

The Lindner Example

Wyatt P. Bensken and Harry Persaud

Version: 2021-02-25

Contents

1	Load data	2
2	Data management	3
2.1	Managing binary variables	3
2.2	Inspecting the clean data	4
3	Codebook	5
4	Table 1	6
5	Task 1: Ignoring covariates, estimate the effect of treatment vs. control on the two outcomes	7
5.1	Quantitative outcome: <code>cardbill</code>	7
5.2	Binary outcome: <code>sixMonthSurvive</code>	8
6	Task 2: Fitting the propensity score model	9
6.1	Comparing distribution of propensity scores across treatment groups	9
6.2	Numerically	9
6.3	Visually	10
7	Task 3: Rubin's Rules For Assessing Overlap Before Propensity Adjustment	11
7.1	Rubin's Rule 1	11
7.2	Rubin's Rule 2	11
8	Task 4: Greedy 1:1 matching on the linear PS	12
8.1	Love Plot of standardized differences before and after 1:1 matching	16
8.2	Using <code>ggplot</code>	16
8.3	Using <code>cobalt</code> to make the Love Plot	17
8.4	Extracting Variance Ratios	18
8.5	Creating a dataframe containing the matched sample	19
8.6	Reassessing Rubin's Rules after 1:1 matching without replacement	19
9	Task 5: Estimating the causal effect of the treatment on both outcomes after 1:1 matching without replacement	19
9.1	The Quantitative Outcome	19
9.2	The Binary Outcome	21
10	Task 6 1:1 Matching With replacement	22
10.1	Love Plot of standardized differences before and after 1:1 matching	26
10.2	Using <code>ggplot</code>	26
10.3	Using <code>cobalt</code> to make the Love Plot	28

10.4	Extracting Variance Ratios	29
10.5	Creating a dataframe containing the matched sample	30
10.6	Reassessing Rubin’s Rules after 1:1 matching with replacement	31
10.7	Estimating the causal effect of the treatment on both outcomes after 1:1 matching with replacement	32
11	Task 7: Subclassification by Propensity Score Quintile	34
11.1	Check Balance and Propensity Score Overlap in Each Quintile	34
11.2	Creating a Standardized Difference Calculation Function	35
11.3	Plotting the post-subclassification standardized differences	36
11.4	Rubin’s Rules post subclassification	37
12	Task 8: Estimated effect after subclassification	38
12.1	Quantitative outcome	38
12.2	Binary Outcome	39
13	Task 9: Weighting	41
13.1	Calculating the ATT and ATE weights	41
13.2	Working with the ATT weights	41
13.3	Working with the ATE weights	45
14	Task 10: Using TWANG for propensity score estimation and ATT weighting	48
14.1	Estimated effect on outcomes after TWANG ATT weighting	52
15	Task 11: After direct adjustment with linear PS	53
15.1	Quantitative outcome	53
15.2	Binary outcome	53
16	Task 12: “Double Robust” Approach: Weighting + Direct Adjustment	54
16.1	Quantitative outcome	54
16.2	Binary outcome	55
16.3	Session Information	56

```
# Load packages
library(broom)
library(patchwork)
library(cobalt)
library(Matching)
library(tableone)
library(twang)
library(janitor)
library(here)
library(magrittr)
library(lme4)
library(tidyverse)
```

Note: we will also use the `broom.mixed` package, but we are not loading it as to prevent it conflicting with the functions of `broom`.

If you notice any errors or encounter any problems with this example, please contact Wyatt Bensken.

1 Load data

Information on the lindner dataset can be found at this site.^{1,2}

¹ Rdocumentation. (n.d.). lindner: Lindner Center Data On 996 PCI Patients Analyzed By Kereiakes Et Al. (2000). Retrieved from <https://www.rdocumentation.org/packages/MatchLinReg/versions/0.7.0/topics/lindner>

² Kereiakes DJ, Obenchain RL, Barber BL, et al. Abciximab provides cost effective survival advantage in high volume interventional practice. Am Heart J 2000; 140: 603-610.

```
lindner_raw <- read.csv("data/lindner.csv")
```

```
lindner_raw %>%  
  head(10)
```

	lifepres	cardbill	abcix	stent	height	female	diabetic	acutemi	ejecfrac
1	0.0	14301	1	0	163	1	1	0	56
2	11.6	3563	1	0	168	0	0	0	56
3	11.6	4694	1	0	188	0	0	0	50
4	11.6	7366	1	0	175	0	1	0	50
5	11.6	8247	1	0	168	1	0	0	55
6	11.6	8319	1	0	178	0	0	0	50
7	11.6	8410	1	0	185	0	0	0	58
8	11.6	8517	1	0	173	1	0	0	30
9	11.6	8763	1	0	152	1	0	0	60
10	11.6	8823	1	0	180	0	0	0	60

	ves1proc	sixMonthSurvive
1	1	FALSE
2	1	TRUE
3	1	TRUE
4	1	TRUE
5	1	TRUE
6	1	TRUE
7	1	TRUE
8	1	TRUE
9	1	TRUE
10	1	TRUE

```
colSums(is.na(lindner_raw))
```

lifepres	cardbill	abcix	stent	height
0	0	0	0	0
female	diabetic	acutemi	ejecfrac	ves1proc
0	0	0	0	0
sixMonthSurvive				
0				

After reading in the data, we can print the first 10 rows to get a sense of what our data looks like. We see it contains information on 996 participants, and there is no missing data.

2 Data managment

2.1 Managing binary variables

In the course of this example, we'll want both a numeric and factored version of each binary variable.

- In all numeric versions of binary variables: 1 indicates 'yes' to having trait/characteristic, 0 indicates 'no' to having trait/characteristic.

- Variable names with trailing "_f" denotes the factored version of each binary variable.

```
# Six month survival (turning logical variable to a factor)
lindner_raw$sixMonthSurvive_f <- factor(lindner_raw$sixMonthSurvive, levels = c(TRUE,FALSE),
                                       labels = c("yes", "no"))

# Creating numeric (1/0) version of six month survival variable
lindner_raw$sixMonthSurvive <- factor(lindner_raw$sixMonthSurvive_f, levels = c("yes","no"),
                                       labels = c(1, 0))

lindner_raw$sixMonthSurvive <- ifelse(lindner_raw$sixMonthSurvive == "1", 1, 0)

#Add variable named treated (same values as abcix variable)
lindner_raw$treated <- lindner_raw$abcix

# Factoring the exposure of interest variable. Change the name to 'treated' too.
lindner_raw$treated_f <- factor(lindner_raw$abcix, levels = c(1,0),
                               labels = c("treated", "control"))

# Factor version of stent variable
lindner_raw$stent_f <- factor(lindner_raw$stent, levels = c(1,0),
                             labels = c("yes", "no"))

# Factoring the female variable
lindner_raw$female_f <- factor(lindner_raw$female, levels = c(1,0),
                              labels = c("female", "male"))

# Factoring the diabetic variable
lindner_raw$diabetic_f <- factor(lindner_raw$diabetic, levels = c(1,0),
                                 labels = c("yes", "no"))

# Factoring the acutemi variable
lindner_raw$acutemi_f <- factor(lindner_raw$acutemi, levels = c(1,0),
                               labels = c("yes", "no"))

# Add patientid

lindner_raw <- tibble::rowid_to_column(lindner_raw, "patientid")

# Make lindner dataset with "clean" name.
lindner_clean <- lindner_raw
```

2.2 Inspecting the clean data

```
mosaic::inspect(lindner_clean)
```

Registered S3 method overwritten by 'mosaic':

```
  method      from
  fortify.SpatialPolygonsDataFrame ggplot2
```

categorical variables:

	name	class	levels	n	missing
1	sixMonthSurvive_f	factor	2	996	0
2	treated_f	factor	2	996	0

```

3      stent_f factor      2 996      0
4      female_f factor    2 996      0
5      diabetic_f factor  2 996      0
6      acutemi_f factor   2 996      0

```

distribution

```

1 yes (97.4%), no (2.6%)
2 treated (70.1%), control (29.9%)
3 yes (66.9%), no (33.1%)
4 male (65.3%), female (34.7%)
5 no (77.6%), yes (22.4%)
6 no (85.6%), yes (14.4%)

```

quantitative variables:

	name	class	min	Q1	median	Q3	max
...1	patientid	integer	1	249.75	498.5	747.25	996.0
...2	lifepres	numeric	0	11.60	11.6	11.60	11.6
...3	cardbill	integer	2216	10218.75	12458.0	16660.00	178534.0
...4	abcix	integer	0	0.00	1.0	1.00	1.0
...5	stent	integer	0	0.00	1.0	1.00	1.0
...6	height	integer	108	165.00	173.0	178.00	196.0
...7	female	integer	0	0.00	0.0	1.00	1.0
...8	diabetic	integer	0	0.00	0.0	0.00	1.0
...9	acutemi	integer	0	0.00	0.0	0.00	1.0
...10	ejecfrac	integer	0	45.00	55.0	56.00	90.0
...11	ves1proc	integer	0	1.00	1.0	2.00	5.0
...12	sixMonthSurvive	numeric	0	1.00	1.0	1.00	1.0
...13	treated	integer	0	0.00	1.0	1.00	1.0

	mean	sd	n	missing
...1	4.985000e+02	2.876647e+02	996	0
...2	1.129719e+01	1.850501e+00	996	0
...3	1.567416e+04	1.118226e+04	996	0
...4	7.008032e-01	4.581362e-01	996	0
...5	6.686747e-01	4.709262e-01	996	0
...6	1.714438e+02	1.065813e+01	996	0
...7	3.473896e-01	4.763800e-01	996	0
...8	2.238956e-01	4.170623e-01	996	0
...9	1.435743e-01	3.508337e-01	996	0
...10	5.096687e+01	1.041326e+01	996	0
...11	1.385542e+00	6.573525e-01	996	0
...12	9.738956e-01	1.595259e-01	996	0
...13	7.008032e-01	4.581362e-01	996	0

3 Codebook

Information was copy/pasted from here ^{1,2} (with some changes to reflect this analysis)

- **cardbill (Quantitative Outcome):** “Cardiac related costs incurred within 6 months of patient’s initial PCI; numeric value in 1998 dollars; costs were truncated by death for the 26 patients with lifepres == 0.”
- **sixMonthSurvive/sixMonthSurvive_f (Binary Outcome):** “Survival at six months a recoded version of lifepres.”
- **treated/treated_f (Exposure):** “Numeric treatment selection indicator; 0 implies usual PCI care

alone; 1 implies usual PCI care deliberately augmented by either planned or rescue treatment with abciximab.”

- **stent/stent_f**: “Coronary stent deployment; numeric, with 1 meaning YES and 0 meaning NO.”
- **height**: “Height in centimeters; numeric integer from 108 to 196.”
- **female/female_f**: “Female gender; numeric, with 1 meaning YES and 0 meaning NO.”
- **diabetic/diabetic_f**: “Diabetes mellitus diagnosis; numeric, with 1 meaning YES and 0 meaning NO.”
- **acutemi/acutemi_f**: “Acute myocardial infarction within the previous 7 days; numeric, with 1 meaning YES and 0 meaning NO.”
- **ejecfrac**: “Left ejection fraction; numeric value from 0 percent to 90 percent.”
- **ves1proc**: “Number of vessels involved in the patient’s initial PCI procedure; numeric integer from 0 to 5.”
- **patientid**: a made-up order of numbers to assign patient IDs for later use
- Note: Percutaneous Coronary Intervention (PCI)

¹ Rdocumentation. (n.d.). lindner: Lindner Center Data On 996 PCI Patients Analyzed By Kereiakes Et Al. (2000). Retrieved from <https://www.rdocumentation.org/packages/MatchLinReg/versions/0.7.0/topics/lindner>

² Kereiakes DJ, Obenchain RL, Barber BL, et al. Abciximab provides cost effective survival advantage in high volume interventional practice. Am Heart J 2000; 140: 603-610.

4 Table 1

```
var_list = c("cardbill", "sixMonthSurvive_f", "stent_f", "height", "female_f", "diabetic_f",
             "acutemi_f", "ejecfrac", "ves1proc")

factor_list = c("sixMonthSurvive_f", "stent_f", "female_f", "diabetic_f", "acutemi_f")

CreateTableOne(vars = var_list, strata = "treated_f",
               data = lindner_clean, factorVars = factor_list)
```

	Stratified by treated_f		p	test
	treated 698	control 298		
n				
cardbill (mean (SD))	16126.68 (9383.83)	14614.22 (14514.00)	0.051	
sixMonthSurvive_f = no (%)	11 (1.6)	15 (5.0)	0.004	
stent_f = no (%)	206 (29.5)	124 (41.6)	<0.001	
height (mean (SD))	171.44 (10.69)	171.45 (10.59)	0.996	
female_f = male (%)	467 (66.9)	183 (61.4)	0.111	
diabetic_f = no (%)	555 (79.5)	218 (73.2)	0.034	
acutemi_f = no (%)	573 (82.1)	280 (94.0)	<0.001	
ejecfrac (mean (SD))	50.40 (10.42)	52.29 (10.30)	0.009	
ves1proc (mean (SD))	1.46 (0.71)	1.20 (0.48)	<0.001	

As we can see, The mean **cardbill** was higher in the treated population and a larger percentage of controls did not survive through 6 months.

