

Lab 13: Covariate Balancing and CBPS: Answers

Eric Parajon with code adapted from Colin Case

Weighting

Randomized controlled trials (RCTs) are often seen as the “gold standard” for estimating the effects of treatments, interventions, and exposures on outcomes. However, in the social sciences, it is not always possible to achieve randomization (and even in experimental studies, this doesn’t happen either!), and we may have unobserved confounding variables that affect someone’s likelihood of receiving treatment. Weighting helps adjust for the covariate imbalance between the treated and the control groups in a study. The weights on observations can help us “balance-out” the influence of each observation on the causal relationship. In this manner, we are making it so the treatment and control groups are *balanced* on average when considering observable characteristics (we obviously cannot balance on unobservable characteristics).

In this exercise we will use the `lalonde` dataset from the `cobalt` package, which is data on the National Supported Work Demonstration, an experimental public employment assistance program in the US in the 1970s. The question we are trying to answer here is: does the participation in the national supported work demonstration (`treat`) help workers gain a higher wage? In thinking about this question, consider who is likely to opt-into this program if it is optional and not randomly assigned?

The outcome variable of interest is workers’ earnings in 1978 `re78`. We are going to control for a couple of things such as pre-treatment wages `re74`, `re75`, if the person earned a degree `nodegree`, marital status `married`, as well as other demographic variables.

Examining Covariate Balance

To start, we are going to look at covariate balance between the treated and the control groups. We are going to use the `bal.tab` function from the `cobalt` package, and we specify that we are interested in the average treatment effect on the treated (ATT):

```
# Clear Environment
rm(list = ls())
# Load Packages

pacman::p_load(WeightIt,
               cobalt,
               tidyverse,
               stargazer,
               survey,
               jtools,
               CBPS,
               lmtest,
               sandwich)

# Load dataset Lalonde from cobalt package
data('lalonde', package = 'cobalt')
```

```
# Evaluate covariate balance on age, educ, married, nodegree, re74, and re75
bal.tab(treat ~ age + educ + married + nodegree + re74 + re75,
        data = lalonde, estimand = "ATT", m.threshold = .05)
```

```
## Balance Measures
##           Type Diff.Un      M.Threshold.Un
## age      Contin. -0.3094 Not Balanced, >0.05
## educ      Contin.  0.0550 Not Balanced, >0.05
## married   Binary -0.3236 Not Balanced, >0.05
## nodegree  Binary  0.1114 Not Balanced, >0.05
## re74      Contin. -0.7211 Not Balanced, >0.05
## re75      Contin. -0.2903 Not Balanced, >0.05
##
## Balance tally for mean differences
##                count
## Balanced, <0.05      0
## Not Balanced, >0.05   6
##
## Variable with the greatest mean difference
## Variable Diff.Un      M.Threshold.Un
##      re74 -0.7211 Not Balanced, >0.05
##
## Sample sizes
##      Control Treated
## All      429      185
```

As you can see from the output, none of the variables are balanced. Those who are older, married, and had higher income in 1974 and 1975 are much less likely to attend training (i.e. receive the treatment). Those without a high school degree and those who are more educated are more likely to attend job training. This raises questions regarding the ATT of the treatment in 1978 : Is it a treatment effect, or a function of the pre-existing differences that have not been taken into account in the treatment assignment process.

Propensity Scores

We will now try to address this by computing propensity score weights. To recall, propensity scores are probabilities of a unit (e.g. a subject) being assigned to a particular treatment given a set of factors/ covariates:

$$e(x) = Pr(T = 1|X = x)$$

for T is the treatment assignment dummy and X is a matrix of covariates.

Propensity Score weighting uses this quantity to adjust for the influence of a particular observation in the treated and control groups. It turns out the weights for the ATT are defined as $w = 1$ for the treated group and $w = \frac{e(x)}{1-e(x)}$ for the control group.

Let's try to compute this by hand. We will first fit a simple GLM regressing treatment on the covariates:

```
# Estimate glm predicting treatment status
ps.mod <- glm(treat~ age + educ + married + nodegree + re74 + re75,
              data = lalonde, family = "binomial")
# Create predictions
ps <- predict(ps.mod, type = "response")
# View propensity scores
head(ps)
```

```
##           1           2           3           4           5           6
## 0.2963350 0.4624595 0.3913369 0.5575412 0.4827900 0.4624595
```

