - mean(y0[D == 0]))
  if (info) cat(' 処置群の個体数:', sum(df$D),
               '; 統制群の個体数:', sample_size - sum(df$D), '\n')
  return(ATE)
}
```

この関数を使ってベルヌーイ実験を 1 度実行してみる。 `sample_size = 100`, `prob = 0.5` とする。

```
bern_exp(population = df_pop, info = TRUE)
```

```
処置群の個体数: 47 ; 統制群の個体数: 53
```

```
[1] 8.99745
```

もう一度。

```
bern_exp(population = df_pop, info = TRUE)
```

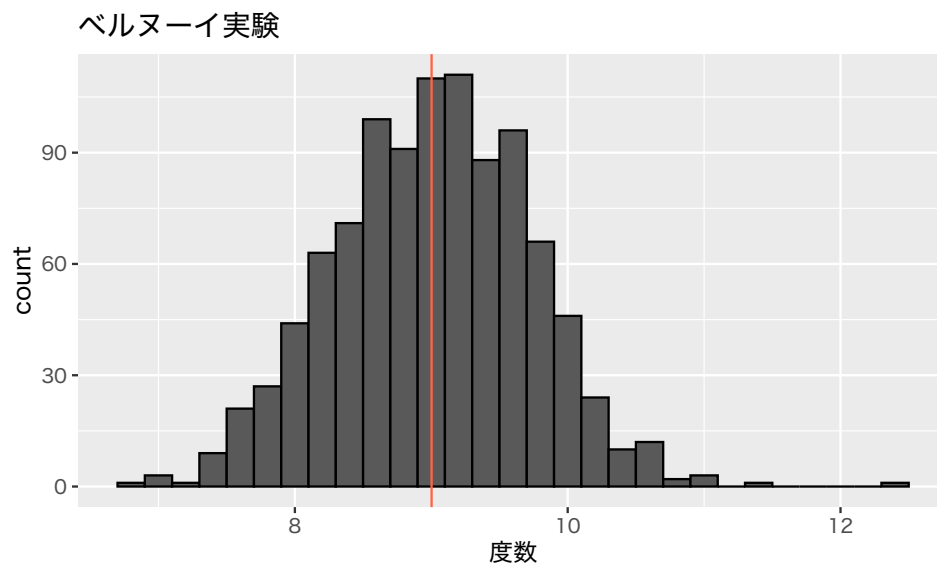
処置群の個体数: 44 ; 統制群の個体数: 56

```
[1] 7.91662
```

当然ながら、実験結果は毎回変わる。

実験を 1,000 回繰り返し、結果を図示する。母集団における平家員処置効果を赤い縦線で示す。標本サイズは 100 である。

```
bern100 <- replicate(1e3, bern_exp(df_pop))
hist_bern100 <- tibble(ATE = bern100) |>
  ggplot(aes(x = ATE)) +
  geom_histogram(binwidth = 0.2, color = 'black') +
  geom_vline(xintercept = pop_ATE, color = 'tomato') +
  labs(x = '度数', title = 'ベルヌーイ実験')
plot(hist_bern100)
```



一つ一つの実験は必ずしも ATE を正確に推定しているわけではないが、平均すれば正しい効果を推定できていることがわかる。「正しい」ATE が約 9 であるのに対し、RCT による推定の平均値は 9.01 である。

処置に割り付ける確率は 0.5 でなくても良い。