5 Task 1: Ignoring covariates, estimate the effect of treatment vs. control on the two outcomes

5.1 Quantitative outcome: cardbill

```
lindner_clean %>%  
  mosaic::favstats(cardbill ~ treated_f)
```

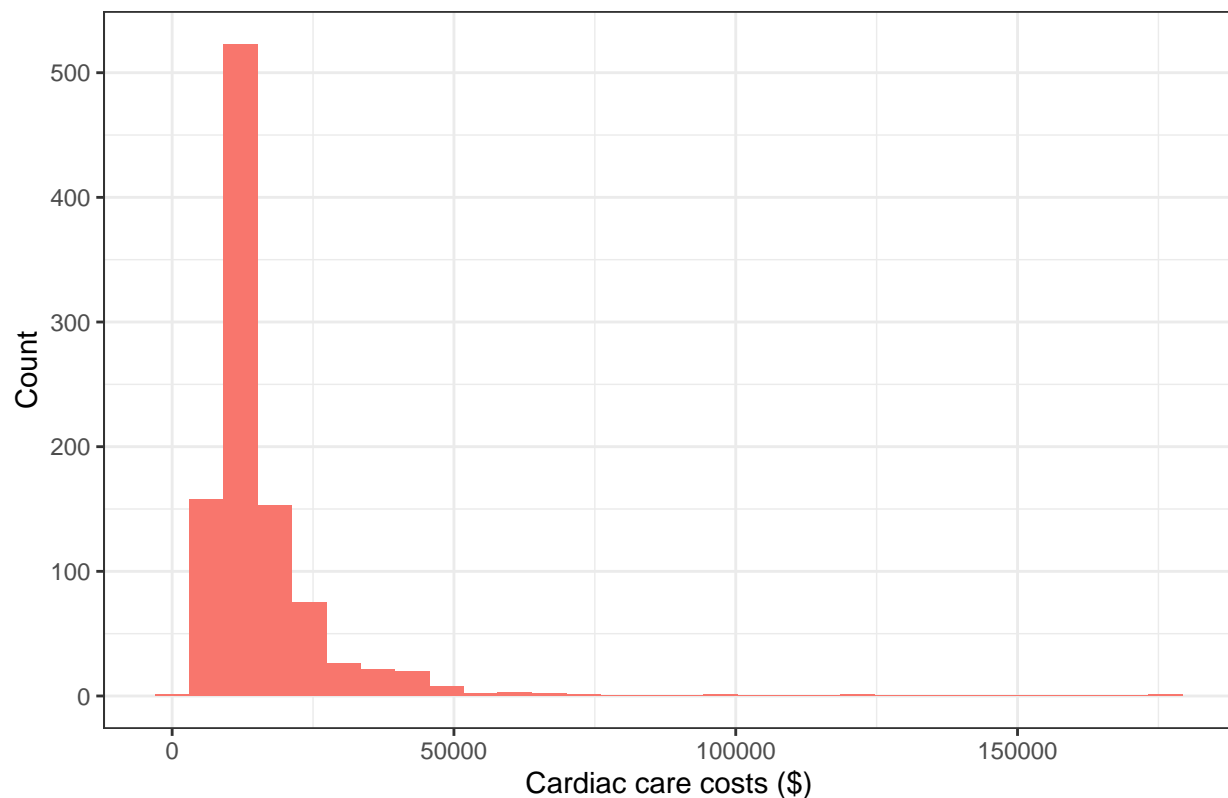
	treated_f	min	Q1	median	Q3	max	mean	sd	n	missing
1	treated	3563	10902.25	12944	17080.75	96741	16126.68	9383.825	698	0
2	control	2216	8300.00	10423	15895.75	178534	14614.22	14513.996	298	0

- Across the entire sample, the mean (\$16,127 vs. \$14,614) and median (\$12,944 vs. \$10,423) cardiac care costs were higher in treated individuals than non-treated controls.

```
ggplot(lindner_clean, aes(x = cardbill, fill = "cardbill")) +  
  geom_histogram() +  
  theme_bw() +  
  labs(y = "Count",  
       x = "Cardiac care costs ($)",  
       title = "Cardbill appears to be right skewed") +  
  guides(fill = FALSE)
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

Cardbill appears to be right skewed



As we can see in this figure, cardbill appears to be right/positively skewed.

```
unadjust_quant_outcome <- lm(cardbill ~ treated, data = lindner_clean)

unadjust_quant_outcome_tidy <- tidy(unadjust_quant_outcome, conf.int = TRUE, conf.level = 0.95) %>%
  filter(term == "treated")

unadjust_quant_outcome_tidy
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>     <dbl>     <dbl>   <dbl>   <dbl>     <dbl>
1 treated    1512.       773.       1.96  0.0506   -3.83    3029.
```

Treated individuals were estimated to spend 1512.46 (95%CI: -3.83, 3028.76) more dollars than non-treated controls

5.2 Binary outcome: sixMonthSurvive

```
Epi::twoby2(table(lindner_clean$treated_f, lindner_clean$sixMonthSurvive_f))
```

2 by 2 table analysis:

Outcome : yes

Comparing : treated vs. control

	yes	no	P(yes)	95% conf. interval
treated	687	11	0.9842	0.9718 0.9913
control	283	15	0.9497	0.9182 0.9694

	95% conf. interval
Relative Risk: 1.0364	1.0080 1.0656
Sample Odds Ratio: 3.3103	1.5020 7.2957
Conditional MLE Odds Ratio: 3.3057	1.3992 8.0624
Probability difference: 0.0346	0.0115 0.0664

Exact P-value: 0.0037

Asymptotic P-value: 0.0030

The odds treated individuals were alive after 6 months was roughly 3.31 times the odds that non-treated individuals were alive after 6 months.

```
unadjust_binary_outcome <- glm(sixMonthSurvive ~ treated, data = lindner_clean, family = binomial())

unadjust_binary_outcome_tidy <- tidy(unadjust_binary_outcome, conf.int = TRUE, conf.level = 0.95, expon
  filter(term == "treated")

unadjust_binary_outcome_tidy
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>     <dbl>     <dbl>   <dbl>   <dbl>     <dbl>
1 treated    3.31      0.403      2.97 0.00299   1.51     7.48
```

The odds of being alive after six months in treated individuals was 3.31 (95%CI: 1.51, 7.48) times higher than the odds that a non-treated control would be alive after six months.

6 Task 2: Fitting the propensity score model

We will now fit the propensity score, which predicts treatment status based on available covariates. Remember, we're not worried about overfitting (including too many covariates) when calculating the propensity scores.

```
psmodel <- glm(treated ~ stent + height + female + diabetic + acutemi + ejecfrac + veslproc, family = b  
summary(psmodel)
```

Call:

```
glm(formula = treated ~ stent + height + female + diabetic +  
    acutemi + ejecfrac + veslproc, family = binomial(), data = lindner_clean)
```

Deviance Residuals:

	Min	1Q	Median	3Q	Max
	-2.5211	-1.2109	0.6399	0.8827	1.5259

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	2.965651	1.731085	1.713	0.08668	.
stent	0.573018	0.150454	3.809	0.00014	***
height	-0.015366	0.009534	-1.612	0.10700	
female	-0.359060	0.206904	-1.735	0.08267	.
diabetic	-0.406810	0.170623	-2.384	0.01711	*
acutemi	1.199548	0.270468	4.435	9.20e-06	***
ejecfrac	-0.014789	0.007403	-1.998	0.04574	*
veslproc	0.760502	0.138437	5.493	3.94e-08	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1215.5 on 995 degrees of freedom
Residual deviance: 1124.3 on 988 degrees of freedom
AIC: 1140.3

Number of Fisher Scoring iterations: 4

Store the raw and linear propensity scores below.

```
lindner_clean$ps <- psmodel$fitted  
lindner_clean$linps <- psmodel$linear.predictors
```

6.1 Comparing distribution of propensity scores across treatment groups

6.2 Numerically

```
lindner_clean %>%  
  mosaic::favstats(ps ~ treated_f)
```

	treated_f	min	Q1	median	Q3	max	mean
1	treated	0.3121753	0.6402644	0.7158289	0.8259514	0.9800181	0.7265015
2	control	0.2323431	0.5558665	0.6462761	0.7093624	0.9583296	0.6406106

sd n missing

```
1 0.1299570 698      0
2 0.1230138 298      0
```

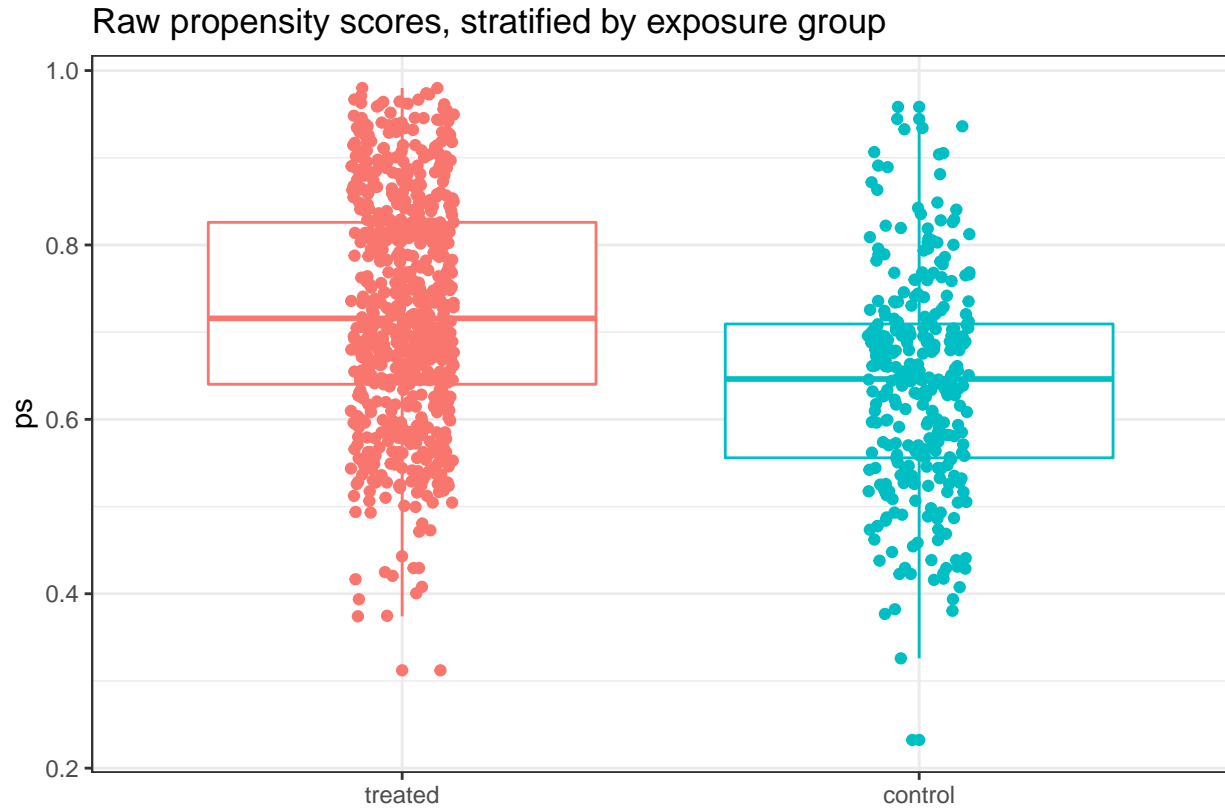
We can see there are no propensity scores equal to, or very close to, 0 or 1.

6.3 Visually

6.3.1 Boxplot

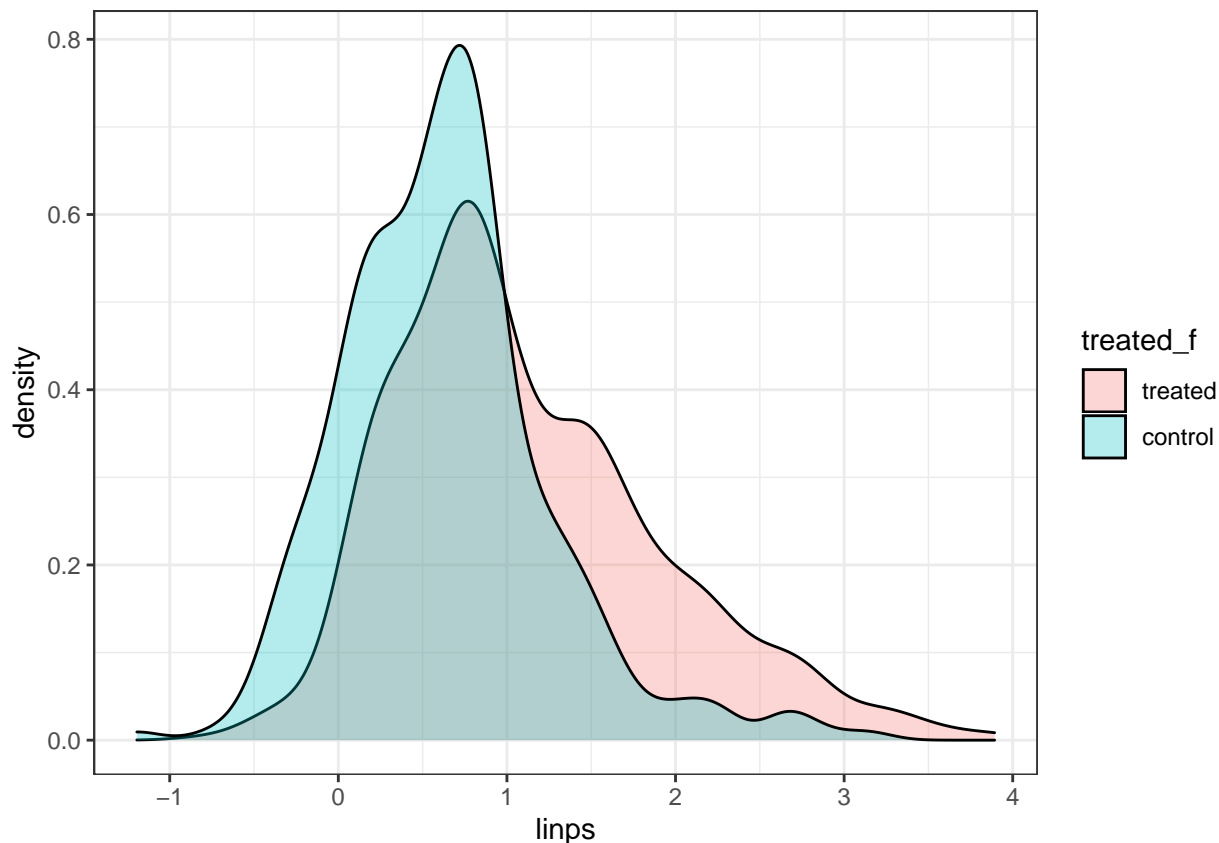
Now we'll visualize the distribution of the propensity scores stratified by treatment status.