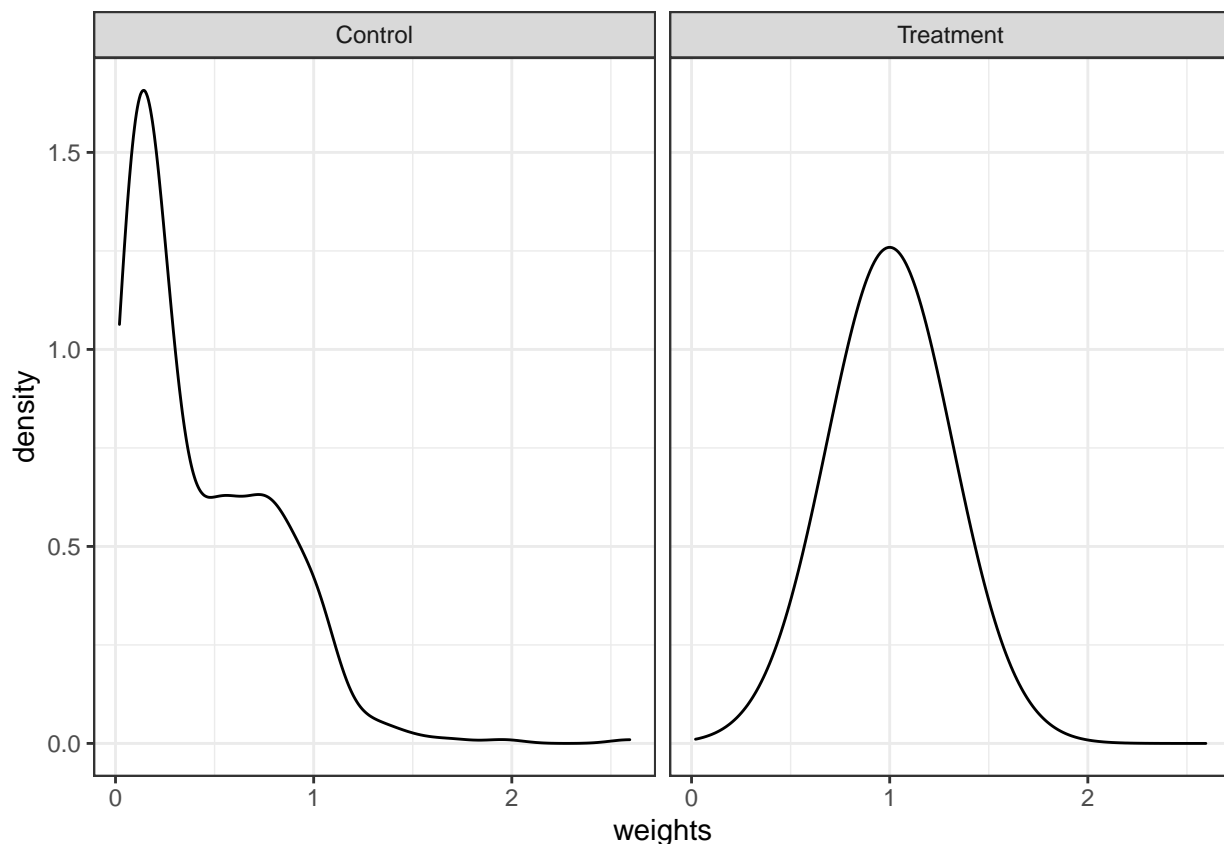
These probabilities are our propensity scores $e(\hat{x})$. Now, let's compute the weights:

```
# Create variable name weights in lalonde df
lalonde$weights <- ifelse(lalonde$treat == 1, 1, # w=1 for treatment
                          ps/(1-ps)) # propensity score/(1-propensity score)
```

We should always check the distribution of our weights. If there are extreme values in the weights computed, we might need to use another weighting method, or trim some of the cases:

```
# Subset to treatment and weights
weights_dist <- lalonde %>% dplyr::select(treat, weights)
# Add labels
weights_dist$treat <- ifelse(weights_dist$treat == 0, "Control", "Treatment")

# Plot weight density by condition looking for outliers
ggplot(weights_dist, aes(x=weights))+
  geom_density()+
  facet_wrap(~treat)+
  theme_bw()
```



As we can see, the control weights are OK - we don't have extreme values. The computed weights range between 0-2. For the treatment group, all weights are equal to 1, so we don't have to worry about it in this case. If the causal estimand of interest is not ATT, then weights are computed differently for the treatment group and you may see some variations in the calculated weights for the treated.

And these are then the weights for the ATE for each observation. We can now refit the model with these

weights using the `weights=` parameter in `lm()`. Since we didn't weight by race, we also want to make sure we control for that factor in this model. Let's compare the results between the model without weighting and with weighting:

```
# Estimate and View model without weights
lalonde_model_noW <- lm(re78 ~ treat + race, data=lalonde)
summary(lalonde_model_noW)

##
## Call:
## lm(formula = re78 ~ treat + race, data = lalonde)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8466  -5839  -2105   3986   54310
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5101.7      709.7   7.188 1.92e-12 ***
## treat           896.2      819.5   1.094  0.2746
## racehispan    1868.1     1073.9   1.740  0.0824 .
## racewhite     2468.5       799.1   3.089  0.0021 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7425 on 610 degrees of freedom
## Multiple R-squared:  0.01702, Adjusted R-squared:  0.01218
## F-statistic:  3.52 on 3 and 610 DF, p-value: 0.01491

# Estimate and view model with weights
lalonde_model_w <- lm(re78 ~ treat + race, data = lalonde, weights = weights)
summary(lalonde_model_w)

##
## Call:
## lm(formula = re78 ~ treat + race, data = lalonde, weights = weights)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
##  -9017  -3274   -862   2627   54193
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4669.3      650.6   7.177 2.08e-12 ***
## treat           1445.2      698.2   2.070  0.0389 *
## racehispan    1773.9     1011.7   1.753  0.0800 .
## racewhite     1327.4       753.2   1.762  0.0785 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5472 on 610 degrees of freedom
## Multiple R-squared:  0.0094, Adjusted R-squared:  0.004528
## F-statistic:  1.93 on 3 and 610 DF, p-value: 0.1236
```

As you can see, after weighting our observations, the treatment is now significant whereas without weights it would be not significant!

