

05. 傾向スコア

honocat

2026-01-06

傾向スコアを用いた因果推論の手順

傾向スコアを使った因果効果の推定は、以下の手順で行う。

1. 共変量の選定
2. 傾向スコアの推定
3. 傾向スコアを用いたバランス調整
 - 重み付け
 - 層別
 - マッチング
4. 共変量のバランスチェック
5. 処置効果(因果効果)の推定
6. 感度分析(Carnegie[2016]を参照。 **treatSens** パッケージが用意されている)

安井(2020)でも分析されている LaLonde(1986) の実験データを使い、傾向スコアを使った分析手順を説明する。

この実験の関心は、「職業訓練(カウンセリングと短期的な就労経験)」が、その後の収入に影響を与えるかどうかである。データセットのなかで、処置変数である職業訓練は `treat` で、結果変数である 1978 年の収入は `re78` で表される。

```
base_url <- 'https://users.nber.org/~rdehejia/data/'
targets <- c('nsw_dw.dta', 'cps_controls.dta', 'cps_controls3.dta')
download.file(url = str_c(base_url, targets),
              destfile = file.path('data', targets))
```

手に入れたデータを読み込む。

```
d_cps1 <- read_dta('data/cps_controls.dta')
d_cps3 <- read_dta('data/cps_controls3.dta')
d_nsw <- read_dta('data/nsw_dw.dta')
```

実験データである `d_nsw` で回帰分析を試みる。

```
covariates0 <- names(d_nsw)[2 : (ncol(d_nsw) - 1)] |>
  str_c(collapse = ' + ')
```

これで、共変量が + で繋がれた文字列ができた。

```
covariates0
```

```
[1] "treat + age + education + black + hispanic + married + nodegree + re74 + re75"
```

このままでは回帰分析の式として完成していないので、さらに次のようにする。

```
rct_formula <- 're78 ~ ' |>
  str_c(covariates0) |>
  formula()
```

これで、`lm()` で利用できる式(`formula`)ができた。

```
rct_formula
```

```
re78 ~ treat + age + education + black + hispanic + married +
  nodegree + re74 + re75
```

線形モデルを仮定した回帰式を推定する。

```
ate_rct <- lm(rct_formula, data = d_nsw) |>
  tidy() |>
  filter(term == 'treat') |>
  pull(estimate) |>
  print()
```

```
[1] 1676.343
```

実験で推定された処置効果は\$1676 であることがわかる。つまり、職業訓練を受けることで、収入は\$1676 増えると推定される。

次に、実験データである `d_nsw` の統制群を削除し、調査データである `d_cps1` を代わりの統制群として扱うデータセットを作る。これにより、セレクションバイアスを含むデータができる。

```
d_cps1_nsw <- d_nsw |>
  filter(treat == 1) |>
  rbind(d_cps1)
```

同様に、実験データである `d_nsw` の統制群を削除し、調査データである `d_cps3` を代わりの統制群として扱うデータセットを作る。

```
d_cps3_nsw <- d_nsw |>
  filter(treat == 1) |>
  rbind(d_cps3)
```

以下では、上の `d_cps1_nsw` を使って、傾向スコアを使った分析の手順を説明する。

共変量の選定

まず、データ `d_cps1_nsw` の中身を確認する。

```
summary(d_cps1_nsw)
```

data_id	treat	age	education
Length:16177	Min. :0.00000	Min. :16.00	Min. : 0.00
Class :character	1st Qu.:0.00000	1st Qu.:24.00	1st Qu.:11.00
Mode :character	Median :0.00000	Median :31.00	Median :12.00
	Mean :0.01144	Mean :33.14	Mean :12.01
	3rd Qu.:0.00000	3rd Qu.:42.00	3rd Qu.:13.00
	Max. :1.00000	Max. :55.00	Max. :18.00
black	hispanic	married	nodegree
Min. :0.00000	Min. :0.00000	Min. :0.0000	Min. :0.0000
1st Qu.:0.00000	1st Qu.:0.00000	1st Qu.:0.0000	1st Qu.:0.0000
Median :0.00000	Median :0.00000	Median :1.0000	Median :0.0000
Mean :0.08234	Mean :0.07189	Mean :0.7058	Mean :0.3006
3rd Qu.:0.00000	3rd Qu.:0.00000	3rd Qu.:1.0000	3rd Qu.:1.0000
Max. :1.00000	Max. :1.00000	Max. :1.0000	Max. :1.0000
re74	re75	re78	
Min. : 0	Min. : 0	Min. : 0	
1st Qu.: 4075	1st Qu.: 4103	1st Qu.: 5493	
Median :14892	Median :14374	Median :16240	
Mean :13880	Mean :13512	Mean :14749	
3rd Qu.:23492	3rd Qu.:22830	3rd Qu.:25565	
Max. :35040	Max. :25244	Max. :60308	

回帰分析の準備として、処置変数である `treat`（職業訓練の有無）と、結果変数である `re78`（1978 年の収入）以外の変数を中心化する。また、`re74` と `re75` の二乗項を標準化した変数も作る。

```
myd <- d_cps1_nsw |>
  mutate(age_c = age - mean(age),
         education_c = education - mean(education),
         black_c = black - mean(black),
         hispanic_c = hispanic - mean(hispanic),
```

```

married_c    = married    - mean(married),
nodegree_c   = nodegree   - mean(nodegree),
re74_c       = re74       - mean(re74),
re75_c       = re75       - mean(re75),
re74_sq_c    = (re74_c ^ 2 - mean(re74_c ^ 2)) / sd(re74_c ^ 2),
re75_sq_c    = (re75_c ^ 2 - mean(re75_c ^ 2)) / sd(re75_c ^ 2)) |>
select(re78, treat, ends_with('_c'))