```
ggplot(lindner_clean, aes(x = treated_f, y = ps, color = treated_f)) +
  geom_boxplot() +
  geom_jitter(width = 0.1) +
  guides(color = FALSE) +
  theme_bw() +
  labs(x = "",
       title = "Raw propensity scores, stratified by exposure group")
```



6.3.2 Density plot

```
ggplot(lindner_clean, aes(x = linps, fill = treated_f)) +
  geom_density(alpha = 0.3) +
  theme_bw()
```



Both of these plots demonstrate good overlap, suggesting a propensity score analysis may be appropriate.

7 Task 3: Rubin's Rules For Assessing Overlap Before Propensity Adjustment

7.1 Rubin's Rule 1

```
rubin1.unadj <- with(lindner_clean,
  abs(100*(mean(linps[treated==1])-mean(linps[treated==0]))/sd(linps)))
rubin1.unadj
```

```
[1] 61.86668
```

As you can see, we fail Rubin's Rule 1 - in which we want below 50%.

7.2 Rubin's Rule 2

```
rubin2.unadj <-with(lindner_clean, var(linps[treated==1])/var(linps[treated==0]))
rubin2.unadj
```

```
[1] 1.672048
```

We also "fail" Rubin's Rule 2 where we are looking for value between 0.8 - 1.2 (ideally, 1).

8 Task 4: Greedy 1:1 matching on the linear PS

The first type of match we will conduct is greedy 1:1 matching, without replacement. As we had only 298 controls, we will not match all of the 698 treated patients.

```
X <- lindner_clean$linps ## matching on the linear propensity score
Tr <- as.logical(lindner_clean$treated)
match1 <- Match(Tr=Tr, X=X, M = 1, replace=FALSE, ties=FALSE)
```

```
Warning in Match(Tr = Tr, X = X, M = 1, replace = FALSE, ties = FALSE):
replace==FALSE, but there are more (weighted) treated obs than control obs. Some
treated obs will not be matched. You may want to estimate ATC instead.
```

```
summary(match1)
```

```
Estimate... 0
SE..... 0
T-stat..... NaN
p.val..... NA
```

```
Original number of observations..... 996
Original number of treated obs..... 698
Matched number of observations..... 298
Matched number of observations (unweighted). 298
```

Below we'll assess the match balance from the 1:1 matching.

```
set.seed(2021)
mb1 <- MatchBalance(treated ~ stent + height + female + diabetic + acutemi + ejecfrac + ves1proc + ps +
match.out = match1, nboots=500)
```

```
***** (V1) stent *****
```

	Before Matching	After Matching
mean treatment.....	0.70487	0.60738
mean control.....	0.58389	0.58389
std mean diff.....	26.505	4.8022
mean raw eQQ diff.....	0.12081	0.02349
med raw eQQ diff.....	0	0
max raw eQQ diff.....	1	1
mean eCDF diff.....	0.060489	0.011745
med eCDF diff.....	0.060489	0.011745
max eCDF diff.....	0.12098	0.02349
var ratio (Tr/Co).....	0.85457	0.98151
T-test p-value.....	0.00032255	0.49878

```
***** (V2) height *****
```

	Before Matching	After Matching
mean treatment.....	171.44	171.77
mean control.....	171.45	171.45
std mean diff.....	-0.033804	3.1486
mean raw eQQ diff.....	0.56376	0.88591

med raw eQQ diff.....	0	0
max raw eQQ diff.....	20	36
mean eCDF diff.....	0.0078996	0.013639
med eCDF diff.....	0.0060095	0.010067
max eCDF diff.....	0.024971	0.053691
var ratio (Tr/Co).....	1.0201	0.93356
T-test p-value.....	0.99608	0.70576
KS Bootstrap p-value..	0.938	0.57
KS Naive p-value.....	0.99947	0.78362
KS Statistic.....	0.024971	0.053691

***** (V3) female *****

	Before Matching	After Matching
mean treatment.....	0.33095	0.37584
mean control.....	0.38591	0.38591
std mean diff.....	-11.672	-2.075
mean raw eQQ diff.....	0.053691	0.010067
med raw eQQ diff.....	0	0
max raw eQQ diff.....	1	1
mean eCDF diff.....	0.02748	0.0050336
med eCDF diff.....	0.02748	0.0050336
max eCDF diff.....	0.05496	0.010067
var ratio (Tr/Co).....	0.93253	0.98988
T-test p-value.....	0.10045	0.79641

***** (V4) diabetic *****

	Before Matching	After Matching
mean treatment.....	0.20487	0.25503
mean control.....	0.26846	0.26846
std mean diff.....	-15.743	-3.0743
mean raw eQQ diff.....	0.063758	0.013423
med raw eQQ diff.....	0	0
max raw eQQ diff.....	1	1
mean eCDF diff.....	0.031793	0.0067114
med eCDF diff.....	0.031793	0.0067114
max eCDF diff.....	0.063585	0.013423
var ratio (Tr/Co).....	0.82788	0.96743
T-test p-value.....	0.03402	0.69509

***** (V5) acutemi *****

	Before Matching	After Matching
mean treatment.....	0.17908	0.0033557
mean control.....	0.060403	0.060403

std mean diff.....	30.931	-98.478
mean raw eQQ diff.....	0.11745	0.057047
med raw eQQ diff.....	0	0
max raw eQQ diff.....	1	1
mean eCDF diff.....	0.05934	0.028523
med eCDF diff.....	0.05934	0.028523
max eCDF diff.....	0.11868	0.057047
var ratio (Tr/Co).....	2.5853	0.058929
T-test p-value.....	4.6617e-09	7.888e-05

***** (V6) ejecfrac *****

	Before Matching	After Matching
mean treatment.....	50.403	53.349
mean control.....	52.289	52.289
std mean diff.....	-18.102	13.166
mean raw eQQ diff.....	2.0503	1.8255
med raw eQQ diff.....	1	0
max raw eQQ diff.....	20	20
mean eCDF diff.....	0.035602	0.026577
med eCDF diff.....	0.011423	0.033557
max eCDF diff.....	0.11383	0.053691
var ratio (Tr/Co).....	1.0238	0.61178
T-test p-value.....	0.0085806	0.15654
KS Bootstrap p-value..	0.006	0.434
KS Naive p-value.....	0.0089219	0.78362
KS Statistic.....	0.11383	0.053691

***** (V7) ves1proc *****

	Before Matching	After Matching
mean treatment.....	1.4628	1.0403
mean control.....	1.2047	1.2047
std mean diff.....	36.545	-67.707
mean raw eQQ diff.....	0.2651	0.16443
med raw eQQ diff.....	0	0
max raw eQQ diff.....	1	2
mean eCDF diff.....	0.043323	0.032886
med eCDF diff.....	0.0090671	0.0067114
max eCDF diff.....	0.18842	0.13087
var ratio (Tr/Co).....	2.1614	0.25567
T-test p-value.....	4.21e-11	5.2489e-08
KS Bootstrap p-value..	< 2.22e-16	< 2.22e-16
KS Naive p-value.....	7.2635e-07	0.012144
KS Statistic.....	0.18842	0.13087

***** (V8) ps *****

	Before Matching	After Matching
mean treatment.....	0.7265	0.60662
mean control.....	0.64061	0.64061
std mean diff.....	66.092	-45.866
mean raw eQQ diff.....	0.085216	0.046911
med raw eQQ diff.....	0.081353	0.035726
max raw eQQ diff.....	0.12087	0.23215
mean eCDF diff.....	0.17141	0.10312
med eCDF diff.....	0.17768	0.083893
max eCDF diff.....	0.27599	0.23154
var ratio (Tr/Co).....	1.1161	0.36304
T-test p-value.....	< 2.22e-16	4.5286e-12
KS Bootstrap p-value..	< 2.22e-16	< 2.22e-16
KS Naive p-value.....	3.042e-14	2.3042e-07
KS Statistic.....	0.27599	0.23154

***** (V9) linps *****

	Before Matching	After Matching
mean treatment.....	1.1148	0.44175
mean control.....	0.63332	0.63332
std mean diff.....	60.484	-61.383
mean raw eQQ diff.....	0.4787	0.2442
med raw eQQ diff.....	0.35992	0.15424
max raw eQQ diff.....	1.0113	2.1601
mean eCDF diff.....	0.17141	0.10312
med eCDF diff.....	0.17768	0.083893
max eCDF diff.....	0.27599	0.23154
var ratio (Tr/Co).....	1.672	0.25702
T-test p-value.....	< 2.22e-16	6.4526e-13
KS Bootstrap p-value..	< 2.22e-16	< 2.22e-16
KS Naive p-value.....	3.042e-14	2.3042e-07
KS Statistic.....	0.27599	0.23154

Before Matching Minimum p.value: < 2.22e-16

Variable Name(s): ves1proc ps linps Number(s): 7 8 9

After Matching Minimum p.value: < 2.22e-16

Variable Name(s): ves1proc ps linps Number(s): 7 8 9

```
covnames <- c("stent", "height", "female", "diabetic", "acutemi", "ejecfrac", "ves1proc", "ps", "linps")
```

This is Dr. Love's code to extract the standardized differences.

```
pre.szd <- NULL; post.szd <- NULL
for(i in 1:length(covnames)) {
pre.szd[i] <- mb1$BeforeMatching[[i]]$sdiff.pooled
post.szd[i] <- mb1$AfterMatching[[i]]$sdiff.pooled
}
```

We can now print our table of standardized differences.

```
match_szd <- data.frame(covnames, pre.szd, post.szd, row.names=covnames)
print(match_szd, digits=3)
```

	covnames	pre.szd	post.szd
stent	stent	25.445	4.80
height	height	-0.034	3.15
female	female	-11.466	-2.08
diabetic	diabetic	-14.983	-3.07
acutemi	acutemi	37.145	-98.48
ejecfrac	ejecfrac	-18.208	13.17
veslproc	veslproc	42.734	-67.71
ps	ps	67.880	-45.87
linps	linps	67.664	-61.38

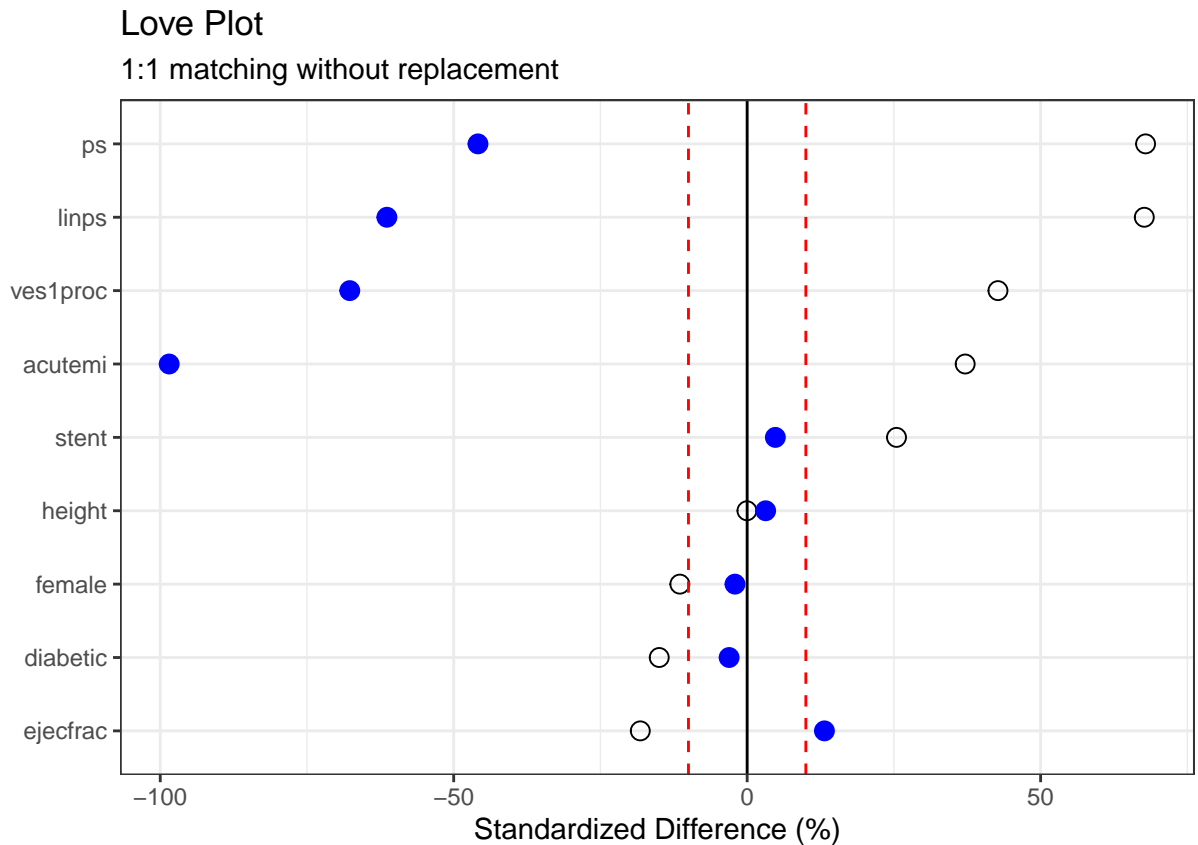
8.1 Love Plot of standardized differences before and after 1:1 matching

8.2 Using ggplot

In this figure, blue points are post-matching while white are pre-match

```
lp_wo_rep <- ggplot(match_szd, aes(x = pre.szd, y = reorder(covnames, pre.szd))) +
  geom_point(col = "black", size = 3, pch = 1) +
  geom_point(aes(x = post.szd, y = reorder(covnames, pre.szd)), size = 3, col = "blue") +
  theme_bw() +
  geom_vline(aes(xintercept = 0)) +
  geom_vline(aes(xintercept = 10), linetype = "dashed", col = "red") +
  geom_vline(aes(xintercept = -10), linetype = "dashed", col = "red") +
  labs(x = "Standardized Difference (%)",
       y = "",
       title = "Love Plot",
       subtitle = "1:1 matching without replacement")

lp_wo_rep
```

Just visually, we can see this match isn't all that great.

8.3 Using cobalt to make the Love Plot

There's a more automated way to build the Love Plot - as we see here.