The WeightIt package

Now that you understand how weighting works under the hood, we will turn to the `WeightIt` package that simplifies this process. `WeightIt` is a very powerful package for computing different kinds of weights and examining the weighting result. Let's look at how you can compute weights with `WeightIt`.

You compute the weights with the `weightit` function, which uses the same set of parameters from the `bal.tab` function:

```
# Use weightit to calculate weights
W.out <- weightit(treat ~ age + educ + married + nodegree + re74 + re75,
                 data = lalonde, estimand = "ATT", method = "ps")
W.out
```

```
## A weightit object
## - method: "ps" (propensity score weighting)
## - number of obs.: 614
## - sampling weights: none
## - treatment: 2-category
## - estimand: ATT (focal: 1)
## - covariates: age, educ, married, nodegree, re74, re75
```

This creates a `Weightit` object, which gives you a summary of the weight computation outcome. You can get more information out of it by calling `summary()` on it:

```
# View w.out
summary(W.out)
```

```
##                               Summary of weights
##
## - Weight ranges:
##
##               Min                               Max
## treated 1.0000                               1.0000
## control 0.0199 |-----| 2.5956
##
## - Units with 5 most extreme weights by group:
##
##               6      4      3      2      1
## treated      1      1      1      1      1
##               595    264    409    269    296
## control 1.4361 1.5373 1.7035 1.9587 2.5956
##
## - Weight statistics:
##
##               Coef of Var   MAD Entropy # Zeros
## treated      0.000 0.000  -0.000      0
## control      0.855 0.711   0.345      0
##
## - Effective Sample Sizes:
##
##               Control Treated
## Unweighted    429.      185
## Weighted      248.09    185
```

Now let's reexamine covariate balance on the `weightit` object:

```
# Re-examine covariate balance after weighting
bal.tab(W.out, m.threshold = .05, disp.v.ratio = TRUE)
```

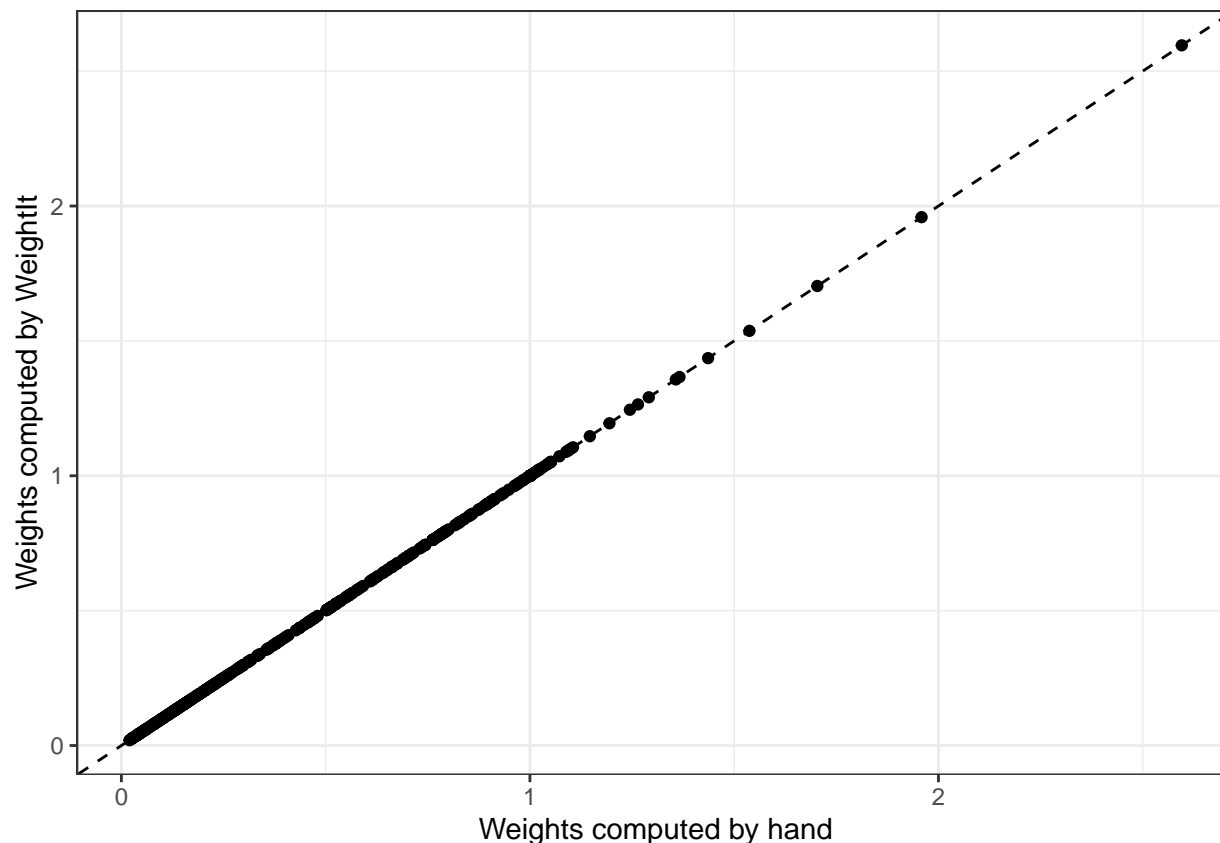
```
## Balance Measures
##           Type Diff.Adj      M.Threshold V.Ratio.Adj
## prop.score Distance   0.0010 Balanced, <0.05    1.0046
## age           Contin.   0.0349 Balanced, <0.05    0.4168
## educ           Contin. -0.0479 Balanced, <0.05    0.6024
## married        Binary   0.0064 Balanced, <0.05      .
## nodegree        Binary   0.0115 Balanced, <0.05      .
## re74           Contin. -0.0220 Balanced, <0.05    1.5241
## re75           Contin. -0.0192 Balanced, <0.05    1.7637
##
## Balance tally for mean differences
##           count
## Balanced, <0.05      7
## Not Balanced, >0.05    0
##
## Variable with the greatest mean difference
## Variable Diff.Adj      M.Threshold
##      educ -0.0479 Balanced, <0.05
##
## Effective sample sizes
##           Control Treated
## Unadjusted  429.      185
## Adjusted    248.09    185
```