```
bern_exp(df_pop, prob = 0.8, info = TRUE)
```

処置群の個体数: 81 ; 統制群の個体数: 19

```
[1] 9.434529
```

```
bern_exp(df_pop, prob = 0.2, info = TRUE)
```

処置群の個体数: 20 ; 統制群の個体数: 80

```
[1] 9.434183
```

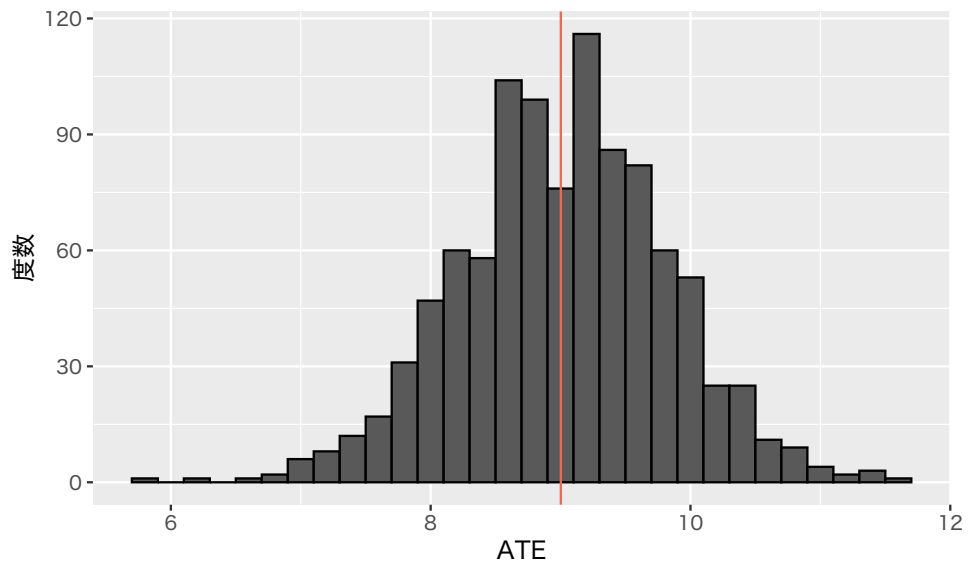
```
bern_exp(df_pop, prob = 0.1, info = TRUE)
```

処置群の個体数: 14 ; 統制群の個体数: 86

```
[1] 8.86861
```

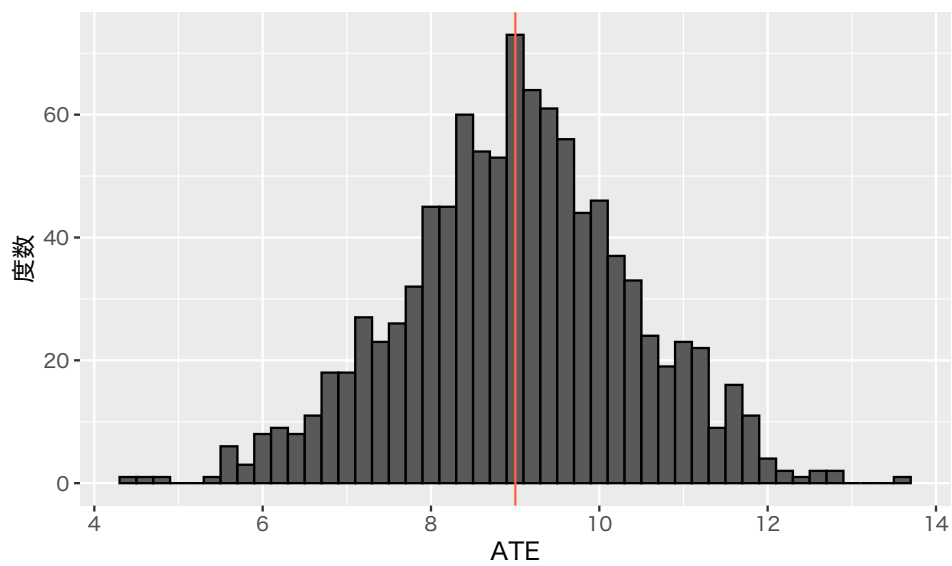
処置に割り付ける確率を 0.8 として、実験を 1,000 回繰り返し、その結果を可視化する。