```

データを確認する。

```
summary(myd)
```

re78		treat	age_c		education_c		
Min.	: 0	Min.	:0.00000	Min.	: -17.141	Min.	: -12.008283
1st Qu.:	5493	1st Qu.:	0.00000	1st Qu.:	-9.141	1st Qu.:	-1.008283
Median	:16240	Median	:0.00000	Median	: -2.141	Median	: -0.008283
Mean	:14749	Mean	:0.01144	Mean	: 0.000	Mean	: 0.000000
3rd Qu.:	25565	3rd Qu.:	0.00000	3rd Qu.:	8.859	3rd Qu.:	0.991717
Max.	:60308	Max.	:1.00000	Max.	: 21.859	Max.	: 5.991717
black_c		hispanic_c		married_c		nodegree_c	
Min.	: -0.08234	Min.	: -0.07189	Min.	: -0.7058	Min.	: -0.3006
1st Qu.:	-0.08234	1st Qu.:	-0.07189	1st Qu.:	-0.7058	1st Qu.:	-0.3006
Median	: -0.08234	Median	: -0.07189	Median	: 0.2942	Median	: -0.3006
Mean	: 0.00000	Mean	: 0.00000	Mean	: 0.0000	Mean	: 0.0000
3rd Qu.:	-0.08234	3rd Qu.:	-0.07189	3rd Qu.:	0.2942	3rd Qu.:	0.6994
Max.	: 0.91766	Max.	: 0.92811	Max.	: 0.2942	Max.	: 0.6994
re74_c		re75_c		re74_sq_c		re75_sq_c	
Min.	: -13880	Min.	: -13512.2	Min.	: -1.36753	Min.	: -1.3609
1st Qu.:	-9805	1st Qu.:	-9408.8	1st Qu.:	-1.02527	1st Qu.:	-1.0265
Median	: 1012	Median	: 862.3	Median	: 0.02418	Median	: 0.0134
Mean	: 0	Mean	: 0.0	Mean	: 0.00000	Mean	: 0.0000
3rd Qu.:	9611	3rd Qu.:	9318.0	3rd Qu.:	0.75710	3rd Qu.:	0.7986
Max.	: 21160	Max.	: 11731.3	Max.	: 5.25847	Max.	: 1.5040

この例では、処置変数と結果変数以外の変数をすべて使って傾向スコアを推定するが、自分の RQ に答えるためにどの変数を傾向スコアの推定に使うかは、問題に応じて考える必要がある。

ここですべての変数を使う理由は、それぞれの変数が少なくとも結果変数には影響する（処置と結果の両者に影響するか、結果のみに影響する）と考えられるからである。

共変量あるいは傾向スコアによる調整を行わずに、処置群と統制群を単純比較してみよう。

```
reg_simple <- lm(re78 ~ treat, data = myd)
tidy(reg_simple)
```

```
# A tibble: 2 x 5
  term          estimate std.error statistic  p.value
<chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  14847.         76.1      195.    0
2 treat        -8498.        712.     -11.9 1.07e-32
```

処置(職業訓練)は収入を\$8498 減らすという誤った効果が推定されている。この回帰分析によれば、この効果は有意水準 0.01 で統計的に有意である。

傾向スコアの推定

次に、傾向スコアを推定する。ここでは、ロジスティック回帰で推定することにする。R でロジスティック回帰(あるいはその他の一般化線形モデル [GLM])を実行するには、`glm()` を使う。

ロジスティック回帰は、結果変数が $Y \in \{0, 1\}$ 、説明変数が X だとすると、

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

$$\text{logit}(\theta_i) = \alpha_0 + \alpha_1 X_{1,i} + \dots \alpha_k X_{k,i}$$

という統計モデルである。つまり、結果変数を生成する確率分布が $\text{Bernoulli}(\theta_i) = \text{Binomial}(1, \theta_i)$ であり、リンク関数がロジット(logit)関数であると仮定している。よって、`glm()` の引数で、`family = binomial(link = 'logit')` を使い、`glm(Y ~ X1 + X2 + X3, family = binomial(link = 'logit'))` のようにして使う。

```
covariates <- names(myd)[3 : ncol(myd)]
ps_formula <- covariates |>
  str_c(collapse = ' + ')
ps_formula <- str_c('treat ~ ', ps_formula) |>
  formula()
```

これで、`glm()` で利用できる式(formula)ができた。

```
ps_formula

treat ~ age_c + education_c + black_c + hispanic_c + married_c +
  nodegree_c + re74_c + re75_c + re74_sq_c + re75_sq_c
```

これを使ってロジスティック回帰式を推定する。

```
fit_logistic <- glm(ps_formula,
  data = myd,
```

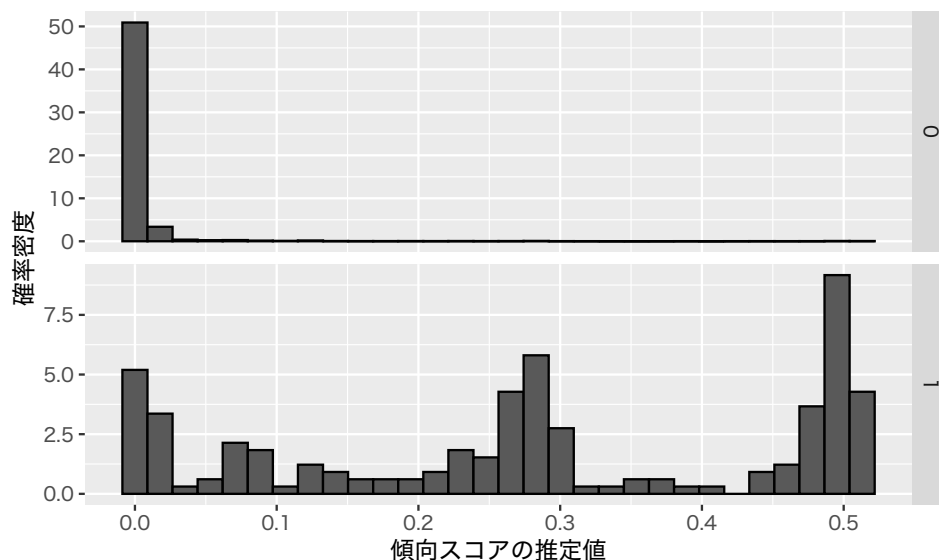
```
family = binomial(link = 'logit'))
```

これで推定ができた。ここでは傾向スコアの推定値にのみ興味がある。傾向スコアすなわち θ_i の推定値は、`fitted()` で取り出せる。

```
myd$ps_logit <- fitted(fit_logistic)
```