Now you can see that every covariate is balanced between the treated and the control group. Now, let's try to compare the hand-computed weights and the weights generated by `WeightIt` to validate our calculations:

```
# Merge hand and weightit weights
wdf <- data.frame(hand = lalonde$weights,
                  weightit = get.w(W.out))
# Calculate correlation
cor(wdf)
```

```
##           hand weightit
## hand         1         1
## weightit     1         1
```

```
# Plot Weights
ggplot(wdf, aes(x=hand, y=weightit))+
  geom_point()+
  geom_abline(slope=1, lty=2)+
  xlab("Weights computed by hand")+
  ylab("Weights computed by WeightIt")+
  theme_bw()
```



And the correlation matrix confirms our calculations - they are exactly the same as those generated by `WeightIt` and they fall perfectly along the 45 degrees diagonal line (perfect correlation).

There is an alternative method, called Covariate Balancing Propensity Score Weighting (CBPS), that performs a similar procedure than the `ps` method we used above. The difference comes in that CBPS estimates propensity scores such that both covariate balance and prediction of treatment assignment are maximized. The method, therefore, avoids an iterative process between model fitting and balance checking and implements both simultaneously. Let's see how these weights compare to the `ps` process we used in `WeightIt`.

```
CPBS.out <- CBPS(treat ~ age + educ + married + nodedegree + re75 + re74,
  data = lalonde, ATT = TRUE)
```