```
bern_p80 <- replicate(1e3, bern_exp(df_pop, prob = 0.8))
hist_bern_p80 <- tibble(ATE = bern_p80) |>
  ggplot(aes(x = ATE)) +
  geom_histogram(binwidth = 0.2, color = 'black') +
  geom_vline(xintercept = pop_ATE, color = 'tomato') +
  labs(y = '度数')
plot(hist_bern_p80)
```



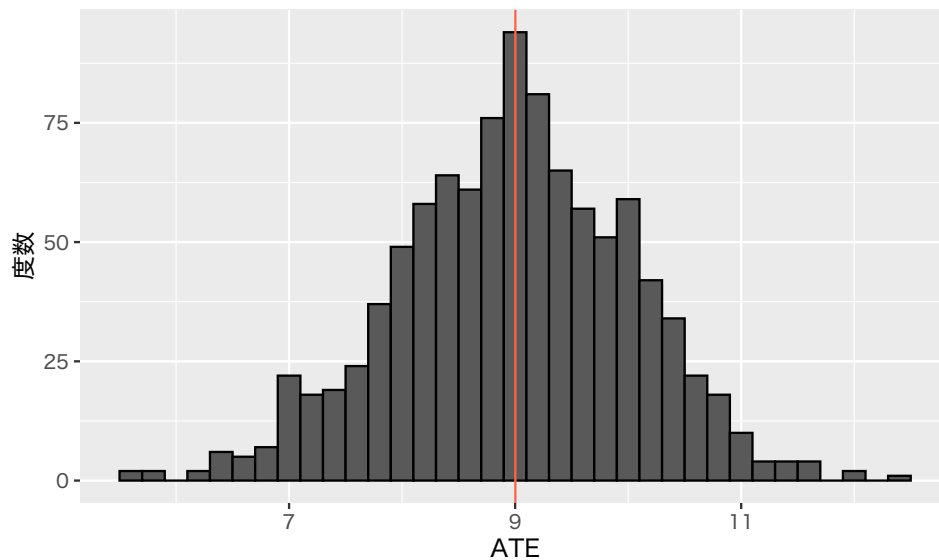
標本サイズを変えてみると、

```
bern30 <- replicate(1e3, bern_exp(df_pop, sample_size = 30))
hist_bern30 <- tibble(ATE = bern30) |>
  ggplot(aes(x = ATE)) +
  geom_histogram(binwidth = 0.2, color = 'black') +
  geom_vline(xintercept = pop_ATE, color = 'tomato') +
  labs(y = '度数')
plot(hist_bern30)
```



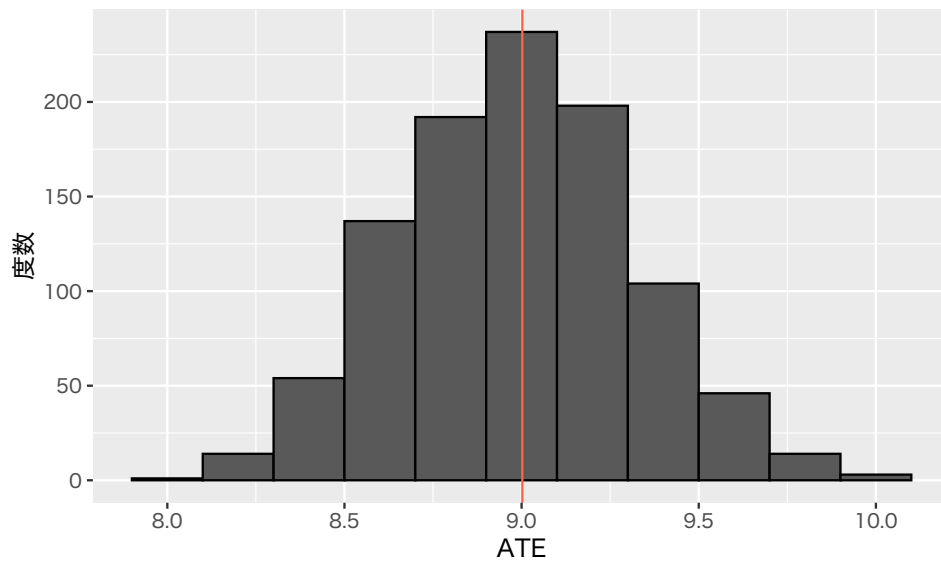
標本サイズが 50 の場合：