推定された傾向スコアの分布を確認してみる。

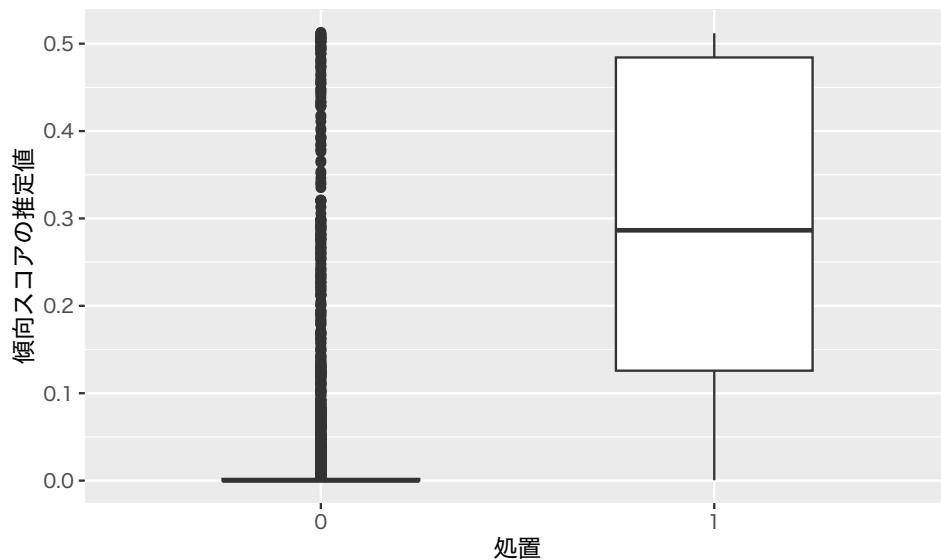
```
hist_ps_logit <- ggplot(myd,
                        aes(x = ps_logit,
                            y = after_stat(density))) +
  geom_histogram(color = 'black') +
  facet_grid(rows = vars(treat), scales = 'free_y') +
  labs(x = '傾向スコアの推定値', y = '確率密度')
plot(hist_ps_logit)
```



統制群(0; 上段)と処置群(1; 下段)で傾向スコアの分布が大きく異なることがわかる(上段と下段で縦軸のスケールが異なる [free_y を指定した] ことに注意)。

念の為、他の種類の図も作る。

```
box_ps_logit <- ggplot(myd,
                      aes(x = as.factor(treat),
                          y = ps_logit)) +
  geom_boxplot(width = 0.5) +
  labs(x = '処置', y = '傾向スコアの推定値')
plot(box_ps_logit)
```



箱ひげ図で見ても、分布が大きく異なる事がわかる(ちなみに、あまりにも分布が異なるので、バイオリン図では違いがよく見えない。また、観測数が多いので、蜂群図を作るのも大変である)。ヒストグラムでは、統制群に傾向スコアの値が 0.1 以上の個体があるかどうかがよくわからなかったが、箱ひげ図を描いてみると、分布の範囲自体は重なり合っていることがわかる。つまり、「共有(共通)サポート(common support)」はありそうだ。

このように、複数の種類の図を描き、多角的に検討することが必要である。

バランス調整、バランスチェック、因果効果の推定

ここからは、バランス調整の方法別に、(1) その方法、(2) バランスチェック、(3) 因果効果の推定の 3 つのステップをあわせて説明する。バランス調整の方法として、次の方法を説明する。

1. 重み付け
 - ATT を推定するための重み付け
 - ATE を推定するための重み付け(IPW)
2. 層別

■重み付け

傾向スコアを利用した重み付けで因果効果を推定する方法を紹介する。推定対象(estimand)によって使う重みが異なるので、自分が何を推定したいのかを明確にしておく必要がある。通常、推定の対象としたいのは、ATE (平均処置効果) または ATT (処置群における平均処置効果) であることが多いので、この 2 つについて説明する。

ATT (処置群における平均処置効果)の推定

ATT を推定するために、傾向スコア $e_i(X)$ を用いて次の重みを計算する。

$$w_i^{\text{ATT}} = D_i + (1 - D_i) \frac{e_i(X)}{1 - e_i(X)}$$

この重みを使うと、処置群($D_i = 1$)の個体の重みは $w_i^{\text{ATT}} = 1$ になる。つまり、処置群の個体には、傾向スコアに関係なく等しい重みが与えられる。

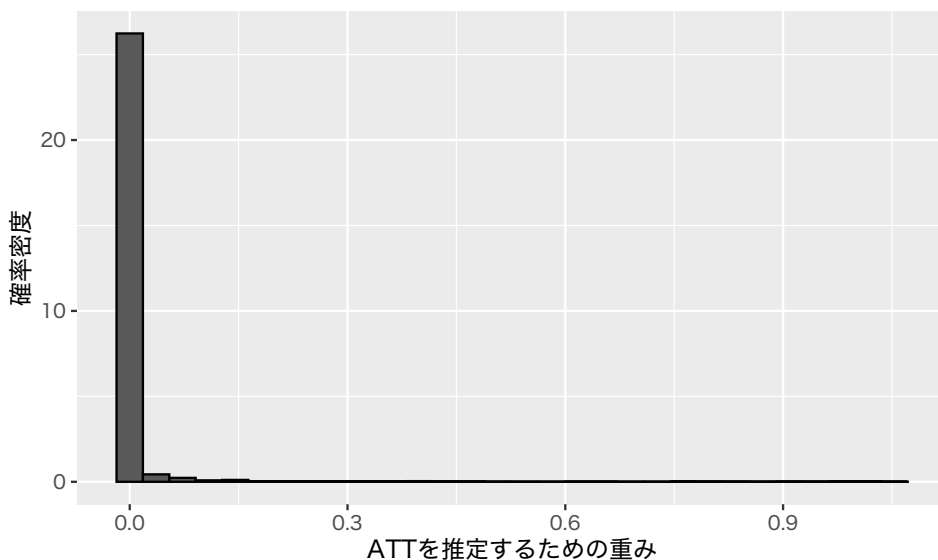
それに対し、統制群($D_i = 0$)の個体の重みは $w_i^{\text{ATT}} = \frac{e_i(X)}{1 - e_i(X)}$ になる。つまり、傾向スコアが高いほど、大きな重みが与えられる。これは、傾向スコアが高い、すなわち処置群に入る確率が高いにもかかわらず統制群に入った個体は、処置群の比較対象として重宝されるということを意味する。反対に、傾向スコアが低い個体は、処置群の比較対象としてあまり重要ではない(処置群にはそれによく似た個体が少ないのに、同種の個体が統制群にたくさんある)ので、割り引いて考えられるということである。

この重みを計算しよう。

```
myd <- myd |>
  mutate(w_att = treat + (1 - treat) * ps_logit / (1 - ps_logit))
```