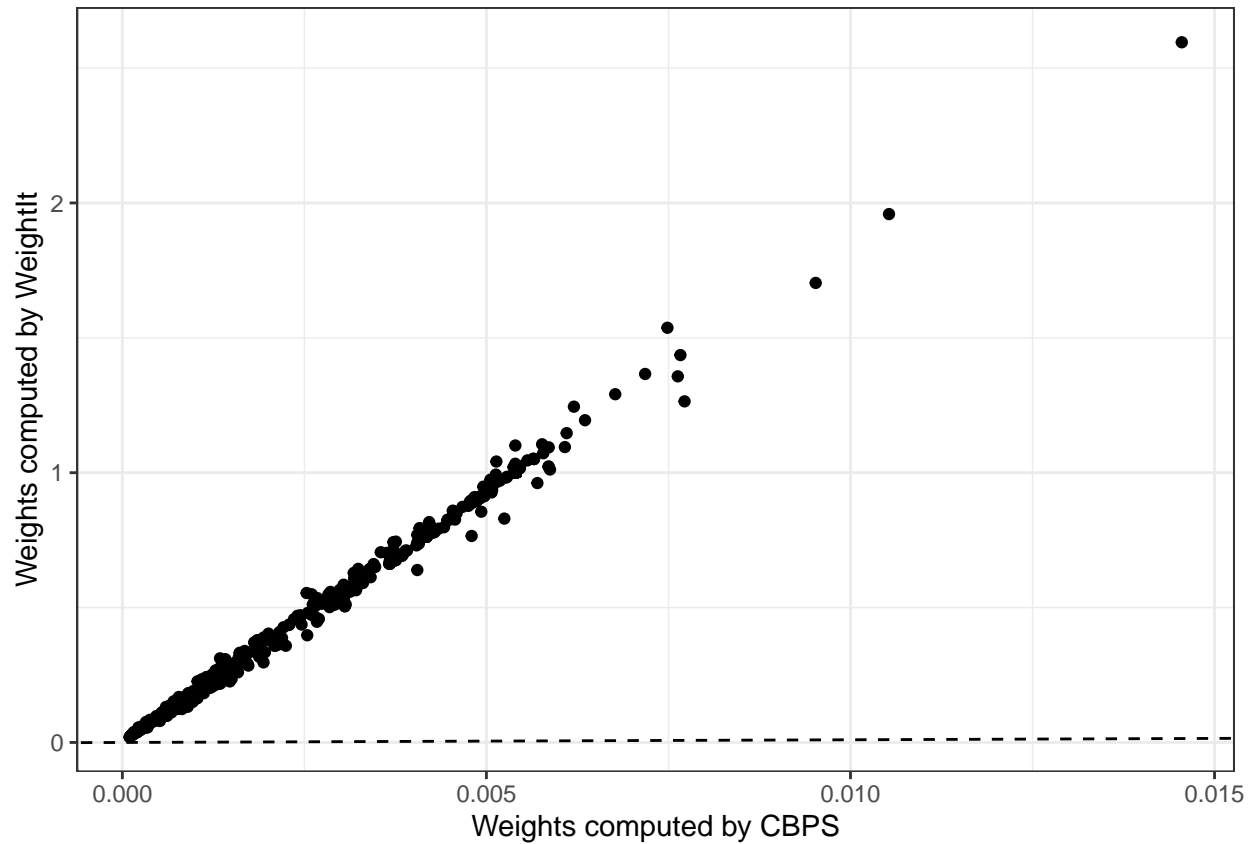
```
## [1] "Finding ATT with T=1 as the treatment. Set ATT=2 to find ATT with T=0 as the treatment"
```

```
# Merge hand and weightit weights
wdf <- data.frame(cbps = CPBS.out$weights,
  weightit = get.w(W.out))
# Calculate correlation
cor(wdf)
```

```
##           cbps weightit
## cbps      1.000000 0.998351
## weightit 0.998351 1.000000
```

```
# Plot Weights
ggplot(wdf, aes(x=cbps, y=weightit))+
  geom_point()+
  geom_abline(slope=1, lty=2)+
  xlab("Weights computed by CBPS")+
  ylab("Weights computed by WeightIt")
```

```
ylab("Weights computed by WeightIt")+
theme_bw()
```



As you can see, the two are still highly correlated, but with some slight differences this time. Let's now see what this looks like when we include the weights in the regression output. We first need add the weights from `W.out` to the data frame, and then use `lm` to estimate the model.

```
# Model using WeightIt Weights
fit1 <- lm(re78 ~ treat + race, data = lalonde, weights = get.w(W.out))
summary(fit1)
```

```
##
## Call:
## lm(formula = re78 ~ treat + race, data = lalonde, weights = get.w(W.out))
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -9017  -3274   -862    2627   54193
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4669.3      650.6   7.177 2.08e-12 ***
## treat         1445.2      698.2   2.070  0.0389 *
## racehispan    1773.9     1011.7   1.753  0.0800 .
## racewhite     1327.4      753.2   1.762  0.0785 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
##
## Residual standard error: 5472 on 610 degrees of freedom
## Multiple R-squared:  0.0094, Adjusted R-squared:  0.004528
## F-statistic: 1.93 on 3 and 610 DF, p-value: 0.1236

# Model using CBPS Weights
fit2 <- lm(re78 ~ treat + race, data = lalonde, weights = CPBS.out$weights)
summary(fit2)

##
## Call:
## lm(formula = re78 ~ treat + race, data = lalonde, weights = CPBS.out$weights)
##
## Weighted Residuals:
##      Min       1Q   Median       3Q      Max
## -654.7  -240.9   -59.5   195.2  3984.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4555.8      647.0    7.041 5.15e-12 ***
## treat         1555.7      695.0    2.238  0.0256 *
## racehispan    1824.7     1008.8    1.809  0.0710 .
## racewhite     1327.8      750.1    1.770  0.0772 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 401.1 on 610 degrees of freedom
## Multiple R-squared:  0.01045, Adjusted R-squared:  0.00558
## F-statistic: 2.147 on 3 and 610 DF, p-value: 0.09323
```

Survey Weighted Regression

How can we make inference out of these weights? We have explored `lm()` and `glm()` with the `weights` option, which finds:

$$\operatorname{argmin}_{\beta} w_i \sum_{i=1}^N (y_i - X_i \beta)^2$$

where w_i refers to the propensity score weights for individual i .

It turns out using `lm()` is not giving us an accurate estimation of the ATT because it only applies the weights on the errors in the least squares estimation (aka Weighted Least Squares - WLS). However, we should also consider the loss of precision due to the presence of sampling weights:

$$\operatorname{argmin}_{\beta} s_i w_i \sum_{i=1}^N (y_i - X_i \beta)^2$$

where s_i is the sampling weight in a survey setting. This type of survey-weighted linear regression model can be fitted with the `survey` package:

```
# Specify svydesign with no clusters
d.w <- svydesign(ids = ~1, weights = get.w(W.out),
               data = lalonde)
# Fit model taking into account survey weights
fit <- svyglm(re78 ~ treat, design = d.w)
```

```
# View Fit
summ(fit, confint = TRUE)
```

Observations	614
Dependent variable	re78
Type	Survey-weighted linear regression

R ²	0.00
Adj. R ²	0.00

	Est.	2.5%	97.5%	t val.	p
(Intercept)	5701.84	4955.51	6448.18	15.00	0.00
treat	647.30	-710.08	2004.68	0.94	0.35

Standard errors: Robust

And it turns out after adjusting for survey-weights, we found the effect of the treatment is actually not statistically significant after considering for survey sampling weights on top of propensity score weights.