```
bern50 <- replicate(1e3, bern_exp(df_pop, sample_size = 50))
hist_bern50 <- tibble(ATE = bern50) |>
  ggplot(aes(x = ATE)) +
  geom_histogram(binwidth = 0.2, color = 'black') +
  geom_vline(xintercept = pop_ATE, color = 'tomato') +
  labs(y = '度数')
plot(hist_bern50)
```



標本サイズが 500 の場合：

```
bern500 <- replicate(1e3, bern_exp(df_pop, sample_size = 500))
hist_bern500 <- tibble(ATE = bern500) |>
  ggplot(aes(x = ATE)) +
  geom_histogram(binwidth = 0.2, color = 'black') +
  geom_vline(xintercept = pop_ATE, color = 'tomato') +
  labs(y = '度数')
plot(hist_bern500)
```



標本サイズが小さくても、平均的には正しい因果効果を推定できている。しかし、標本サイズが小さいと、推定のばらつき(標準誤差)が大きくなる。

```
sd(bern30)
```

```
[1] 1.371831
```

```
sd(bern50)
```

```
[1] 1.040279
```

```
sd(bern500)
```

```
[1] 0.3296721
```

標準誤差が大きいと、過小推定や過大推定が多くなることがわかる。実験の回数が少ない場合、標本サイズを十分な大きさにしないと、過小推定や過大推定をしてしまうかもしれない。

### 完全ランダム化実験

次に、完全ランダム化実験(completely random experiments)を行う。`sample_size`と処置に割り付ける個体の数`n_treated`を選べるようにする。ただし、`sample_size > n_treated`である。

```
cr_exp <- function(population,
                    sample_size = 100,
                    n_treated    = ceiling(sample_size / 2),
                    info         = FALSE) {
```

```

if (sample_size <= n_treated | n_treated <= 0)
  stop('n_treated must be a positive integer
       smaller than sample_size.')
df <- slice_sample(.data = population, n = sample_size) |>
  mutate(D = sample(rep(c(1, 0),
                        c(n_treated, sample_size - n_treated)),
                size = sample_size,
                replace = FALSE))
ATE <- with(df, mean(y1[D == 1]) - mean(y0[D == 0]))
if (info) cat(' 処置群の個体数:', n_treated,
              '; 統制群の個体数:', sample_size - n_treated,
              '\n')
return(ATE)
}

```

sample\_size = 100, n\_treated = 50 にして、一度実験を行ってみる。

```
cr_exp(df_pop, info = TRUE)
```

処置群の個体数: 50 ; 統制群の個体数: 50

```
[1] 9.415287
```

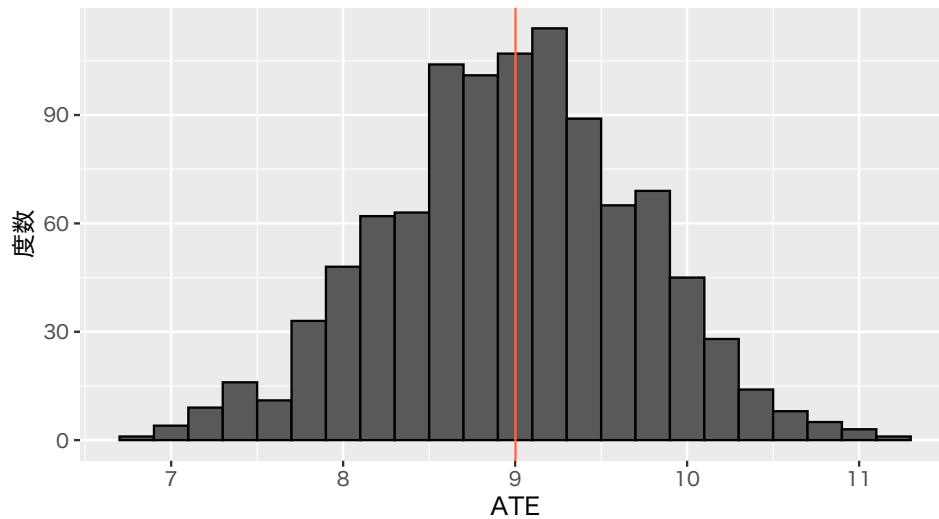
この実験を 1,000 回繰り返し、結果を可視化する。