```
cobalt_tab <- bal.tab(match1, treated ~ stent + height + female + diabetic + acutemi + ejecfrac + ves1proc)
cobalt_tab
```

Balance Measures

	Type	Diff.Un	Diff.Adj
stent	Binary	0.1210	0.0235
height	Contin.	-0.0003	0.0301
female	Binary	-0.0550	-0.0101
diabetic	Binary	-0.0636	-0.0134
acutemi	Binary	0.1187	-0.0570
ejecfrac	Contin.	-0.1810	0.1018
ves1proc	Contin.	0.3654	-0.2329
ps	Contin.	0.6609	-0.2616
linps	Contin.	0.6048	-0.2407

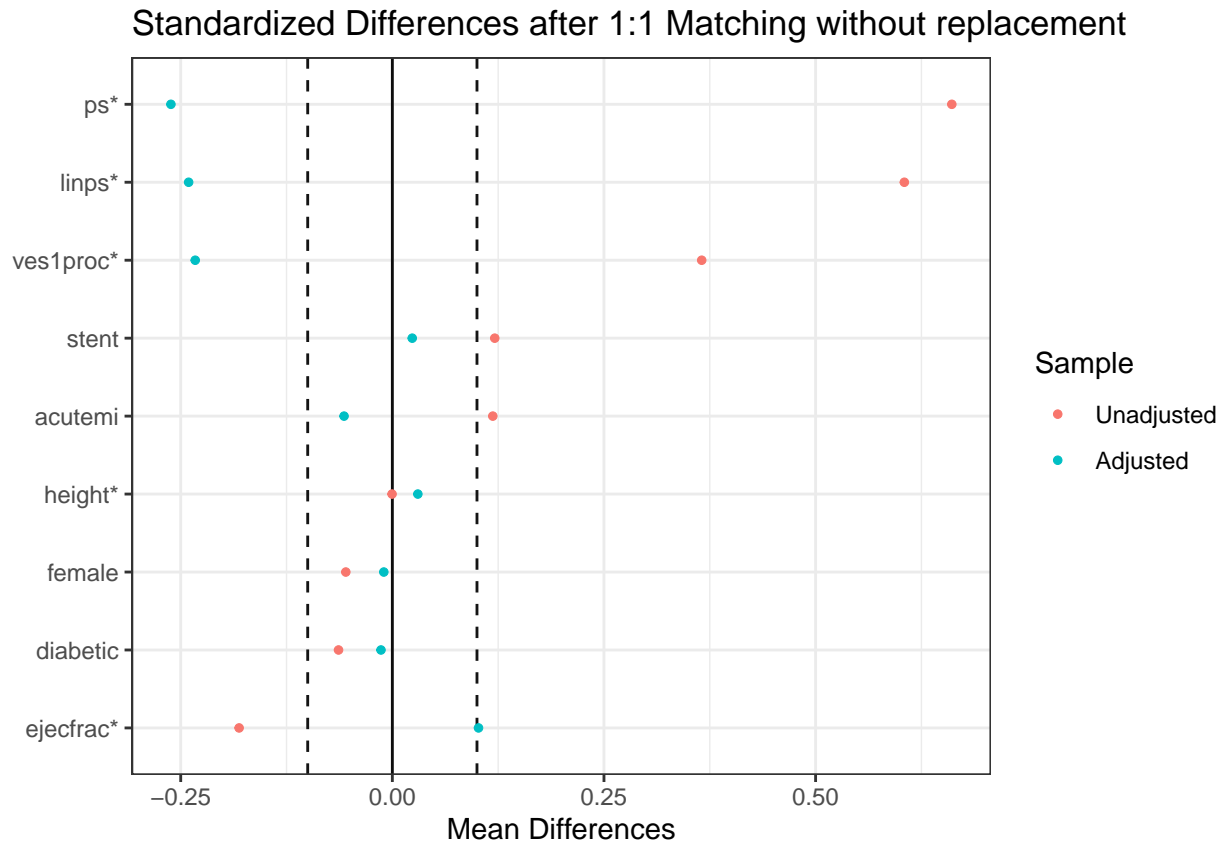
Sample sizes

	Control	Treated
All	298	698
Matched	298	298

```
Unmatched      0      400
```

```
p <- love.plot(cobalt_tab, threshold = .1, size = 1.5,
               var.order = "unadjusted",
               title = "Standardized Differences after 1:1 Matching without replacement",
               stars = "std")
```

```
p + theme_bw()
```



8.4 Extracting Variance Ratios

We can also look at variance ratios.

```
pre.vratio <- NULL; post.vratio <- NULL
for(i in 1:length(covnames)) {
  pre.vratio[i] <- mb1$BeforeMatching[[i]]$var.ratio
  post.vratio[i] <- mb1$AfterMatching[[i]]$var.ratio
}
## Table of Variance Ratios
match_vrat <- data.frame(names = covnames, pre.vratio, post.vratio, row.names=covnames)
print(match_vrat, digits=2)
```

	names	pre.vratio	post.vratio
stent	stent	0.85	0.982
height	height	1.02	0.934
female	female	0.93	0.990

diabetic	diabetic	0.83	0.967
acutemi	acutemi	2.59	0.059
ejecfrac	ejecfrac	1.02	0.612
veslproc	veslproc	2.16	0.256
ps	ps	1.12	0.363
linps	linps	1.67	0.257

8.5 Creating a dataframe containing the matched sample

We will create a dataframe which includes our matched sample, and do a quick count for a sanity check.

```
matches <- factor(rep(match1$index.treated, 2))
lindner_clean.matchedsample <- cbind(matches, lindner_clean[c(match1$index.control, match1$index.treated)])
lindner_clean.matchedsample %>% count(treated_f)
```

```
  treated_f    n
1   treated 298
2   control 298
```

8.6 Reassessing Rubin's Rules after 1:1 matching without replacement

8.6.1 Rubin's Rule 1

```
rubin1.match <- with(lindner_clean.matchedsample,
  abs(100*(mean(linps[treated==1])-mean(linps[treated==0]))/sd(linps)))
rubin1.match
```

```
[1] 38.54801
```

The new value for Rubin's Rule 1 is 38.55. While not ideal this technically passes Rubin's Rule 1 and is an improvement from the pre-match value of 61.87.

8.6.2 Rubin's Rule 2

```
rubin2.match <- with(lindner_clean.matchedsample, var(linps[treated==1])/var(linps[treated==0]))
rubin2.match
```

```
[1] 0.2570156
```

The new value for Rubin's Rule 2 is 0.26. This does not pass Rubin's Rule 2 and is not an improvement from the pre-match value of 1.67.

9 Task 5: Estimating the causal effect of the treatment on both outcomes after 1:1 matching without replacement

9.1 The Quantitative Outcome

We'll use a mixed model to estimate the effect of the treatment on `cardbill`. The matches will be treated as a random effect in the model (syntax "(1| matches.f)"), and the treatment group will be treated as a fixed effect. We will use restricted maximum likelihood (REML) to estimate coefficient values.

```
#to appease lme4, factor the matches
lindner_clean.matchedsample$matches.f <- as.factor(lindner_clean.matchedsample$matches)
```

```
# fit the mixed model
```

```
matched_mixedmodel.out1 <- lmer(cardbill ~ treated + (1 | matches.f), REML = TRUE, data=lindner_clean.m
```

```
summary(matched_mixedmodel.out1)
```

Linear mixed model fit by REML ['lmerMod']

Formula: cardbill ~ treated + (1 | matches.f)

Data: lindner_clean.matchedsample

REML criterion at convergence: 12815.1

Scaled residuals:

Min	1Q	Median	3Q	Max
-1.0478	-0.4272	-0.2546	0.0840	13.7428

Random effects:

Groups	Name	Variance	Std.Dev.
matches.f	(Intercept)	6986028	2643
	Residual	127700825	11300

Number of obs: 596, groups: matches.f, 298

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	14614.2	672.3	21.738
treated	-385.5	925.8	-0.416

Correlation of Fixed Effects:

	(Intr)
treated	-0.689

```
confint(matched_mixedmodel.out1)
```

Computing profile confidence intervals ...

	2.5 %	97.5 %
.sig01	0.000	4758.834
.sigma	10432.993	12196.988
(Intercept)	13296.648	15931.795
treated	-2202.772	1431.799

```
tidy_mixed_matched <- broom.mixed::tidy(matched_mixedmodel.out1, conf.int = TRUE, conf.level = 0.95) %>%
  filter(term == "treated")
```

```
tidy_mixed_matched
```

```
# A tibble: 1 x 8
```

effect	group	term	estimate	std.error	statistic	conf.low	conf.high
<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 fixed	<NA>	treated	-385.	926.	-0.416	-2200.	1429.

Treated individuals were estimated to spend \$-385.49 (95%CI: -2199.96, 1428.99) less than non-treated individuals. As this result is not significant at an α of 0.05, a sensitivity analysis on the quantitative outcome will not make sense.

```
#check the mean cardbill in the matched sample
lindner_clean.matchedsample %>% group_by(treated_f) %>% summarise(mean = mean(cardbill))
```

```
# A tibble: 2 x 2
  treated_f    mean
* <fct>      <dbl>
1 treated    14229.
2 control    14614.
```

```
#check the mean cardbill in the entire sample
lindner_clean %>% group_by(treated_f) %>% summarise(mean = mean(cardbill))
```

```
# A tibble: 2 x 2
  treated_f    mean
* <fct>      <dbl>
1 treated    16127.
2 control    14614.
```

In treated individuals, the mean `cardbill` was lower within the matched sample than the entire sample (note the mean within the control group was the same as every control participant is in the matched sample. The mean changed in the treated group as only 298/698 treated patients are in the matched sample). This is a sanity check to assess if the mixed model results make sense; and it looks like they do.

9.2 The Binary Outcome

We will use conditional logistic regression to estimate the log odds (and ORs) of being alive after 6 months based on treatment status.

```
binary_outcome_adjusted <- survival::clogit(sixMonthSurvive ~ treated + strata(matches), data=lindner_c
summary(binary_outcome_adjusted)
```

Call:

```
coxph(formula = Surv(rep(1, 596L), sixMonthSurvive) ~ treated +
      strata(matches), data = lindner_clean.matchedsample, method = "exact")
```

```
n= 596, number of events= 578
```

```
      coef exp(coef) se(coef)      z Pr(>|z|)
treated 1.6094     5.0000  0.6325 2.545  0.0109 *
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
      exp(coef) exp(-coef) lower .95 upper .95
treated         5         0.2    1.448    17.27
```

```
Concordance= 0.833 (se = 0.124 )
Likelihood ratio test= 8.73 on 1 df,  p=0.003
Wald test               = 6.48 on 1 df,  p=0.01
Score (logrank) test = 8 on 1 df,  p=0.005
```

```
#Tidy model
```

```
tidy_binary_outcome_adjusted <- tidy(binary_outcome_adjusted, exponentiate = TRUE, conf.int = 0.95)
```

The odds of being alive after six months were 5 times higher in treated individuals than non-treated individuals (95%CI 1.45, 17.27)

10 Task 6 1:1 Matching With replacement

- As we saw in the 1:1 matching without replacement, 400 treated participants were excluded from the sample. This is a waste of data and we'll address this by again matching 1 treated participant to 1 control participant. However, this time we'll match with replacement, meaning each time a control participant is matched to a treated participant, the control participant will be placed back into the pool of possible patients a treated individual can be matched to. Thus, some control participants will be matched multiple times (not all control participants have to be matched to a treated participant). In the Lindner dataset 1:1 matching with replacement is a more reasonable choice.

```
X <- lindner_clean$linps ## matching on the linear propensity score
Tr <- as.logical(lindner_clean$treated)
match1 <- Match(Tr=Tr, X=X, M = 1, replace=TRUE, ties=FALSE) # notice replace = TRUE
summary(match1)
```

```
Estimate... 0
SE..... 0
T-stat..... NaN
p.val..... NA
```

```
Original number of observations..... 996
Original number of treated obs..... 698
Matched number of observations..... 698
Matched number of observations (unweighted). 698
```

As you can see, this time we matched 698 treated individuals with 698 control participants. To reiterate, as we matched with replacement, and there were less control participants than treated participants, some control participants were matched multiple times.

Below we'll assess the match balance from the 1:1 matching with replacement.