If you have survey weights already, one way you could incorporate them in `WeightIt` is to specify the `s.weights` argument in `weightit()`. Then, the ATT weights are the product of the propensity score weights and the survey weights supplied. Using these ATT weights, you can then fit the WLS as usual but the standard errors will have to be adjusted by using the sandwich estimator (`sandwich::vcovHC`).

Exercise: Priming Norwegian Voters about the Municipal Reform, 2014

In Norway, there is a lot of small municipalities (similar to U.S. Counties) scattered across the country. There has been a debate on merging these small municipalities for roughly 2 decades. Proponents for the reform argue that this helps achieve the economies of scale of providing services to the localities, whereas opponents argue that each locality is unique. Given that the center-periphery cleavage is quite salient in Norway and overlaps with the linguistic conflict between Nynorsk and Bokmål speakers, municipal reform was a highly salient political issue in Norway at the time.

In 2014, the Norwegian Citizens Panel (NCP) (Wave 3) conducted a randomized experiment on the respondents by coincidence. There was a question about whether the participant would pay 1000 Norwegian Krone (~\$115) to maintain the current municipal structure. This primes the respondent that the current municipal structure can be costly and merging them could save the taxpayers' money. Note that the assignment of this group of questions about municipal reforms is random, so even if the priming is not intentional, it remains a valid experimental treatment nonetheless.

In this exercise, you are going to explore **if priming respondents the cost of the existing municipal structure would lead to support for the Conservative Party**. The Conservative Party of Norway supports municipal reforms of cutting down the number of municipalities, along with the Center Party and Progress Party. The Conservative Party would go on to form a center-right cabinet after emerging second in the 2017 parliamentary election. They will go on to implement the reform after forming a center-right government. The reforms were eventually completed on 1/1/2020.

Since the treatment assignment is unintentional, it is unclear if the covariates are balanced for the treated and control group. Therefore, it is necessary to check covariate balance and generate weights to balance out the influence of particular observations given their covariate values.

In the dataset `ncp_final.rds`, I have attached the survey weights from the NCP for you. The variables are as follows:

- **conservative** : Self-reported affinity to the Conservative Party of Norway. Continuous. 1 = Intensely dislike; 7= Intensely like.
- **treat** : If the respondent has been asked the question : “Would you rather pay NOK 1000 per year to maintain the current municipal structure?”. Dummy.
- **interest** : Self-reported Political Interest. Continuous. 1= Very Interested; 5 = Not Interested.
- **gender** : Biological Sex. Dummy. 1 = Male, 2= Female.
- **region** : Region of the respondent. 1 = Oslo. 2= Østlandet. 3 = Sørlandet. 4= Vestlandet. 5=Trøndelag. 6 = Nord-Norge.
- **edu** : Highest level of education attained. 1 = No Education/ Elementary School. 2= Upper Secondary School. 3= Univesity/ Community College.
- **yob** : Year of Birth. 1 = 1939 or earlier. 2= 1940-1949. 3= 1950-1959. ... 7= 1990 or later.
- **weight** : Survey weights from NCP. Continuous.

Take note that this dataset already has survey weights which are different from the weights we will calculate.

Using `bal.tab`, look at the balance of demographic variables (`gender`, `region`, `edu`, `yob`) on the treatment before and after weighting. Focus on the ATT. How do they look?

```
# Load data
ncp_final <- readRDS("ncp_final.rds")

# Look at balance of demographic variables
bal.tab(treat~ gender + region + edu + yob,
        data = ncp_final, estimand = "ATT", m.threshold = .05)
```

```
## Balance Measures
##           Type Diff.Un      M.Threshold.Un
## gender_2   Binary -0.0275    Balanced, <0.05
## gender:<NA> Binary -0.5935 Not Balanced, >0.05
## region     Contin.  0.0367    Balanced, <0.05
## edu        Contin.  0.0698 Not Balanced, >0.05
## edu:<NA>    Binary -0.5713 Not Balanced, >0.05
## yob        Contin. -0.0816 Not Balanced, >0.05
## yob:<NA>    Binary -0.5935 Not Balanced, >0.05
##
## Balance tally for mean differences
##           count
## Balanced, <0.05      2
## Not Balanced, >0.05   5
##
## Variable with the greatest mean difference
## Variable Diff.Un      M.Threshold.Un
## yob:<NA> -0.5935 Not Balanced, >0.05
##
## Sample sizes
##           Control Treated
## All    17843    1261
```