```

cr100 <- replicate(1e3, cr_exp(df_pop))
hist_cr100 <- tibble(ATE = cr100) |>
  ggplot(aes(x = ATE)) +
  geom_histogram(binwidth = 0.2, color = 'black') +
  geom_vline(xintercept = pop_ATE, color = 'tomato') +
  labs(y = '度数', title = '完全ランダム化実験')
plot(hist_cr100)

```

### 完全ランダム化実験



やはり、平均的にはうまく推定できているようだ。ATE の推定値の平均値は、8.98 である。

完全ランダム化実験で処置に割り付ける数は、標本サイズの半分でなくても良い。

```
cr_exp(df_pop, sample_size = 100, n_treated = 70, info = TRUE)
```

処置群の個体数: 70 ; 統制群の個体数: 30

```
[1] 8.243557
```

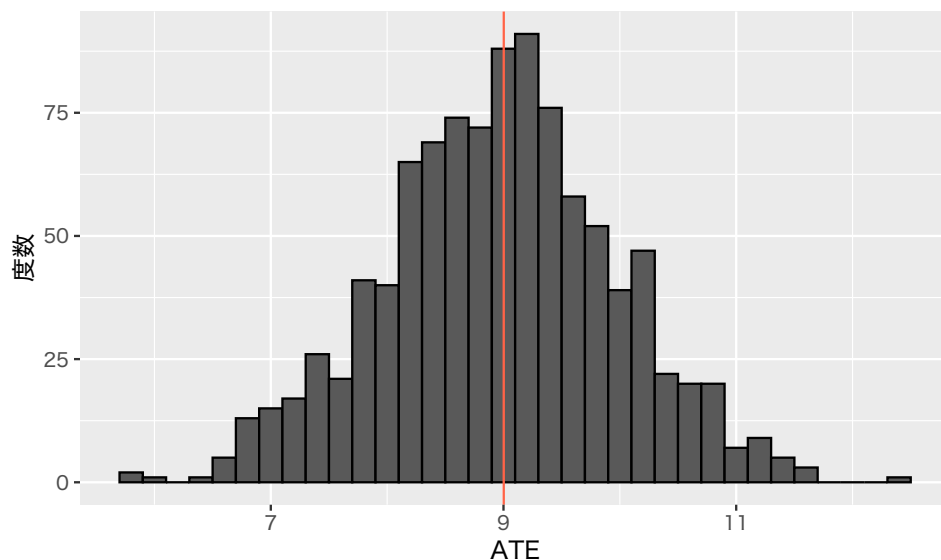
```
cr_exp(df_pop, sample_size = 100, n_treated = 25, info = TRUE)
```

処置群の個体数: 25 ; 統制群の個体数: 75