```
set.seed(202102)
mb1 <- MatchBalance(treated ~ stent + height + female + diabetic + acutemi + ejecfrac + ves1proc + ps +
match.out = match1, nboots=500)
```

***** (V1) stent *****

	Before Matching	After Matching
mean treatment.....	0.70487	0.70487
mean control.....	0.58389	0.72779
std mean diff.....	26.505	-5.0222
mean raw eQQ diff.....	0.12081	0.022923
med raw eQQ diff.....	0	0
max raw eQQ diff.....	1	1
mean eCDF diff.....	0.060489	0.011461
med eCDF diff.....	0.060489	0.011461
max eCDF diff.....	0.12098	0.022923
var ratio (Tr/Co).....	0.85457	1.0501
T-test p-value.....	0.00032255	0.23555

***** (V2) height *****

	Before Matching	After Matching
mean treatment.....	171.44	171.44

mean control.....	171.45	171.62
std mean diff.....	-0.033804	-1.6209
mean raw eQQ diff.....	0.56376	0.83811
med raw eQQ diff.....	0	0
max raw eQQ diff.....	20	22
mean eCDF diff.....	0.0078996	0.010261
med eCDF diff.....	0.0060095	0.008596
max eCDF diff.....	0.024971	0.038682
var ratio (Tr/Co).....	1.0201	0.80936
T-test p-value.....	0.99608	0.7661
KS Bootstrap p-value..	0.968	0.396
KS Naive p-value.....	0.99947	0.67329
KS Statistic.....	0.024971	0.038682

***** (V3) female *****

	Before Matching	After Matching
mean treatment.....	0.33095	0.33095
mean control.....	0.38591	0.29943
std mean diff.....	-11.672	6.6934
mean raw eQQ diff.....	0.053691	0.031519
med raw eQQ diff.....	0	0
max raw eQQ diff.....	1	1
mean eCDF diff.....	0.02748	0.015759
med eCDF diff.....	0.02748	0.015759
max eCDF diff.....	0.05496	0.031519
var ratio (Tr/Co).....	0.93253	1.0555
T-test p-value.....	0.10045	0.1537

***** (V4) diabetic *****

	Before Matching	After Matching
mean treatment.....	0.20487	0.20487
mean control.....	0.26846	0.22923
std mean diff.....	-15.743	-6.0301
mean raw eQQ diff.....	0.063758	0.024355
med raw eQQ diff.....	0	0
max raw eQQ diff.....	1	1
mean eCDF diff.....	0.031793	0.012178
med eCDF diff.....	0.031793	0.012178
max eCDF diff.....	0.063585	0.024355
var ratio (Tr/Co).....	0.82788	0.92199
T-test p-value.....	0.03402	0.21126

***** (V5) acutemi *****

	Before Matching	After Matching
mean treatment.....	0.17908	0.17908
mean control.....	0.060403	0.16762
std mean diff.....	30.931	2.9871
mean raw eQQ diff.....	0.11745	0.011461
med raw eQQ diff.....	0	0
max raw eQQ diff.....	1	1
mean eCDF diff.....	0.05934	0.0057307
med eCDF diff.....	0.05934	0.0057307
max eCDF diff.....	0.11868	0.011461
var ratio (Tr/Co).....	2.5853	1.0537
T-test p-value.....	4.6617e-09	0.44149

***** (V6) ejecfrac *****

	Before Matching	After Matching
mean treatment.....	50.403	50.403
mean control.....	52.289	50.812
std mean diff.....	-18.102	-3.9327
mean raw eQQ diff.....	2.0503	0.80516
med raw eQQ diff.....	1	0
max raw eQQ diff.....	20	20
mean eCDF diff.....	0.035602	0.012247
med eCDF diff.....	0.011423	0.008596
max eCDF diff.....	0.11383	0.06447
var ratio (Tr/Co).....	1.0238	1.1088
T-test p-value.....	0.0085806	0.43271
KS Bootstrap p-value..	0.004	0.044
KS Naive p-value.....	0.0089219	0.1099
KS Statistic.....	0.11383	0.06447

***** (V7) ves1proc *****

	Before Matching	After Matching
mean treatment.....	1.4628	1.4628
mean control.....	1.2047	1.4642
std mean diff.....	36.545	-0.20289
mean raw eQQ diff.....	0.2651	0.044413
med raw eQQ diff.....	0	0
max raw eQQ diff.....	1	1
mean eCDF diff.....	0.043323	0.0074021
med eCDF diff.....	0.0090671	0.004298
max eCDF diff.....	0.18842	0.018625
var ratio (Tr/Co).....	2.1614	1.0942

T-test p-value.....	4.21e-11	0.95523
KS Bootstrap p-value..	< 2.22e-16	0.594
KS Naive p-value.....	7.2635e-07	0.99973
KS Statistic.....	0.18842	0.018625

***** (V8) ps *****

	Before Matching	After Matching
mean treatment.....	0.7265	0.7265
mean control.....	0.64061	0.7262
std mean diff.....	66.092	0.23256
mean raw eQQ diff.....	0.085216	0.0014016
med raw eQQ diff.....	0.081353	0.00063595
max raw eQQ diff.....	0.12087	0.021689
mean eCDF diff.....	0.17141	0.0031873
med eCDF diff.....	0.17768	0.0014327
max eCDF diff.....	0.27599	0.024355
var ratio (Tr/Co).....	1.1161	1.0083
T-test p-value.....	< 2.22e-16	0.0032848
KS Bootstrap p-value..	< 2.22e-16	0.978
KS Naive p-value.....	3.042e-14	0.98578
KS Statistic.....	0.27599	0.024355

***** (V9) linps *****

	Before Matching	After Matching
mean treatment.....	1.1148	1.1148
mean control.....	0.63332	1.108
std mean diff.....	60.484	0.859
mean raw eQQ diff.....	0.4787	0.016276
med raw eQQ diff.....	0.35992	0.0028864
max raw eQQ diff.....	1.0113	0.75735
mean eCDF diff.....	0.17141	0.0031873
med eCDF diff.....	0.17768	0.0014327
max eCDF diff.....	0.27599	0.024355
var ratio (Tr/Co).....	1.672	1.0466
T-test p-value.....	< 2.22e-16	0.0016161
KS Bootstrap p-value..	< 2.22e-16	0.978
KS Naive p-value.....	3.042e-14	0.98578
KS Statistic.....	0.27599	0.024355

Before Matching Minimum p.value: < 2.22e-16
Variable Name(s): ves1proc ps linps Number(s): 7 8 9

After Matching Minimum p.value: 0.0016161
Variable Name(s): linps Number(s): 9

```
covnames <- c("stent", "height", "female", "diabetic", "acutemi", "ejecfrac", "veslproc", "ps", "linps")
```

Dr. Love's code to extract the standardized differences.

```
pre.szd <- NULL; post.szd <- NULL
for(i in 1:length(covnames)) {
  pre.szd[i] <- mb1$BeforeMatching[[i]]$sdiff.pooled
  post.szd[i] <- mb1$AfterMatching[[i]]$sdiff.pooled
}
```

Table of standardized differences

```
match_szd <- data.frame(covnames, pre.szd, post.szd, row.names=covnames)
print(match_szd, digits=3)
```

	covnames	pre.szd	post.szd
stent	stent	25.445	-5.022
height	height	-0.034	-1.621
female	female	-11.466	6.693
diabetic	diabetic	-14.983	-6.030
acutemi	acutemi	37.145	2.987
ejecfrac	ejecfrac	-18.208	-3.933
veslproc	veslproc	42.734	-0.203
ps	ps	67.880	0.233
linps	linps	67.664	0.859

10.1 Love Plot of standardized differences before and after 1:1 matching

10.2 Using ggplot

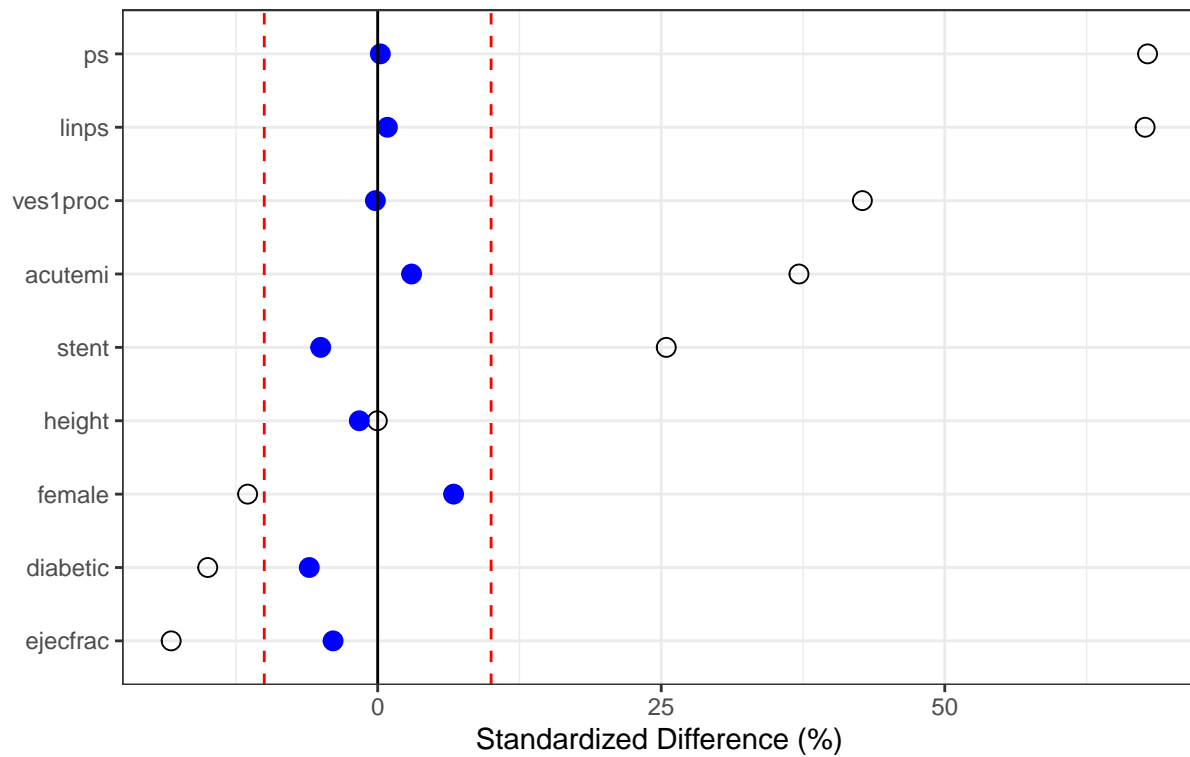
In this figure, blue points are post-matching while white are pre-match.

```
lp_w_rep <- ggplot(match_szd, aes(x = pre.szd, y = reorder(covnames, pre.szd))) +
  geom_point(col = "black", size = 3, pch = 1) +
  geom_point(aes(x = post.szd, y = reorder(covnames, pre.szd)), size = 3, col = "blue") +
  theme_bw() +
  geom_vline(aes(xintercept = 0)) +
  geom_vline(aes(xintercept = 10), linetype = "dashed", col = "red") +
  geom_vline(aes(xintercept = -10), linetype = "dashed", col = "red") +
  labs(x = "Standardized Difference (%)",
       y = "",
       title = "Love Plot",
       subtitle = "1:1 matching with replacement")

lp_w_rep
```

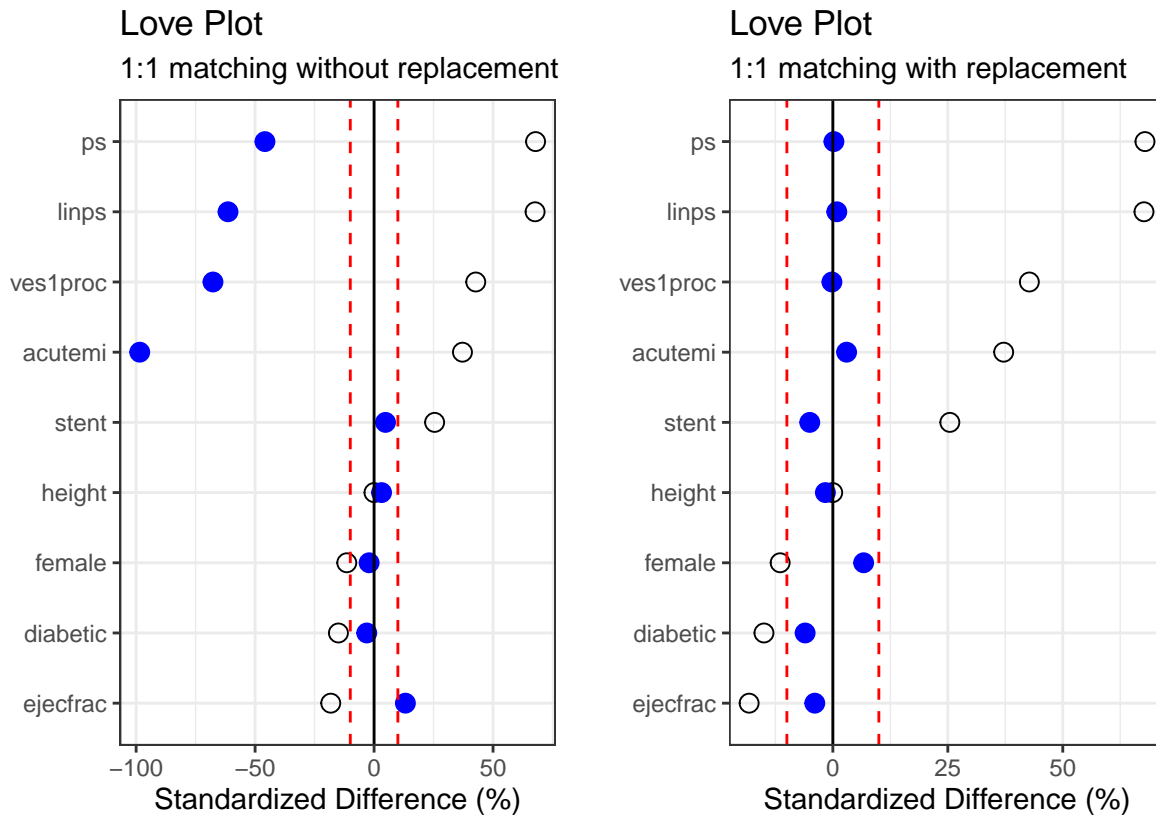
Love Plot

1:1 matching with replacement



- Visually, the Love Plot using 1:1 matching with replacement looks pretty good.