The balance table suggests that education and year of birth are unbalanced between the treated and the control group. Use `weightit` to weight them (you should do this even on variables that are listed as balanced).

```
# Create weights
W.out <- weightit(treat ~ gender + region + edu + yob,
                 data = ncp_final, estimand = "ATT", method = "ps")

# Re-check balance with weights
bal.tab(W.out, m.threshold = .05, disp.v.ratio = TRUE)
```

```
## Balance Measures
##           Type Diff.Adj      M.Threshold V.Ratio.Adj
## prop.score Distance -0.0019 Balanced, <0.05      0.9779
## gender_2      Binary  0.0002 Balanced, <0.05      .
## gender:<NA>    Binary -0.0000 Balanced, <0.05      .
## region        Contin. -0.0004 Balanced, <0.05      0.9789
## edu           Contin. -0.0008 Balanced, <0.05      0.9787
## edu:<NA>       Binary  0.0000 Balanced, <0.05      .
## yob           Contin. -0.0004 Balanced, <0.05      1.0178
## yob:<NA>       Binary -0.0000 Balanced, <0.05      .
##
## Balance tally for mean differences
##           count
## Balanced, <0.05      8
## Not Balanced, >0.05  0
##
## Variable with the greatest mean difference
## Variable Diff.Adj      M.Threshold
##      edu -0.0008 Balanced, <0.05
##
## Effective sample sizes
##           Control Treated
## Unadjusted 17843.      1261
## Adjusted   7102.26     1261
```

Regress the outcome variable on the treatment, `interest`, and the demographic variables. Fit two models: 1) One with no adjustment; 2) a Survey-Weighted GLM with propensity score weights. Then, use `summ` to take a look at the 95% Confidence Interval on the effect of the treatment from the survey-weighted GLM.

```
# without any adjustment
tmod1 <- lm(conservative ~ treat + interest + as.factor(gender) +
            as.factor(region)+ as.factor(edu) + as.factor(yob) , ncp_final)

# With Survey-weights & PS weights adjustment

# Create survey weights using svydesign
design <- svydesign(ids = ~1, weights = get.w(W.out),
                  data = ncp_final)

# Estimate model
tmod2 <- svyglm(conservative ~ treat + interest + as.factor(gender) +
                as.factor(region)+ as.factor(edu) + as.factor(yob),
                design = design)

# 95% CI
summ(tmod2, confint = TRUE)
```

Observations	7665
Dependent variable	conservative
Type	Survey-weighted linear regression

R ²	0.04
Adj. R ²	-1.20

Compute propensity score weights, INCLUDING survey weights provided (`weights`). Then, fit a WLS with

	Est.	2.5%	97.5%	t val.	p
(Intercept)	4.16	3.83	4.48	25.15	0.00
treat	0.10	0.00	0.20	1.96	0.05
interest	0.14	0.08	0.21	4.34	0.00
as.factor(gender)2	-0.28	-0.38	-0.18	-5.46	0.00
as.factor(region)2	-0.22	-0.36	-0.08	-3.07	0.00
as.factor(region)3	-0.24	-0.47	-0.00	-1.97	0.05
as.factor(region)4	-0.20	-0.34	-0.06	-2.77	0.01
as.factor(region)5	-0.66	-0.85	-0.46	-6.62	0.00
as.factor(region)6	-0.64	-0.82	-0.45	-6.70	0.00
as.factor(edu)2	0.24	0.05	0.43	2.53	0.01
as.factor(edu)3	0.28	0.10	0.46	3.00	0.00
as.factor(yob)2	0.10	-0.17	0.37	0.71	0.48
as.factor(yob)3	-0.20	-0.48	0.07	-1.48	0.14
as.factor(yob)4	-0.07	-0.35	0.20	-0.54	0.59
as.factor(yob)5	-0.09	-0.37	0.20	-0.59	0.56
as.factor(yob)6	-0.40	-0.69	-0.11	-2.72	0.01
as.factor(yob)7	-0.52	-0.85	-0.19	-3.09	0.00

Standard errors: Robust

weights computed and correct their standard errors with the sandwich estimator.

To include our survey weights, we specify them with the `s.weights` parameter in `weightit()`. Then, we multiply the resulting weights with the survey weights. When fitting the WLS, we use this product of weights, then we will correct the standard errors with the sandwich estimator.

```
library(lmtest)
library(sandwich)
# include survey weight variants:

ncp_clean <- ncp_final %>% dplyr::select(treat, interest,
                                         gender, region, edu, yob,
                                         conservative, weights)

# remove NAs, otherwise there will be NAs in the weights
ncp_clean <- na.omit(ncp_clean)

# Calculate weights with survey weights
W.out <- weightit(treat ~ gender + region + edu + yob,
                 data = ncp_clean, estimand = "ATT", method = "ps",
                 s.weights="weights")

# Check balance
bal.tab(W.out, m.threshold = .05, disp.v.ratio = TRUE, un=TRUE)

## Balance Measures
##           Type Diff.Un V.Ratio.Un Diff.Adj      M.Threshold V.Ratio.Adj
## prop.score Distance  0.1120    1.0048  0.0009 Balanced, <0.05    1.0168
## gender_2      Binary -0.0158      .    0.0004 Balanced, <0.05      .
## region        Contin.  0.0284    0.9995 -0.0001 Balanced, <0.05    0.9887
## edu           Contin.  0.0627    0.9342 -0.0002 Balanced, <0.05    0.9779
## yob           Contin. -0.0796    1.0120 -0.0009 Balanced, <0.05    1.0355
##
```

```
## Balance tally for mean differences
##               count
## Balanced, <0.05      5
## Not Balanced, >0.05    0
##
## Variable with the greatest mean difference
## Variable Diff.Adj      M.Threshold
##      yob -0.0009 Balanced, <0.05
##
## Effective sample sizes
##           Control Treated
## Unadjusted 6196.42 1099.27
## Adjusted   6200.21 1099.27
```