統制群でこの重みがどのように分布しているか確認する。

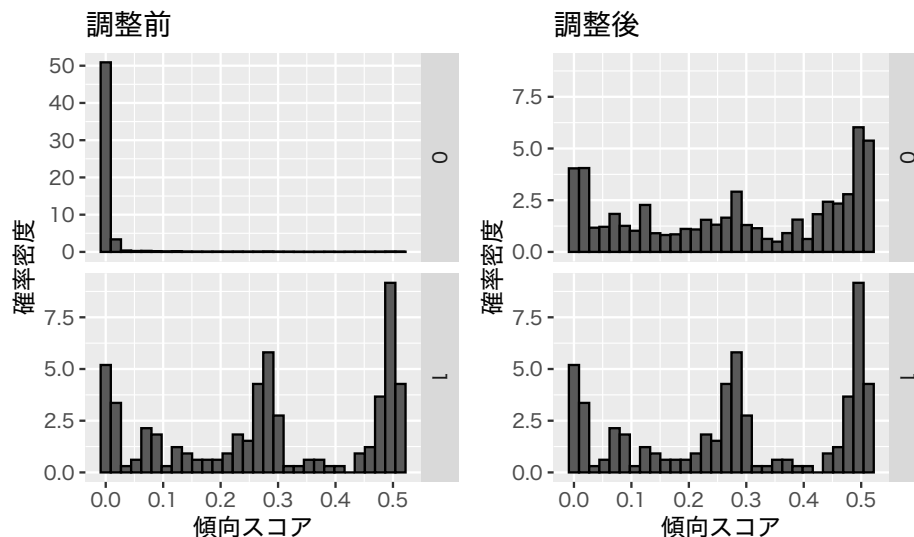
```
hist_w_att <- myd |>
  filter(treat == 0) |>
  ggplot(aes(x = w_att, y = after_stat(density))) +
  geom_histogram(color = 'black') +
  labs(x = 'ATTを推定するための重み', y = '確率密度')
plot(hist_w_att)
```



統制群には傾向スコアが小さな個体ばかり集まっていて、重みも 0 付近に集中していることがわかる。

処置群と統制群の傾向スコアの分布を、調整前と調整後で比較する。


```
att_before <- ggplot(myd, aes(x = ps_logit,
                             y = after_stat(density))) +
  geom_histogram(color = 'black') +
  facet_grid(row = vars(treat), scale = 'free_y') +
  labs(x = '傾向スコア', y = '確率密度', title = '調整前')
att_after <- ggplot(myd,
                   aes(x = ps_logit,
                       y = after_stat(density),
                       weight = w_att)) +
  geom_histogram(color = 'black') +
  facet_grid(row = vars(treat)) +
  labs(x = '傾向スコア', y = '確率密度', title = '調整後')
plot(att_before | att_after)
```

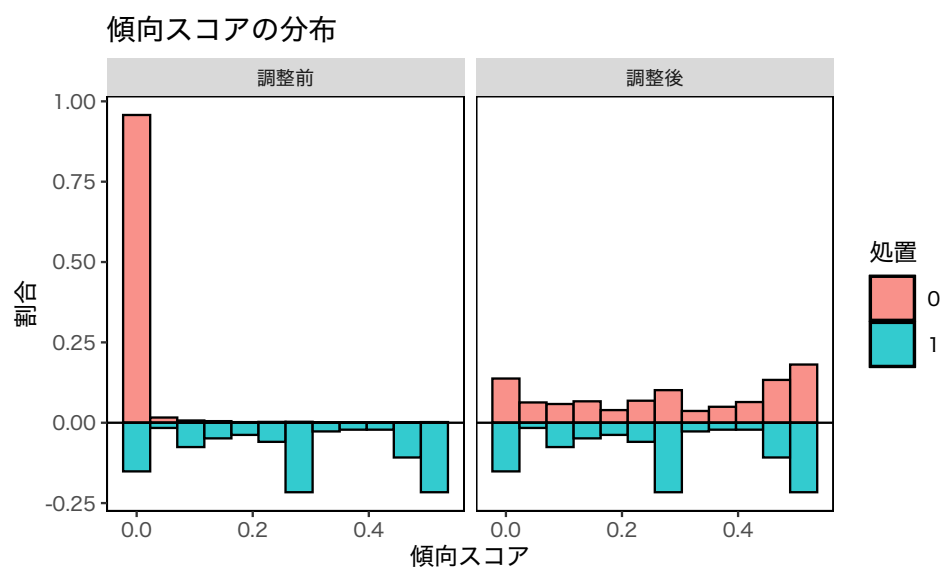


調整前と調整後を比較するために便利なパッケージとして `cobalt` がある。これを利用するためには、`WeightIt::weightit()` で重みを計算する必要がある(他の方法もあるが)。ここでは、上で計算した傾向スコアを使うために、`ps = myd$ps_logit` を指定する。ここで指定するのは重み(`w_att`)ではなく傾向スコアである。また、推定対象(estimand)に `ATT` をしている。

```
w_att2 <- WeightIt::weightit(ps_formula,
                             data      = myd,
                             ps        = myd$ps_logit,
                             estimand = 'ATT')
```

`weightit()` の結果を使うと、`cobalt` パッケージの `cobalt::bal.plot()` で重みによる調整前後の傾向スコアの分布が確認できる。

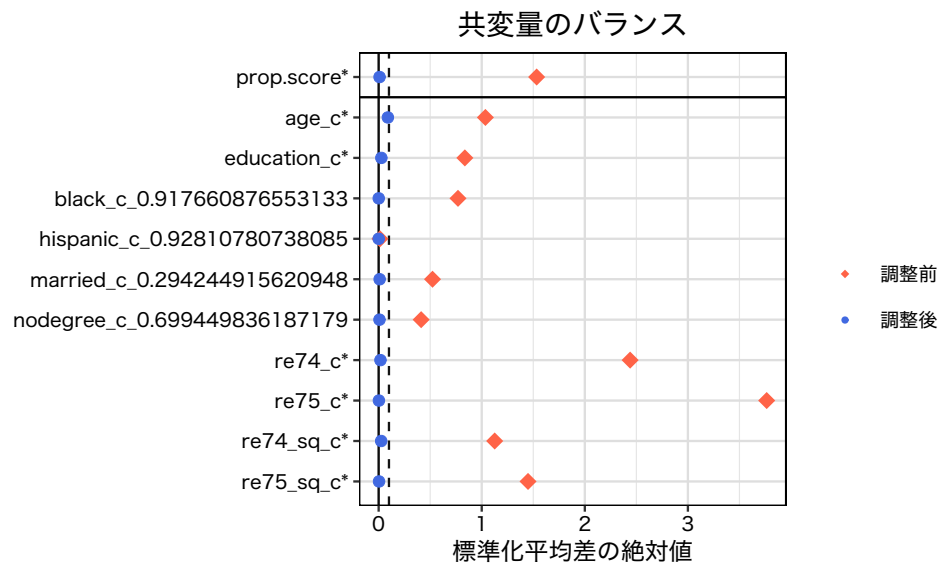
```
hist_w_att2 <- cobalt::bal.plot(w_att2,
                                var.name      = 'prop.score',
                                which         = 'both',
                                type         = 'histogram',
                                mirror       = TRUE,
                                sample.names = c('調整前', '調整後')) +
  scale_fill_discrete(name = '処置') +
  labs(x = '傾向スコア', y = '割合', title = '傾向スコアの分布')
plot(hist_w_att2)
```