```
# comparison of love plots with and without replacement  
lp_wo_rep + lp_w_rep
```



When we look at the plots without replacement and with replacement side-by-side, it definitely looks better than the 1:1 matching without replacement.

10.3 Using cobalt to make the Love Plot

Again, we can also use an automated way to make the Love Plot.

```
cobalt_tab <- bal.tab(match1, treated ~ stent + height + female + diabetic + acutemi + ejecfrac + ves1proc,
  stats = c("m", "v"))
```

```
cobalt_tab
```

Balance Measures

	Type	Diff.Un	V.Ratio.Un	Diff.Adj	V.Ratio.Adj
stent	Binary	0.1210	.	-0.0229	.
height	Contin.	-0.0003	1.0201	-0.0162	0.8033
female	Binary	-0.0550	.	0.0315	.
diabetic	Binary	-0.0636	.	-0.0244	.
acutemi	Binary	0.1187	.	0.0115	.
ejecfrac	Contin.	-0.1810	1.0238	-0.0393	1.1004
ves1proc	Contin.	0.3654	2.1614	-0.0020	1.0860
ps	Contin.	0.6609	1.1161	0.0023	1.0007
linps	Contin.	0.6048	1.6720	0.0086	1.0387

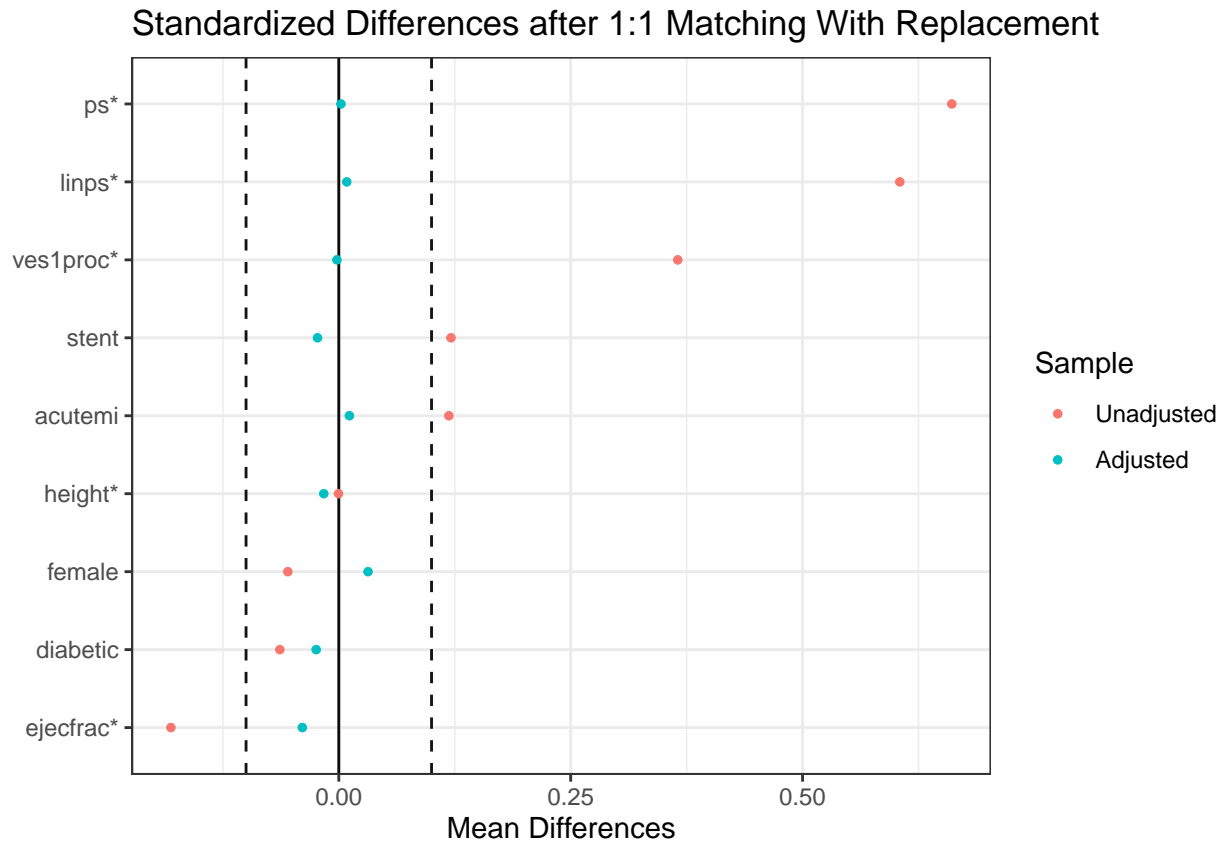
Sample sizes

Control Treated

All	298.	698
Matched (ESS)	112.16	698
Matched (Unweighted)	229.	698
Unmatched	69.	0

```
p <- love.plot(cobalt_tab, threshold = .1, size = 1.5,
               var.order = "unadjusted",
               title = "Standardized Differences after 1:1 Matching With Replacement",
               stars = "std")
```

```
p + theme_bw()
```



10.4 Extracting Variance Ratios

```
pre.vratio <- NULL; post.vratio <- NULL
for(i in 1:length(covnames)) {
  pre.vratio[i] <- mb1$BeforeMatching[[i]]$var.ratio
  post.vratio[i] <- mb1$AfterMatching[[i]]$var.ratio
}
## Table of Variance Ratios
match_vrat <- data.frame(names = covnames, pre.vratio, post.vratio, row.names=covnames)
print(match_vrat, digits=2)
```

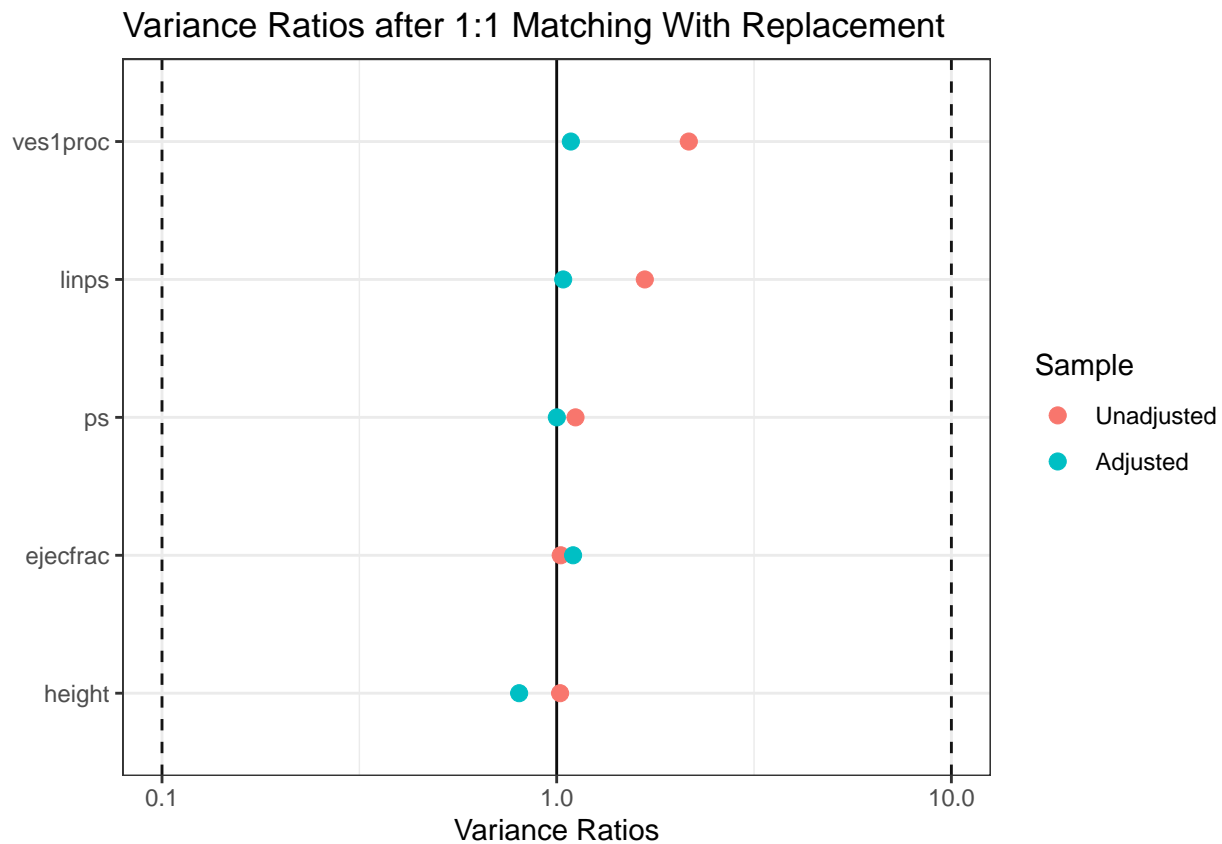
	names	pre.vratio	post.vratio
stent	stent	0.85	1.05

height	height	1.02	0.81
female	female	0.93	1.06
diabetic	diabetic	0.83	0.92
acutemi	acutemi	2.59	1.05
ejecfrac	ejecfrac	1.02	1.11
ves1proc	ves1proc	2.16	1.09
ps	ps	1.12	1.01
linps	linps	1.67	1.05

10.4.1 Using 'cobalt' to make a Love Plot of Variance Ratios

```
p <- love.plot(cobalt_tab, stats = "v", threshold = .1, size = 3,
               var.order = "unadjusted",
               title = "Variance Ratios after 1:1 Matching With Replacement",
               stars = "")

p + theme_bw()
```



10.5 Creating a dataframe containing the matched sample

```
matches <- factor(rep(match1$index.treated, 2))
lindner_clean.matchedsample <- cbind(matches, lindner_clean[c(match1$index.control, match1$index.treated)])
```

```
lindner_clean.matchedsample %>% count(treated_f)
```

```
  treated_f    n
1   treated 698
2   control 698
```

10.5.1 How many times were the non-treated patients matched?

```
lindner_clean.matchedsample %>% filter(treated_f == "control") %>%
  count(patientid) %>%
  janitor::tabyl(n)
```

```
  n n_n    percent
1  84 0.366812227
2  53 0.231441048
3  36 0.157205240
4  20 0.087336245
5   8 0.034934498
6   6 0.026200873
7   5 0.021834061
8   3 0.013100437
9   1 0.004366812
10  2 0.008733624
11  2 0.008733624
12  2 0.008733624
13  1 0.004366812
15  1 0.004366812
16  3 0.013100437
17  2 0.008733624
```

10.6 Reassessing Rubin's Rules after 1:1 matching with replacement

10.6.1 Rubin's Rule 1

```
rubin1.match.rep <- with(lindner_clean.matchedsample,
  abs(100*(mean(linps[treated==1])-mean(linps[treated==0]))/sd(linps)))
rubin1.match.rep
```

```
[1] 0.8690187
```

The new value for Rubin's Rule 1 is 0.87. This value passes Rubin's Rule 1 and is an improvement from the Rubin's Rule 1 value obtained during 1:1 matching without replacement, 38.55. The pre-match value was 61.87.

10.6.2 Rubin's Rule 2

```
rubin2.match.rep <- with(lindner_clean.matchedsample, var(linps[treated==1])/var(linps[treated==0]))
rubin2.match.rep
```

```
[1] 1.046553
```

The new value for Rubin's Rule 2 is 1.05. This passes Rule 2 and is an improvement from the Rubin's Rule 2 value obtained during 1:1 matching without replacement, 0.26. The pre-match value was 1.67.

10.7 Estimating the causal effect of the treatment on both outcomes after 1:1 matching with replacement

10.7.1 The Quantitative Outcome

Again, we'll use a mixed model to estimate the effect of the treatment on `cardbill`. The matches will be treated as a random effect in the model (syntax "(1| matches.f)". and the treatment group will be treated as a fixed effect. We will use restricted maximum likelihood (REML) to estimate coefficient values.

```
#to appease lme4, factor the matches
lindner_clean.matchedsample$matches.f <- as.factor(lindner_clean.matchedsample$matches)

# fit the mixed model
matched_mixedmodel.rep.out1 <- lmer(cardbill ~ treated + (1 | matches.f), REML = TRUE, data=lindner_clean.matchedsample)

summary(matched_mixedmodel.rep.out1)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: cardbill ~ treated + (1 | matches.f)
Data: lindner_clean.matchedsample
```

```
REML criterion at convergence: 30148
```

```
Scaled residuals:
```

Min	1Q	Median	3Q	Max
-1.1569	-0.4789	-0.2828	0.1116	13.2194

```
Random effects:
```

Groups	Name	Variance	Std.Dev.
matches.f	(Intercept)	7604218	2758
Residual		135785615	11653

```
Number of obs: 1396, groups: matches.f, 698
```

```
Fixed effects:
```

	Estimate	Std. Error	t value
(Intercept)	16337.4	453.2	36.046
treated	-210.7	623.8	-0.338

```
Correlation of Fixed Effects:
```

```
(Intr)
treated -0.688
```

```
confint(matched_mixedmodel.rep.out1)
```

```
Computing profile confidence intervals ...
```

	2.5 %	97.5 %
.sig01	0.000	4287.944
.sigma	11059.236	12282.671
(Intercept)	15449.068	17225.702
treated	-1434.047	1012.643


```
tidy_mixed_matched_rep <- broom.mixed::tidy(matched_mixedmodel.rep.out1, conf.int = TRUE, conf.level = 0.95)
  filter(term == "treated")
```

```
tidy_mixed_matched_rep
```

```
# A tibble: 1 x 8
```

```
  effect group term      estimate std.error statistic conf.low conf.high
  <chr>   <chr> <chr>      <dbl>     <dbl>     <dbl>    <dbl>    <dbl>
1 fixed  <NA> treated    -211.      624.     -0.338   -1433.    1012.
```

Treated individuals were estimated to spend \$-210.7 less (95%CI -1433.24, 1011.84) than non-treated individuals. This finding is not significant at an α of 0.05, thus, the sensitivity analysis on the Quantitative outcome will still not make sense.