```
[1] 9.62966
```

サンプルサイズ 100 のうち、処置に 20 個体を割り付ける実験を 1,000 回繰り返し、結果を可視化する。

```
cr_prop20 <- replicate(1e3, cr_exp(df_pop,
                                   sample_size = 100,
                                   n_treated = 20))
hist_cr_prop20 <- tibble(ATE = cr_prop20) |>
  ggplot(aes(x = ATE)) +
  geom_histogram(binwidth = 0.2, color = 'black') +
  geom_vline(xintercept = pop_ATE, color = 'tomato') +
  labs(y = '度数')
plot(hist_cr_prop20)
```



標本サイズ(被験者数)の 20% のみを処置に割り付けても、平均的には正しい ATE を推定できる。つまり、無作為化比較試験でサンプルを 2 群に分けると、サンプルを半々に分ける必要はないことがわかる(必要がないだけで半々にしたほうが効率はやい)。

### ブロック別ランダム化実験

続いて、ブロック別ランダム化実験(stratified randomized experiments)を実施してみる。ここでは、性別(男性ダミー変数)によってブロックを作る。処置に割り付ける数は、各群における処置の割合で指定できるようにする。例えば、男性ブロックでは 7 割を処置、女性ブロックでは 4 割を処置するなら、`prop = c(0.7, 0.4)` と指定できるようにする。

```
strat_exp <- function(population,
                      sample_size = 100,
                      prop         = c(0.5, 0.5),
                      info         = FALSE) {
  if (min(prop) <= 0 | max(prop) >= 1) stop('prop must be in (0, 1).')

  df <- slice_sample(.data = population, n = sample_size)
  Male <- df |> filter(male == 1)
  Female <- df |> filter(male == 0)
  n_treated <- ceiling(c(nrow(Male), nrow(Female)) * prop)
  ATE_male <- cr_exp(Male,
                    sample_size = nrow(Male),
                    n_treated    = n_treated[1])
  ATE_female <- cr_exp(Female,
                     sample_size = nrow(Female),
```

```

        n_treated    = n_treated[2])
male_prop <- mean(df$male)
ATE <- male_prop * ATE_male + (1 - male_prop) * ATE_female
if (info) {
  cat(' 男性ブロック 処置群の個体数:',
      n_treated[1],
      '; 統制群の個体数:',
      nrow(Male) - n_treated[1],
      '\n')
  cat(' 女性ブロック 処置群の個体数:',
      n_treated[2],
      '; 統制群の個体数:',
      nrow(Female) - n_treated[2],
      '\n')
}
return(ATE)
}

```

sample\_size = 100、prop = c(0.5, 0.5) として、1 回実験を行う。

```
strat_exp(df_pop, info = TRUE)
```

男性ブロック 処置群の個体数: 22 ; 統制群の個体数: 22

女性ブロック 処置群の個体数: 28 ; 統制群の個体数: 28

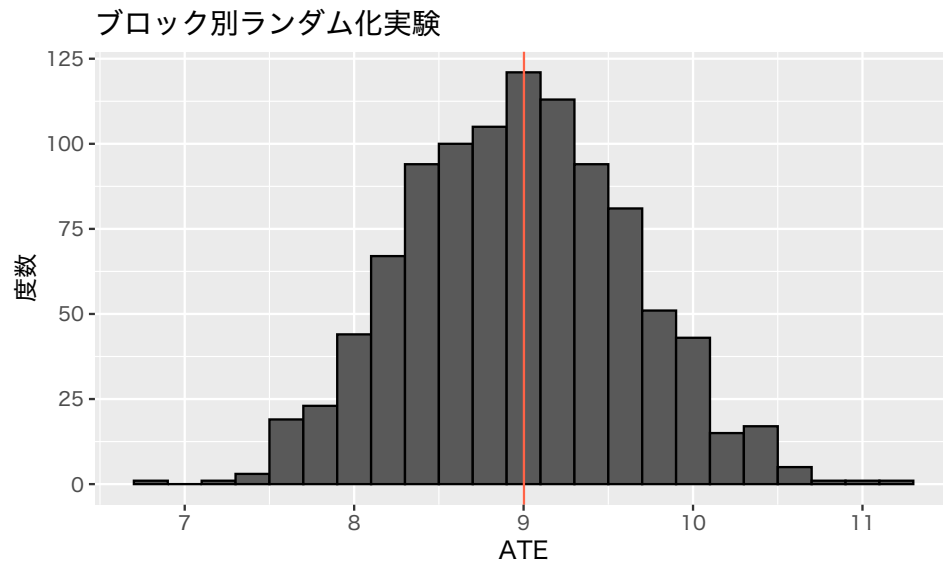
[1] 8.745238

この実験を 1,000 回繰り返し、結果を可視化する。