処置群における因果効果を推定するために処置群の分布に合わせて、**統制群の分布が調整された**ことがわかる。

この調整によって、共変量のバランスが改善したかどうかを確認する。

```
bal_att <- cobalt::love.plot(w_att2,
                              thresholds    = 0.1,
                              abs           = TRUE,
                              grid          = TRUE,
                              shapes        = c(18, 20),
                              color         = c('tomato', 'royalblue'),
                              sample.names = c('調整前', '調整後'),
                              title         = '共変量のバランス',
                              stars         = 'std') +
  labs(x = '標準化平均差の絶対値',
       shape = '', size = '', stroke = '', color = '')
plot(bal_att)
```



すべての共変量について、重み付けによって処置群と統制群の間のバランスが改善し、標準化平均差の絶対値が 0.1 以下になっていることがわかる。

この重みを使い、 $ATT = E[Y(1)|D = 1] - E[Y(0)|D = 1]$ を推定する。 $E[Y(1)|D = 1]$ は、

```
e_y1_d1 <- myd |>
  filter(treat == 1) |>
  pull(re78) |>
  mean() |>
  print()
```

```
[1] 6349.144
```

と推定される。 $E[Y(0)|D = 1]$ は重みを使うと、

```
e_y0_d1 <- myd |>
  filter(treat == 0) |>
  with(weighted.mean(re78, w = w_att)) |>
  print()
```

```
[1] 5104.037
```

と推定される。これらの差を計算すると、ATT は、

```
e_y1_d1 - e_y0_d1
```

```
[1] 1245.106
```

であると推定される。RCT による ATE の推定結果は 1676.34 なので、効果が小さく推定されているが、重みを使わない単純な平均値の差に比べると、バイアスがかなり小さくなったことがわかる。

lm() で weight を指定すれば、これと同じ結果が得られる。

```
ps_weight_att1 <- lm(re78 ~ treat,
                     data = myd,
                     weight = w_att)
tidy(ps_weight_att1)
```

```
# A tibble: 2 x 5
  term      estimate std.error statistic  p.value
<chr>      <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  5104.        78.8        64.8  0
2 treat       1245.        111.         11.2 6.58e-29
```

WeightIt パッケージの weightit() で推定した重みを使っても、同じ結果が得られる。

```
ps_weight_att2 <- lm(re78 ~ treat,
                     data = myd,
                     weight = w_att2$weights)
tidy(ps_weight_att2)
```

```
# A tibble: 2 x 5
  term      estimate std.error statistic  p.value
<chr>      <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  5104.        78.8        64.8  0
2 treat       1245.        111.         11.2 6.58e-29
```

IPW: ATE (平均処置効果)の推定

続いて、ATE を推定するための重み付けについて説明する。ATE を推定するためには、**IPW** (inverse probability weighting; 逆確率重み付け) という方法を用いる。ATE = $\mathbb{E}[Y(1)] - \mathbb{E}[Y(0)]$ のそれぞれの項を、傾向スコアを利用した重み付けによって推定する。

$\mathbb{E}[Y(1)]$ の推定値は、重み

$$w_{1,i} = \frac{D_i}{e_i(X)}$$

を使って、

$$\mathbb{E}[\widehat{Y(1)}] = \sum \frac{w_{1,i} Y_i}{\sum w_{1,i}} = \sum \frac{\frac{D_i}{e_i(X)}}{\sum [\frac{D_i}{e_i(X)}]} Y_i$$

である。

このように、処置群と統制群のそれぞれについてその群に割り付けられる傾向スコアの逆数で重みが決まるので、この名前がついている。IPW では、それぞれの群で珍しい個体ほど重みが大きくなる。この重み、

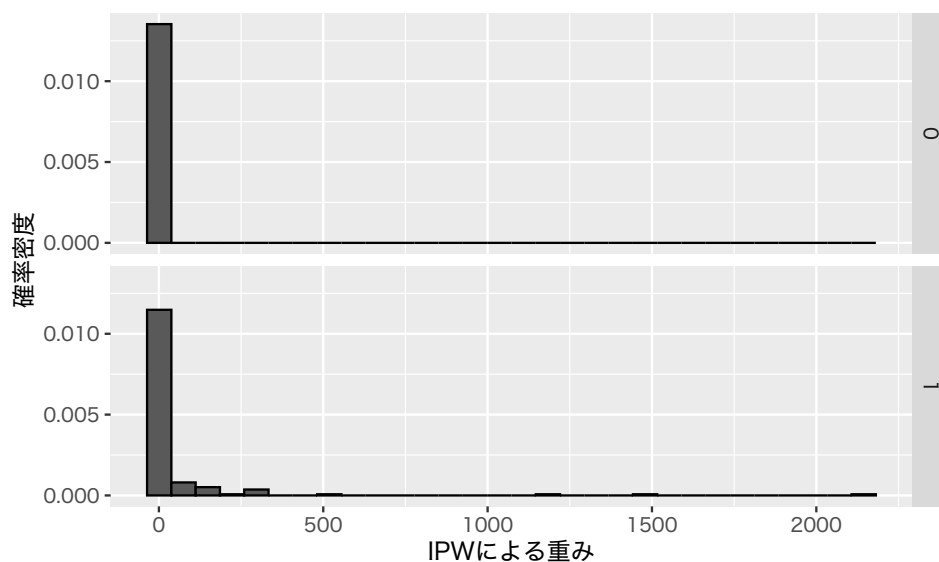
$$w_i^{\text{ATE}} = \frac{D_i}{e_i(X)} + \frac{1 - D_i}{1 - e_i(X)}$$

を計算する。

```
myd <- myd |>
  mutate(w_ate = ifelse(treat == 1,
                        1 / ps_logit,
                        1 / (1 - ps_logit)))
```

この重みがどのように分布しているか確認する。

```
hist_w_ate <- myd |>
  ggplot(aes(x = w_ate, y = after_stat(density))) +
  geom_histogram(color = 'black') +
  facet_grid(row = vars(treat)) +
  labs(x = 'IPW による重み', y = '確率密度')
plot(hist_w_ate)
```