```
#sanity check for model
```

```
lindner_clean.matchedsample %>% group_by(treated_f) %>% summarise(mean_card = mean(cardbill))
```

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

```
  treated_f mean_card
* <fct>      <dbl>
1 treated    16127.
2 control    16337.
```

- The mixed model above predicted treated individuals would spend roughly \$-210.7 less than control participants. After doing a quick check of the mean `cardbill` within the matched sample, the mixed model results make sense.

10.7.2 The Binary Outcome

We will use conditional logistic regression to estimate the log odds (and ORs) of being alive after 6 months based on treatment status.

```
binary_outcome_adjusted_rep <- survival::clogit(sixMonthSurvive ~ treated + strata(matches), data=lindner_clean.matchedsample)
```

```
summary(binary_outcome_adjusted_rep)
```

Call:

```
coxph(formula = Surv(rep(1, 1396L), sixMonthSurvive) ~ treated +
      strata(matches), data = lindner_clean.matchedsample, method = "exact")
```

```
n= 1396, number of events= 1321
```

```
      coef exp(coef) se(coef)      z Pr(>|z|)
treated 1.8405    6.3000   0.3404  5.407 6.41e-08 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
      exp(coef) exp(-coef) lower .95 upper .95
treated      6.3    0.1587    3.233    12.28
```

```
Concordance= 0.863 (se = 0.057 )
```

```
Likelihood ratio test= 42.88 on 1 df, p=6e-11
```

```
Wald test = 29.24 on 1 df, p=6e-08
```

```
Score (logrank) test = 38.48 on 1 df, p=6e-10
```

```
#Tidy model
```

```
tidy_binary_outcome_adjusted_rep <- tidy(binary_outcome_adjusted_rep, exponentiate = TRUE, conf.int = 0
```

```
tidy_binary_outcome_adjusted_rep
```

```
# A tibble: 1 x 7
```

	term	estimate	std.error	statistic	p.value	conf.low	conf.high
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	treated	6.30	0.340	5.41	0.0000000641	3.23	12.3

The odds of being alive after six months were 6.3 times higher in treated individuals than non-treated controls (95%CI 3.23, 12.28)

11 Task 7: Subclassification by Propensity Score Quintile

```
#cut into quintiles
```

```
lindner_clean$stratum <- Hmisc::cut2(lindner_clean$ps, g=5)
```

```
lindner_clean$quintile <- factor(lindner_clean$stratum, labels=1:5)
```

```
#Sanity check: check to make sure quintiles are evenish, numbers make sense, etc.
```

```
lindner_clean %>% count(stratum, quintile)
```

	stratum	quintile	n
1	[0.232,0.581)	1	200
2	[0.581,0.669)	2	199
3	[0.669,0.726)	3	200
4	[0.726,0.826)	4	199
5	[0.826,0.980]	5	198

11.1 Check Balance and Propensity Score Overlap in Each Quintile

11.1.1 Numerically

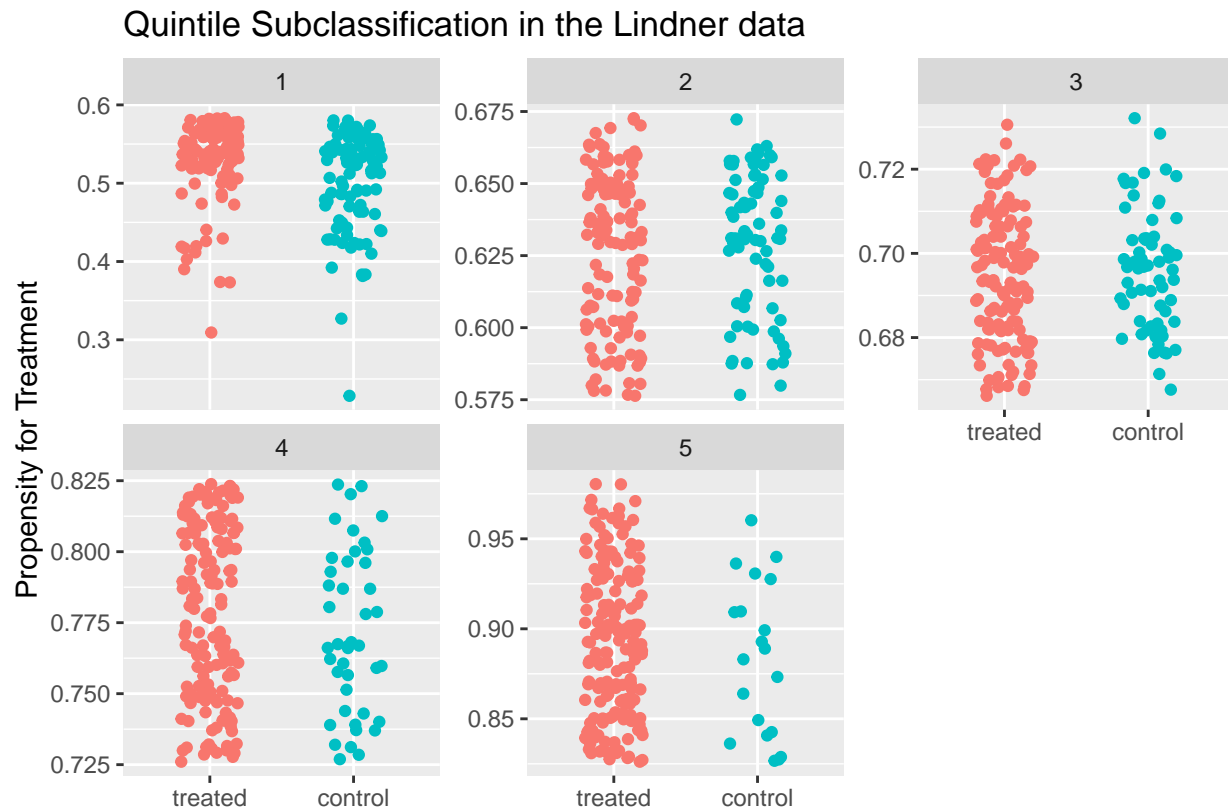
Only 20 controls were in the largest quintile, which seems a bit low.

```
lindner_clean %>% count(quintile, treated_f)
```

	quintile	treated_f	n
1	1	treated	105
2	1	control	95
3	2	treated	124
4	2	control	75
5	3	treated	135
6	3	control	65
7	4	treated	156
8	4	control	43
9	5	treated	178
10	5	control	20

11.1.2 Graphically

```
ggplot(lindner_clean, aes(x = treated_f, y = round(ps,2), group = quintile, color = treated_f)) +  
  geom_jitter(width = 0.2) +  
  guides(color = FALSE) +  
  facet_wrap(~ quintile, scales = "free_y") +  
  labs(x = "", y = "Propensity for Treatment",  
       title = "Quintile Subclassification in the Lindner data")
```



11.2 Creating a Standardized Difference Calculation Function

Here we implement Dr. Love's function to calculate the standardized differences is utilized below.

```
szd <- function(covlist, g) {  
  covlist2 <- as.matrix(covlist)  
  g <- as.factor(g)  
  res <- NA  
  for(i in 1:ncol(covlist2)) {  
    cov <- as.numeric(covlist2[,i])  
    num <- 100*diff(tapply(cov, g, mean, na.rm=TRUE))  
    den <- sqrt(mean(tapply(cov, g, var, na.rm=TRUE)))  
    res[i] <- round(num/den,2)  
  }  
  names(res) <- names(covlist)  
  res
```

```
}
```

Now we'll split data into quintiles - and give them each their own dataframe.

```
quin1 <- filter(lindner_clean, quintile==1)
quin2 <- filter(lindner_clean, quintile==2)
quin3 <- filter(lindner_clean, quintile==3)
quin4 <- filter(lindner_clean, quintile==4)
quin5 <- filter(lindner_clean, quintile==5)
```

Now we'll run the function above to calculate the standardized differences for each covariate in each quintile.

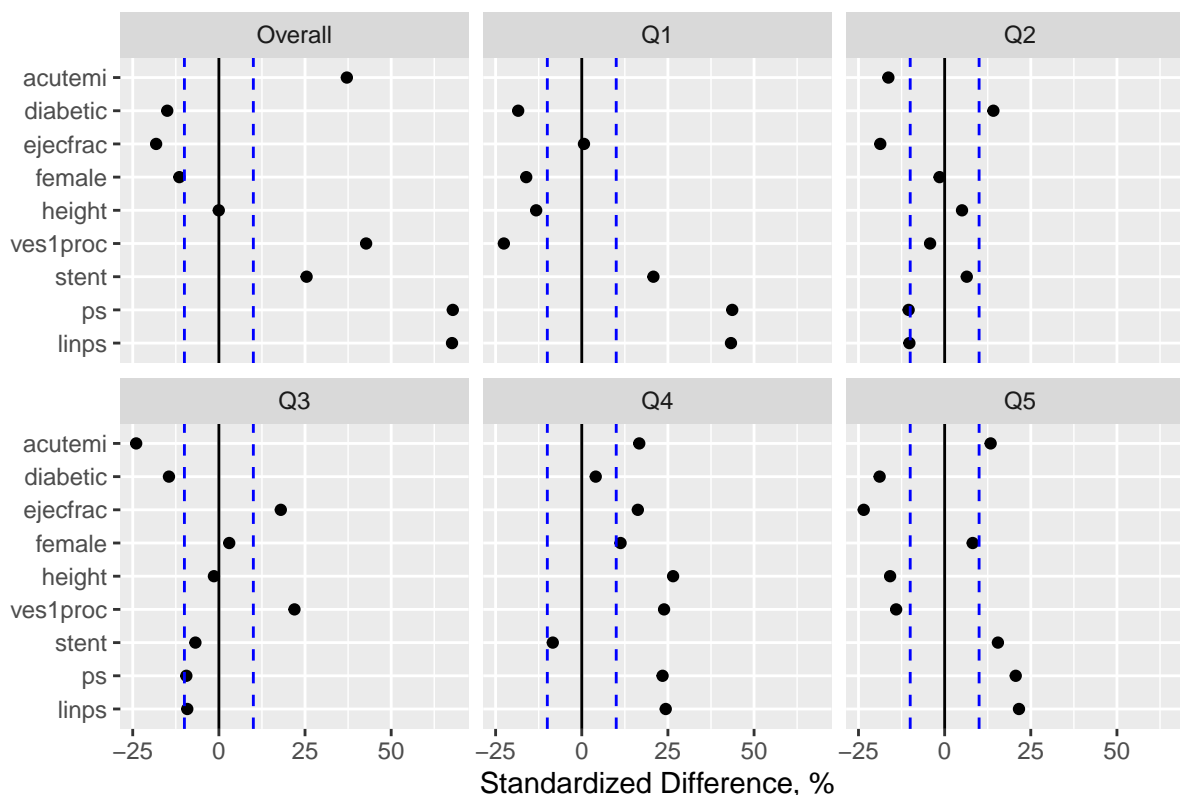
```
covs <- c("stent", "height", "female", "diabetic", "acutemi", "ejecfrac", "ves1proc", "ps", "linps")
d.q1 <- szd(quin1[covs], quin1$treated)
d.q2 <- szd(quin2[covs], quin2$treated)
d.q3 <- szd(quin3[covs], quin3$treated)
d.q4 <- szd(quin4[covs], quin4$treated)
d.q5 <- szd(quin5[covs], quin5$treated)
d.all <- szd(lindner_clean[covs], lindner_clean$treated)
lindner_clean.szd <- tibble(covs, Overall = d.all, Q1 = d.q1, Q2 = d.q2, Q3 = d.q3, Q4 = d.q4, Q5 = d.q5)
lindner_clean.szd <- gather(lindner_clean.szd, "quint", "sz.diff", 2:7)
```

11.3 Plotting the post-subclassification standardized differences

```
ggplot(lindner_clean.szd, aes(x = sz.diff, y = reorder(covs, -sz.diff), group = quint)) +
  geom_point() +
  geom_vline(xintercept = 0) +
  geom_vline(xintercept = c(-10,10), linetype = "dashed", col = "blue") +
  facet_wrap(~ quint) +
  labs(x = "Standardized Difference, %", y = "",
       title = "Comparing Standardized Differences by PS Quintile")
```

Warning: Removed 1 rows containing missing values (geom_point).

Comparing Standardized Differences by PS Quintile



The results of the standardized differences by quintile are fairly variable.

11.4 Rubin's Rules post subclassification

11.4.1 Rule 1