Again, the covariates are all balanced now.

```
# Multiply the survey weights with the balance weights to create att.weights
att.weights <- W.out$weights * ncp_clean$weights #product of weights

# Estimate the model using lm and specifying weights
tmod3 <- lm(conservative ~ treat + interest + as.factor(gender) +
            as.factor(region)+ as.factor(edu) + as.factor(yob) ,
            data = ncp_clean, weights = att.weights)

#correct SE with the sandwich estimator:
corr_se <- sandwich::vcovHC(tmod3) %>% diag %>% sqrt
```

5. Report all models in the same table using stargazer (make sure to specify html document). What is your conclusion?

```
#Putting it all together:

stargazer(tmod1, tmod2, tmod3, type="html",
          covariate.labels = c("Primed",
                              "Political Interest",
                              "Female",
                              "Østlandet",
                              "Sørlandet",
                              "Vestlandet",
                              "Trøndelag",
                              "Nord-Norge",
                              "Completed Upper Secondary Education",
                              "Completed University/ College Education",
                              "Born between 1940-1949",
                              "Born between 1950-1959",
                              "Born between 1960-1969",
                              "Born between 1970-1979",
                              "Born between 1980-1989",
                              "Born in or after 1990",
                              "Intercept"),
          se=list(NULL, NULL, corr_se),
          column.labels = c("No Weights OLS", "PS Weights SWLS", "PS/NCP Weights SWLS"),
          star.cutoffs = c(.10,.05,.01,.001),
          star.char = c("+","*","**","***"),
          column.sep.width = "10pt",
          notes = c("p<.10+", "p<.05*", "p<.01**", "p<.001***"),
```

```
single.row = TRUE,  
notes.append=FALSE)
```

Dependent variable:

conservative

OLS

survey-weighted

OLS

normal

No Weights OLS

PS Weights SWLS

PS/NCP Weights SWLS

(1)

(2)

(3)

Primed

0.099+ (0.050)

0.099* (0.051)

0.098+ (0.052)

Political Interest

0.140*** (0.023)

0.143*** (0.033)

0.133*** (0.034)

Female

-0.222*** (0.037)

-0.281*** (0.052)

-0.283*** (0.053)

Østlandet

-0.233*** (0.051)

-0.223** (0.073)

-0.218** (0.074)

Sørlandet

-0.106 (0.089)

-0.236* (0.120)

-0.220+ (0.126)

Vestlandet

-0.123* (0.049)

-0.196** (0.071)

-0.202** (0.072)

Trøndelag

-0.597*** (0.070)

-0.656*** (0.099)

-0.648*** (0.099)

Nord-Norge

-0.597*** (0.072)

-0.636*** (0.095)

-0.631*** (0.097)

Completed Upper Secondary Education

0.284*** (0.064)

0.242* (0.096)

0.209* (0.098)

Completed University/ College Education

0.295*** (0.063)

0.277** (0.092)

0.252** (0.094)

Born between 1940-1949

0.062 (0.108)

0.099 (0.140)

0.145 (0.144)

Born between 1950-1959

-0.193+ (0.105)

-0.204 (0.138)

-0.171 (0.143)

Born between 1960-1969

-0.097 (0.106)

-0.075 (0.138)

-0.062 (0.143)

Born between 1970-1979

-0.184+ (0.107)

-0.085 (0.145)

-0.048 (0.149)

Born between 1980-1989

-0.462*** (0.110)

-0.403** (0.148)
 -0.382* (0.153)
 Born in or after 1990
 -0.503*** (0.118)
 -0.523** (0.169)
 -0.482** (0.172)
 Intercept
 4.112*** (0.124)
 4.159*** (0.165)
 4.180*** (0.170)
 Observations
 7,665
 7,665
 7,665
 R2
 0.035
 0.040
 Adjusted R2
 0.033
 0.038
 Log Likelihood
 -15,441.080
 Akaike Inf. Crit.
 30,916.150
 Residual Std. Error (df = 7648)
 1.578
 0.870
 F Statistic (df = 16; 7648)
 17.561***
 19.839***
 Note:
 p<.10+, p<.05, p<.01, **p<.001**

From the three models reported, we see that there is a small positive effect of being primed the cost of the existing municipal structure. Across three models, the effect is significant, p<.10+. These primed respondents are likely to increase their rating of the conservative party .1 point more on a 1-7 scale. The effect size is modest but it is statistically significant in all three models, regardless of the weighting method used.