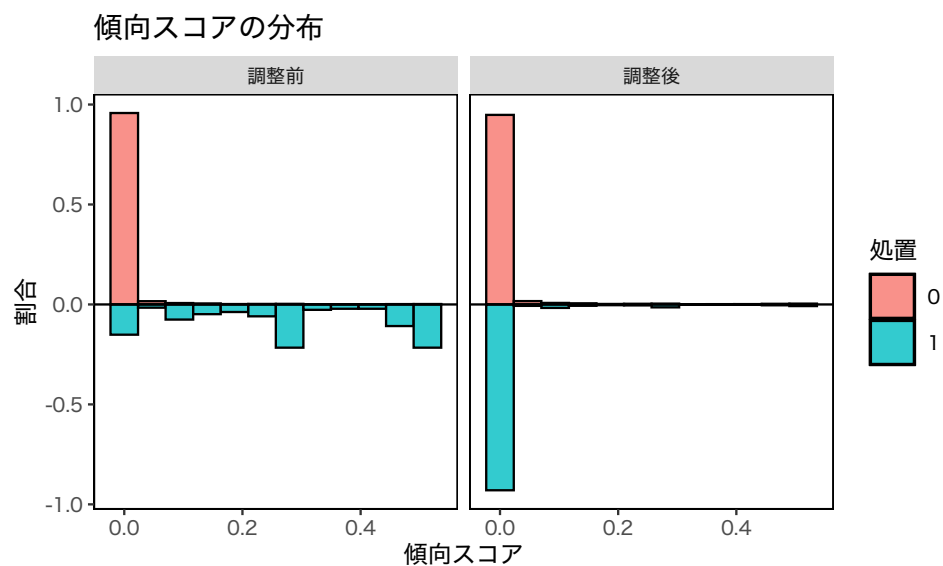
処置群で傾向スコアが大きい個体の重みが割り引かれている一方で、統制群に傾向スコアが大きい個体がないため、重みが割増されている個体はあまりないことがわかる。

`bal.plot()` と `love.plot()` を使うために、`weightit()` で重みを計算する。推定対象(estimand)に ATE を指定する。

```
w_ate2 <- WeightIt::weightit(ps_formula,
                             data      = myd,
                             ps        = myd$ps_logit,
                             estimand = 'ATE')
```

cobalt::bal.plot() で重みによる調製前後の傾向スコアの分布を確認する。

```
hist_w_ate2 <- cobalt::bal.plot(w_ate2,
                                var.name   = 'prop.score',
                                which      = 'both',
                                type       = 'histogram',
                                mirror     = TRUE,
                                sample.names = c('調整前',
                                                  '調整後')) +
  scale_fill_discrete(name = '処置') +
  labs(x = '傾向スコア', y = '割合', title = '傾向スコアの分布')
plot(hist_w_ate2)
```



調整後に、処置群の分布が統制群の分布に近づいていることがわかる。これは、処置群の観測数が 185 であるのに対して、統制群の観測数が 16,177 もあるためである。

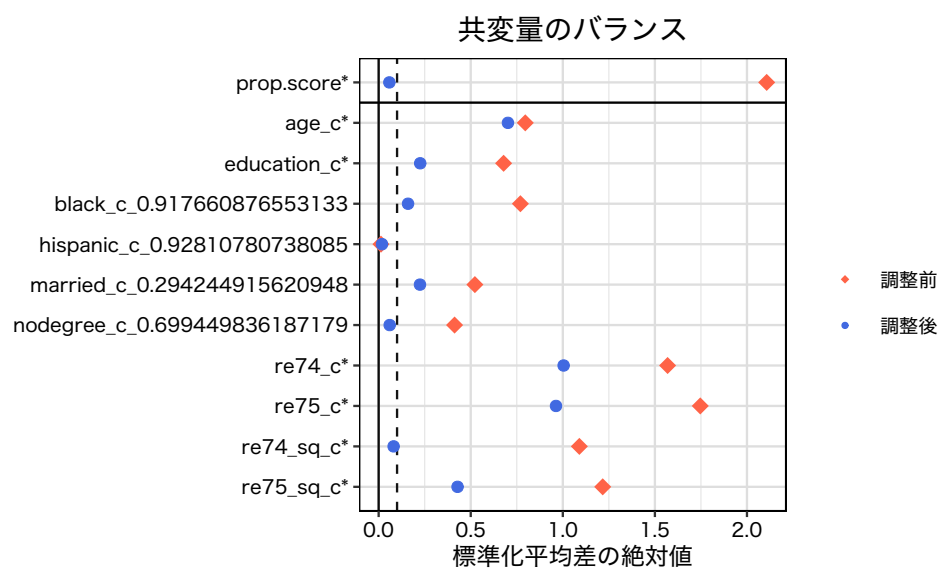
この調製によって、共変量のバランスが改善したかどうか確認する。

```
bal_ate <- cobalt::love.plot(w_ate2,
                              thresholds = 0.1,
                              abs        = TRUE,
                              grid       = TRUE,
                              shapes     = c(18, 20),
```

```

color      = c('tomato',
               'royalblue'),
sample.names = c('調整前',
                 '調整後'),
title      = '共変量のバランス',
stars      = 'std') +
labs(x = '標準化平均差の絶対値',
     shape = '', size = '', stroke = '', colour = '')
plot(bal_ate)

```



すべての共変量についてバランスの改善は見られるものの、標準化平均差の絶対値が 0.1 未満になった共変量は 3 つだけであり、処置群と統制群のバランスがあまりよくないことがわかる。そのため、このまま処置効果を推定してもうまくいかないことが予想される。

実際のデータ分析では、このような場合には処置効果を推定しないほうがいい。ここでは、推定をしてしまうとどうなるか見てみよう。

上で計算した重みを使い、 $ATE = E[Y(1)] - E[Y(0)]$ を推定する。まず、 $Y(1)$ の期待値の推定値は、

```

e_y1 <- myd |>
  filter(treat == 1) |>
  with(weighted.mean(re78, w = w_ate)) |>
  print()

```

```
[1] 7108.675
```

である。 $Y(0)$ の期待値の推定値は、

```
e_y0 <- myd |>
  filter(treat == 0) |>
  with(weighted.mean(re78, w = w_ate)) |>
  print()
```

```
[1] 14735.48
```

これらの差を取ると、ATE は、

```
e_y1 - e_y0
```

```
[1] -7626.806
```

であると推定される。安井(2020; 127-130)が説明する通り、推定に大きなバイアスがあることがわかる。

lm() で weight を指定すれば、これと同じ結果が得られる。

```
ps_weight_ate1 <- lm(re78 ~ treat,
                     data = myd,
                     weight = w_ate)
tidy(ps_weight_ate1)
```

```
# A tibble: 2 x 5
  term          estimate std.error statistic p.value
<chr>         <dbl>    <dbl>    <dbl>    <dbl>
1 (Intercept)  14735.      82.8     178.      0
2 treat        -7627.     135.    -56.5      0
```