```

strat100 <- replicate(1e3, strat_exp(df_pop))
hist_strat100 <- tibble(ATE = strat100) |>
  ggplot(aes(x = ATE)) +
  geom_histogram(binwidth = 0.2, color = 'black') +
  geom_vline(xintercept = pop_ATE, color = 'tomato') +
  labs(y = '度数', title = 'ブロック別ランダム化実験')
plot(hist_strat100)

```



やはり、平均的にはうまく推定できているようだ。ATE の推定値の平均値は、8.97 である。ブロックごとに処置に割り付ける割合が違って、平均的には正しい推定ができる。

#### ペア毎のランダム化実験

4 番目の方法として、ペア毎のランダム化試験(対ランダム化実験、paired randomized experiments)を実施する。サンプルの中で、性別が同じで、年齢が近い者同士をペアにして、ペア毎に一方を処置に、他方を統制に割り付ける。ペアを組む都合上、標本サイズは偶数に制限し、サンプル内の男女の数がそれぞれ偶数になるように調整する。

```
paired_exp <- function(population, sample_size = 100) {
  if (sample_size %% 2 != 0) stop('sample_size must be an even number.')
  n_male <- sample_size / 2
  if (n_male %% 2 != 0) n_male <- n_male + sample(c(-1, 1), size = 1)
  n_female <- sample_size - n_male

  Male <- population |>
    filter(male == 1) |>
    slice_sample(n = n_male) |>
    arrange(age) |>
    mutate(D = as.vector(replicate(n() / 2, sample(0 : 1, size = 2))))
  Female <- population |>
    filter(male == 0) |>
    slice_sample(n = n_female) |>
    arrange(age) |>
    mutate(D = as.vector(replicate(n() / 2, sample(0 : 1, size = 2))))
}
```

```
df <- bind_rows(Male, Female)
ATE <- with(df, mean(y1[D == 1]) - mean(y0[D == 0]))
return(ATE)
}
```

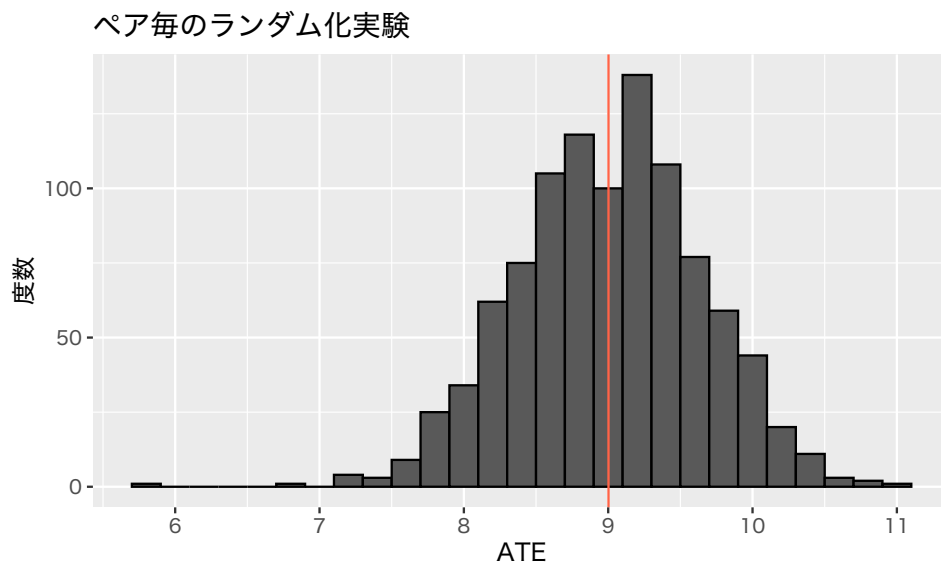
一度使ってみる。