WeightIt::weightit() で計算した重みを使っても同じ結果が得られる。

```
ps_weight_ate2 <- lm(re78 ~ treat,
                     data = myd,
                     weights = w_ate2$weights)
tidy(ps_weight_ate2)
```

```
# A tibble: 2 x 5
  term          estimate std.error statistic p.value
<chr>         <dbl>    <dbl>    <dbl>    <dbl>
1 (Intercept)  14735.      82.8     178.      0
2 treat        -7627.     135.    -56.5      0
```

このように、共変量のバランスが取れていない状況で因果効果を推定しようとするバイアスのある推定結果を得ることになるので注意が必要である。

■層別

推定した傾向スコアを利用して、サンプルを層(strata)にわけます。まず、層の数を決める必要があります。ここでは5にする(層の数については、Thoermmes and Kim [2011] を参照)。処置群と統制群の合計数がほぼ同じになるように、傾向スコアによって5つの層を作ろう。

```
myd |>
  mutate(subclass = cut(ps_logit,
                        breaks = quantile(ps_logit,
                                           prob = seq(0, 1, by = 0.2)),
                        include.lowest = TRUE)) |>
  with(table(treat, subclass))
```

```
      subclass
treat [3.72e-06,2.88e-05] (2.88e-05,7.85e-05] (7.85e-05,0.000463]
0      3238              3233              3235
1         0                0                0

      subclass
treat (0.000463,0.00293] (0.00293,0.513]
0      3231              3055
1         4               181
```

このような5つの層(stratum; subclass)ができる。

この層別では、3つの層で処置群の個体数が0になってしまった。そのため、それら3つの層別では処置効果が推定できない。この例から、処置群と統制群で傾向スコアの分布が大きく異なると層別が難しいことがわかる。統制群のほとんどの個体にとって処置群に比較対象となる個体が存在しない状況なので、ATE や ATC を推定するのは困難である。

MatchIt パッケージの `matchit()` を使えば、傾向スコアの推定と層別が一挙にできる。層別のために `method = 'subclass'` を指定する。また、層の数は `subclass = 5` にする。処置群と統制群の合計数、つまりサンプル全体の数を基準にした層別によって ATE を推定を目指すため、`estimand = 'ATE'` を指定する。

```
MatchIt::matchit(ps_formula,
                 data      = myd,
                 method    = 'subclass',
                 subclass  = 5,
                 estimand  = 'ATE') |>
  with(table(treat, subclass))
```

```
      subclass
treat   1    2    3    4    5
```

```

0 3235 3236 3235 3231 3055
1    1    1    1    1    1 181

```

先ほどとほぼ同じの 5 つの層ができた (MatchIt の機能により、0 のセルができないように調整されているが、実質的には上の層別と同じである)。

このデータでは、統制群の観測数に対して処置群の観測数が非常に少ないので、処置群の観測数が等しくなるような 5 層を作って ATT の推定を目指すことにしよう。

まず、処置群の個体を 5 分割するためのカットポイントを見つける。すべてのデータを含むために、最下限と最上限は 0 と 1 にする。

```

cutpoints <- myd |>
  filter(treat == 1) |>
  pull(ps_logit) |>
  quantile(prob = seq(0, 1, by = 0.2))
cutpoints[1] <- 0
cutpoints[6] <- 1

```

このカットポイントを使って、サンプルを 5 層に分ける。

```

myd |>
  mutate(subclass = cut(ps_logit,
                        breaks      = cutpoints,
                        include.lowest = TRUE)) |>
  with(table(treat, subclass))

```

```

      subclass
treat [0,0.0791] (0.0791,0.265] (0.265,0.299] (0.299,0.495] (0.495,1]
0      15607      239      38      79      29
1       37      37      37      37      37

```

このように、処置群の観測数が等しい 5 つの層ができた。

MatchIt::matchit() で層別を行ってみる。処置群に基づいて層別を行うことで ATT の推定を目指すため、estimand = 'ATT' を指定する。

```

strat <- MatchIt::matchit(ps_formula,
                          data      = myd,
                          method    = 'subclass',
                          subclass  = 5,
                          estimand  = 'ATT')
with(strat, table(treat, subclass))

```

```

      subclass

```

```
treat      1      2      3      4      5
      0 15607   239    38    79    29
      1   37    37    37    37    37
```

上と同じ層別ができた。層の名前が整数値で 1 から 5 担っているこちらの層別のほうが便利なので、以下ではこのカテゴリ変数を使う。

```
myd$subclass <- strat$subclass
```

バランスチェックは、MatchIt パッケージの機能を使って行うことができる。MatchIt::matchit() の結果に対し、summary() を使う。

```
summary(strat, standardize = TRUE)
```

Call:

```
MatchIt::matchit(formula = ps_formula, data = myd, method = "subclass",
  estimand = "ATT", subclass = 5)
```

Summary of Balance for All Data:

	Means Treated	Means Control	Std. Mean Diff.	Var. Ratio	eCDF Mean
distance	0.2835	0.0083	1.5328	17.1199	0.4935
age_c	-7.3243	0.0847	-1.0355	0.4196	0.1863
education_c	-1.6623	0.0192	-0.8363	0.4905	0.0908
black_c	0.7609	-0.0088	2.1113	1.9506	0.3849
hispanic_c	-0.0124	0.0001	-0.0530	0.8411	0.0063
married_c	-0.5166	0.0060	-1.3306	0.7517	0.2613
nodegree_c	0.4076	-0.0047	0.9044	0.9975	0.2061
re74_c	-11784.8956	136.3310	-2.4396	0.2607	0.4591
re75_c	-11980.1583	138.5899	-3.7645	0.1206	0.4751
re74_sq_c	1.0393	-0.0120	1.1257	0.8820	0.2455
re75_sq_c	1.0529	-0.0122	1.4492	0.5443	0.2759

	eCDF Max
distance	0.8156
age_c	0.3427
education_c	0.4123
black_c	0.7697
hispanic_c	0.0126
married_c	0.5225
nodegree_c	0.4123
re74_c	0.6031
re75_c	0.6509

re74_sq_c	0.5939
re75_sq_c	0.5663

Summary of Balance Across Subclasses

	Means Treated	Means Control	Std. Mean Diff.	Var. Ratio	eCDF Mean
distance	0.2835	0.2706	0.0716	0.9515	0.0834
age_c	-7.3243	-6.6073	-0.1002	0.4197	0.0897
education_c	-1.6623	-1.1273	-0.2661	0.5931	0.0323
black_c	0.7609	0.7113	0.1360	0.8090	0.0248
hispanic_c	-0.0124	-0.0406	0.1190	1.8520	0.0141
married_c	-0.5166	-0.4607	-0.1422	0.8311	0.0279
nodegree_c	0.4076	0.2816	0.2763	0.8517	0.0630
re74_c	-11784.8956	-10024.1557	-0.3603	0.4425	0.0712
re75_c	-11980.1583	-10126.7011	-0.5757	0.2228	0.0723
re74_sq_c	1.0393	0.9158	0.1323	1.1926	0.0220
re75_sq_c	1.0529	0.9758	0.1048	0.8292	0.0217

	eCDF Max
distance	0.1446
age_c	0.1992
education_c	0.1259
black_c	0.0496
hispanic_c	0.0282
married_c	0.0559
nodegree_c	0.1259
re74_c	0.2122
re75_c	0.1069
re74_sq_c	0.2176
re75_sq_c	0.0938

Sample Sizes:

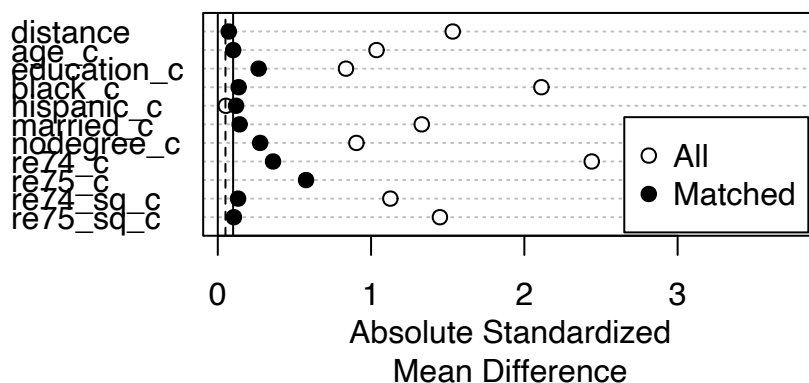
	Control	Treated
All	15992.	185
Matched (ESS)	321.73	185
Matched	15992.	185
Unmatched	0.	0
Discarded	0.	0

サンプル全体での共変量のバランス (Summary of Balance for All Data) と、層間での共変量のバランス (Summary of Balance Across Subclasses) が表示されている。ほとんどの共変量について、全体よりも層別したほうが標準化平均差 (Std. Mean Diff.) の絶対値が小さくなっている。しかし、基準である 0.1 または 0.25 よ

りも大きな差が複数残されており、層別によるバランスはあまり良くないことがわかる。ヒスパニックであることを示す変数については全体でのバランスが元々良いため、層別によってバランスがむしろ悪化している(バランスの改善率 [Percent Balance Improvement] がマイナス 124% である)。

図でバランスチェックをするには、例えば次のようにする。

```
strat |>
  summary(standardize = TRUE) |>
  plot()
```



このように、処置群と統制群の間のバランスが悪いので、実際のデータ分析であれば因果効果を推定しない方が良い。しかし、ここではこの層別を利用して ATT を推定するとどうなるか確認してみよう。

まず、層別の ATT を推定する。各層で処置群と統制群の平均値の差を計算すれば良い。

```
subclass_att <- myd |>
  group_by(subclass, treat) |>
  summarize(mean = mean(re78),
             .groups = 'drop_last') |>
  group_by(subclass) |>
  summarize(att = diff(mean),
             .groups = 'drop') |>
  pull(att) |>
  print()
```

```
[1] -8198.9656 1112.3861 4218.7575 2462.4248 -254.7994
```

この層別の加重平均が ATT である。その際の重みは、各層に属する処置群の個体数を、処置群全体の個体数で割ったものである。同じ個体数で 5 つの層に分けたので当然全ての重みが 0.2 であるが、一応計算しておく。

```
w_subclass <- myd |>
  filter(treat == 1) |>
  with(table(subclass) / sum(table(subclass))) |>
  print()
```

```
subclass
  1   2   3   4   5
0.2 0.2 0.2 0.2 0.2
```

(ちなみに、ATE を推定する場合の各層の重みは、各層に属する個体数をサンプルサイズで割ったものである。つまり、全サンプルに占める各層の割合が重みとなる。)

よって、ATT の推定値は、

```
weighted.mean(subclass_att, w = w_subclass)
```

```
[1] -132.0394
```

```
# すべて同じ重みなので、単純平均(`mean(subclass_att)`)でも同じ
```

である。効果が過小推定されており、バイアスが残っていることがわかる。やはり、**バランスが悪いのに因果効果を推定しようとしてはいけない。**

注意

survey パッケージが正しく動作しない(バージョンが違うのかも)。

標準誤差も知りたいので、survey パッケージを利用して計算しよう。まず、`survey::svydesign()` で利用するデータセットを指定する。引数 `ids` が必須だが、ここでは `id` (観測個体の識別子) を使用した分析を行わないので `ids = ~ 0` とする(`svydesign()` の引数の `strata` があるが、これはサーベイの層別を指定するための引数で、傾向スコアによる層別を指定するものではないので注意。ここでは何も指定しない)。

```
survey_design <- survey::svydesign(ids = ~ 0, data = myd)
```

次に、`survey::svyby()` を使って、層別に結果変数の平均値を計算する。`formula` に平均値を計算したい結果変数を指定し、`by` に処置変数と層別の変数を指定する。平均値を計算するので、`FUN = svymean` とする。

```
sub_means <- survey::svyby(formula = ~ re78,
                           by      = ~ treat + subclass,
                           design  = survey_design,
                           FUN     = svymean) |>
print()
```

このように、各層別に、処置群と統制群のそれぞれで結果変数の平均値が計算される。

最後に ATT の推定値と標準誤差を、`survey::svycontrast()` で計算する。そのために、重みをリストで与える必要がある。上の表(処置・層別の平均値の表)で表示されている順番に重みを指定する。その際、処置群には先程計算した重みである 0.2 を与え、統制群にはそれを負の値にしたものを与える。

```
survey::svycontrast(sub_means,
                    list(ATT = rep(c(-0.2, 0.2), 5)))
```

上で計算したものと同一推定値が得られた。また、標準誤差は 636.8 であることがわかった(これは有意水準 0.05 で統計的に有意ではない)。

このように、調査・観察データを使った因果推論には、様々な工夫が必要であり、慎重に分析を進める必要がある。また、さまざまな仮定をおいて分析を行っており、仮定が成り立たない場合には因果効果を正しく推定できない。自分がどのような仮定に基づいて、何を推定対象(estimand)としたデータを分析しているのか、常に意識するようにしよう。