```
paired_exp(df_pop)
```

```
[1] 9.238002
```

この実験を 1,000 回繰り返し、結果を可視化する。

```
pair100 <- replicate(1e3, paired_exp(df_pop))
hist_pair100 <- tibble(ATE = pair100) |>
  ggplot(aes(x = ATE)) +
  geom_histogram(binwidth = 0.2, color = 'black') +
  geom_vline(xintercept = pop_ATE, color = 'tomato') +
  labs(y = '度数', title = 'ペア毎のランダム化実験')
plot(hist_pair100)
```



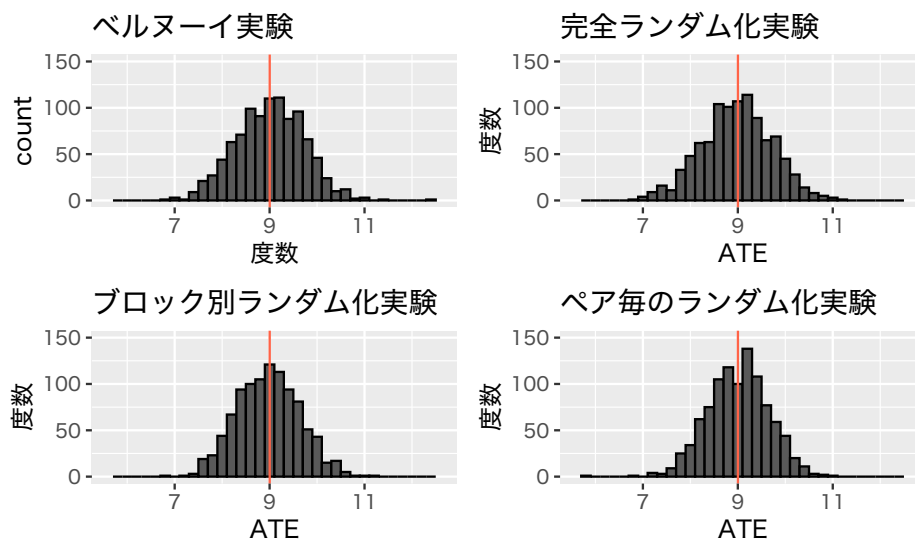
やはり、平均的にはうまく推定できているようだ。ATE の推定値の平均値は、9 である。各ペアで処置に割り付ける確率は、0.5 でなくてもいい。

#### 4つのRCTを比較する

最後に、それぞれのRCTの推定精度を比較しよう。どれも標本サイズは100で、処置の確率・割合・数が全体の半分(程度)になるようにしたもの比べる。

1,000回の実験をシミュレートしたヒストグラムを再掲する。

```
MIN <- min(c(bern100, cr100, strat100, pair100)) - 0.2
MAX <- max(c(bern100, cr100, strat100, pair100)) + 0.2
plot(
  ((hist_bern100 + xlim(MIN, MAX) + ylim(0, 150)) |
   (hist_cr100 + xlim(MIN, MAX) + ylim(0, 150))) /
  ((hist_strat100 + xlim(MIN, MAX) + ylim(0, 150)) |
   (hist_pair100 + xlim(MIN, MAX) + ylim(0, 150)))
)
```



標準誤差を比較する。

```
sd(bern100)
```

```
[1] 0.7141867
```

```
sd(cr100)
```

```
[1] 0.7340488
```

```
sd(strat100)
```

```
[1] 0.6532495
```

```
sd(pair100)
```

```
[1] 0.6395695
```

このように、処置のパタンの数が最も少ない(かつブロック [ペア] がより同質的な)ペア毎のランダム化実験の標準誤差が最も小さく、推定精度が高いことがわかる。