```
rubin1.q1 <- with(quin1, abs(100*(mean(linps[treated==1]) - mean(linps[treated==0]))/sd(linps)))
rubin1.q2 <- with(quin2, abs(100*(mean(linps[treated==1]) - mean(linps[treated==0]))/sd(linps)))
rubin1.q3 <- with(quin3, abs(100*(mean(linps[treated==1]) - mean(linps[treated==0]))/sd(linps)))
rubin1.q4 <- with(quin4, abs(100*(mean(linps[treated==1]) - mean(linps[treated==0]))/sd(linps)))
rubin1.q5 <- with(quin5, abs(100*(mean(linps[treated==1]) - mean(linps[treated==0]))/sd(linps)))

rubin1.sub <- c(rubin1.q1, rubin1.q2, rubin1.q3, rubin1.q4, rubin1.q5)
names(rubin1.sub)=c("Q1", "Q2", "Q3", "Q4", "Q5")

rubin1.sub
```

```
      Q1      Q2      Q3      Q4      Q5
42.633282 10.122973  9.054266 23.662028 20.717673
```

All are under 50. Not great, but OK. For comparison, the original Rubin's Rule 1 value was 61.87.

11.4.2 Rule 2

```
rubin2.q1 <- with(quin1, var(linps[treated==1])/var(linps[treated==0]))
rubin2.q2 <- with(quin2, var(linps[treated==1])/var(linps[treated==0]))
rubin2.q3 <- with(quin3, var(linps[treated==1])/var(linps[treated==0]))
rubin2.q4 <- with(quin4, var(linps[treated==1])/var(linps[treated==0]))
rubin2.q5 <- with(quin5, var(linps[treated==1])/var(linps[treated==0]))

rubin2.sub <- c(rubin2.q1, rubin2.q2, rubin2.q3, rubin2.q4, rubin2.q5)
names(rubin2.sub)=c("Q1", "Q2", "Q3", "Q4", "Q5")
rubin2.sub
```

```
      Q1      Q2      Q3      Q4      Q5
0.6582169 1.2083230 1.1754770 1.2154060 1.2353984
```

All but Q1 are at least close to passing Rule 2. For comparison, the original Rubin's Rule 2 value was 1.67.

12 Task 8: Estimated effect after subclassification

12.1 Quantitative outcome

```
quin1.out1 <- lm(cardbill ~ treated, data=quin1)
quin2.out1 <- lm(cardbill ~ treated, data=quin2)
quin3.out1 <- lm(cardbill ~ treated, data=quin3)
quin4.out1 <- lm(cardbill ~ treated, data=quin4)
quin5.out1 <- lm(cardbill ~ treated, data=quin5)
```

```
coef(summary(quin1.out1)); coef(summary(quin2.out1)); coef(summary(quin3.out1)); coef(summary(quin4.out1)); coef(summary(quin5.out1))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	14262.49474	1083.197	13.16704155	7.497113e-29
treated	-67.69474	1494.953	-0.04528217	9.639280e-01

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	15038.427	1794.884	8.378497	1.000329e-14
treated	1412.154	2273.799	0.621055	5.352814e-01

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	13259.415	1099.734	12.05693	1.846022e-25
treated	2837.814	1338.554	2.12006	3.524616e-02

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	14474.19	1620.396	8.932501	2.966193e-16
treated	2979.16	1830.144	1.627828	1.051596e-01

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	19398.350	1967.305	9.860368	7.011002e-19
treated	-3498.063	2074.886	-1.685906	9.340509e-02

The mean of the five quintile-specific estimated regression coefficients is below.

```
est.st <- (coef(quin1.out1)[2] + coef(quin2.out1)[2] + coef(quin3.out1)[2] +
coef(quin4.out1)[2] + coef(quin5.out1)[2])/5
```

```
est.st
```

```
treated
732.674
```

The mean SE is below.

```
se.q1 <- summary(quin1.out1)$coefficients[2,2]
se.q2 <- summary(quin2.out1)$coefficients[2,2]
se.q3 <- summary(quin3.out1)$coefficients[2,2]
se.q4 <- summary(quin4.out1)$coefficients[2,2]
se.q5 <- summary(quin5.out1)$coefficients[2,2]

se.st <- sqrt((se.q1^2 + se.q2^2 + se.q3^2 + se.q4^2 + se.q5^2)*(1/25))
se.st
```

```
[1] 821.008
```

The mean estimate, with a 95% CI, is below.

```
strat.result1 <- data_frame(estimate = est.st,
  conf.low = est.st - 1.96*se.st,
  conf.high = est.st + 1.96*se.st)
```

Warning: `data_frame()` was deprecated in tibble 1.1.0.
Please use `tibble()` instead.

```
strat.result1
```

```
# A tibble: 1 x 3
  estimate conf.low conf.high
    <dbl>    <dbl>    <dbl>
1    733.    -877.    2342.
```

So treated individuals were estimated to spend \$732.67 more (95%CI -876.5, 2341.85) than non treated individuals.

12.2 Binary Outcome

```
quin1.out2 <- glm(sixMonthSurvive ~ treated, data=quin1, family=binomial())
quin2.out2 <- glm(sixMonthSurvive ~ treated, data=quin2, family=binomial())
quin3.out2 <- glm(sixMonthSurvive ~ treated, data=quin3, family=binomial())
quin4.out2 <- glm(sixMonthSurvive ~ treated, data=quin4, family=binomial())
quin5.out2 <- glm(sixMonthSurvive ~ treated, data=quin5, family=binomial())
```

```
coef(summary(quin1.out2)); coef(summary(quin2.out2)); coef(summary(quin3.out2)); coef(summary(quin4.out2));
```

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.124565	0.5108708	6.116155	9.586018e-10
treated	1.519826	1.1272001	1.348319	1.775557e-01

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	2.876386	0.5138915	5.597262	2.177636e-08
treated	1.935799	1.1278865	1.716306	8.610597e-02

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	3.028522	0.5911534	5.123073	3.005960e-07
treated	1.869318	1.1648042	1.604834	1.085303e-01

	Estimate	Std. Error	z value	Pr(> z)
--	----------	------------	---------	----------

```
(Intercept) 3.737670 1.011815 3.6940239 0.0002207331
treated      0.194156 1.167726 0.1662684 0.8679457146
```

```
      Estimate Std. Error z value Pr(>|z|)
(Intercept) 1.734601 0.6262243 2.769936 0.005606735
treated      1.809253 0.7732630 2.339764 0.019295953
```

Estimated log-odds (averaged over the quintiles).

```
est.st.log <- (coef(quin1.out2)[2] + coef(quin2.out2)[2] + coef(quin3.out2)[2] +
coef(quin4.out2)[2] + coef(quin5.out2)[2])/5
```

```
est.st.log
```

```
treated
1.46567
```

Estimated odds ratio (averaged over the quintiles).

```
exp(est.st.log)
```

```
treated
4.330444
```

The average SE (averaged over the quintiles).

```
se.q1.log <- summary(quin1.out2)$coefficients[2,2]
se.q2.log <- summary(quin2.out2)$coefficients[2,2]
se.q3.log <- summary(quin3.out2)$coefficients[2,2]
se.q4.log <- summary(quin4.out2)$coefficients[2,2]
se.q5.log <- summary(quin5.out2)$coefficients[2,2]
```

```
se.st.log <- sqrt((se.q1.log^2 + se.q2.log^2 + se.q3.log^2 + se.q4.log^2 + se.q5.log^2)*(1/25))
se.st.log #log odds
```

```
[1] 0.4841899
```

```
strat.result2 <- data_frame(estimate = exp(est.st.log),
conf.low = exp(est.st.log - 1.96*se.st.log),
conf.high = exp(est.st.log + 1.96*se.st.log))
```

```
strat.result2
```

```
# A tibble: 1 x 3
  estimate conf.low conf.high
  <dbl>     <dbl>     <dbl>
1    4.33     1.68    11.2
```

The odds of being alive after 6 months was 4.33 (95%CI 1.68, 11.19) times higher in treated individuals than non-treated individuals.

13 Task 9: Weighting

13.1 Calculating the ATT and ATE weights

13.1.1 ATT weights

First, we can use the average treatment effect on the treated (ATT) approach where we weight treated subjects as 1 and controls as $ps/(1-ps)$

```
lindner_clean$wts1 <- ifelse(lindner_clean$treated==1, 1, lindner_clean$ps/(1-lindner_clean$ps))
```

13.1.2 ATE weights

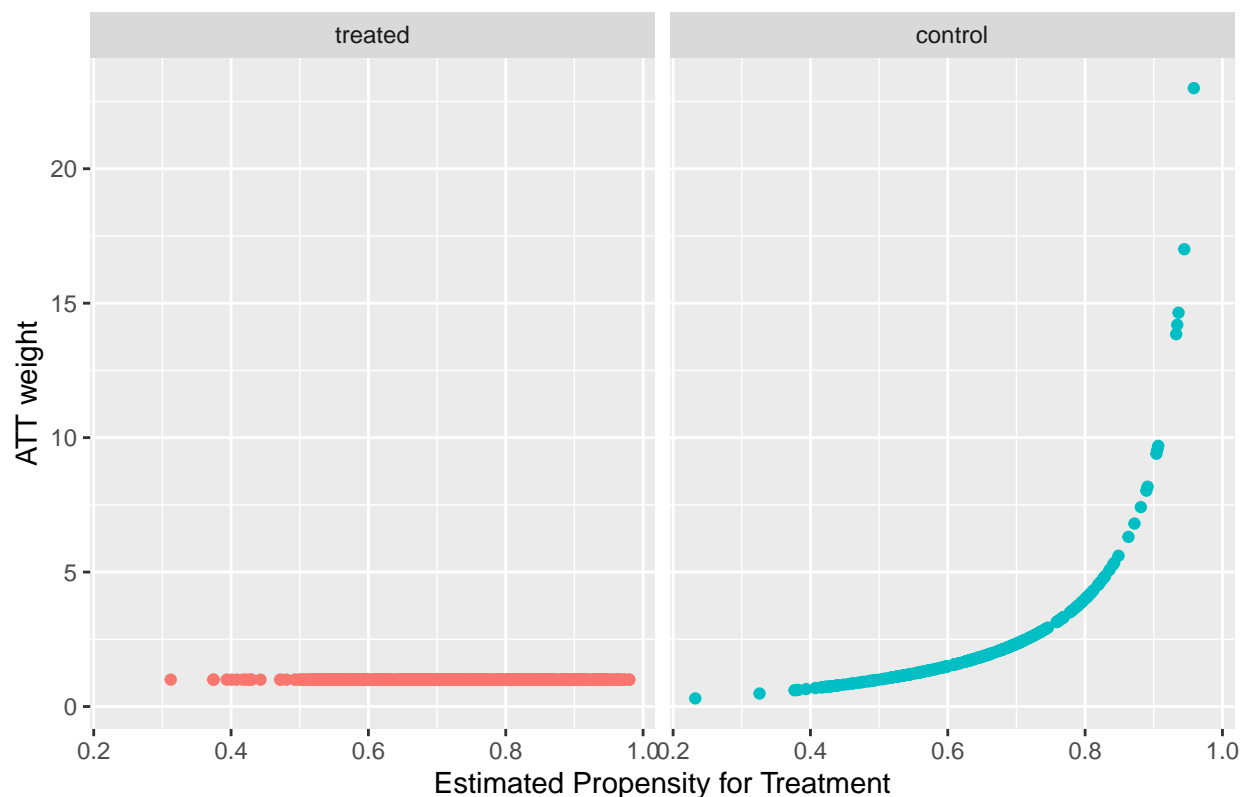
We can also use the average treatment effect (ATE) weights where we weight treated subjects by $1/ps$ and controls by $1/(1-PS)$

```
lindner_clean$wts2 <- ifelse(lindner_clean$treated==1, 1/lindner_clean$ps, 1/(1-lindner_clean$ps))
```

13.2 Working with the ATT weights

```
ggplot(lindner_clean, aes(x = ps, y = wts1, color = treated_f)) +  
  geom_point() +  
  guides(color = FALSE) +  
  facet_wrap(~ treated_f) +  
  labs(x = "Estimated Propensity for Treatment",  
       y = "ATT weight",  
       title = "ATT weighting structure")
```

ATT weighting structure



```
#turn dataset into a dataframe for twang (its a tibble now)
```

```
lindner_clean_df <- data.frame(lindner_clean)
```

```
#name covariates
```

```
covlist <- c("stent", "height", "female", "diabetic", "acutemi", "ejecfrac", "veslproc", "ps", "linps")
```

```
bal.wts1 <- dx.wts(x=lindner_clean_df$wts1, data=lindner_clean_df, vars=covlist,  
treat.var="treated", estimand="ATT")
```

```
bal.wts1
```

```
  type n.treat n.ctrl ess.treat ess.ctrl    max.es    mean.es    max.ks  
1  unw     698   298      698 298.0000 0.66091743 0.29567509 0.27599469  
2      698   298      698 149.4503 0.08471131 0.03315857 0.06089807  
  mean.ks iter  
1 0.13749095  NA  
2 0.03182485  NA
```

```
bal.table(bal.wts1)
```

```
$unw
```

	tx.mn	tx.sd	ct.mn	ct.sd	std.eff.sz	stat	p	ks	ks.pval
stent	0.705	0.456	0.584	0.494	0.265	3.624	0.000	0.121	0.004
height	171.443	10.695	171.446	10.589	0.000	-0.005	0.996	0.025	0.999
female	0.331	0.471	0.386	0.488	-0.117	-1.647	0.100	0.055	0.531
diabetic	0.205	0.404	0.268	0.444	-0.157	-2.127	0.034	0.064	0.349
acutemi	0.179	0.384	0.060	0.239	0.309	5.923	0.000	0.119	0.005
ejecfrac	50.403	10.419	52.289	10.297	-0.181	-2.640	0.008	0.114	0.008

veslproc	1.463	0.706	1.205	0.480	0.365	6.693	0.000	0.188	0.000
ps	0.727	0.130	0.641	0.123	0.661	9.928	0.000	0.276	0.000
linps	1.115	0.796	0.633	0.616	0.605	10.321	0.000	0.276	0.000

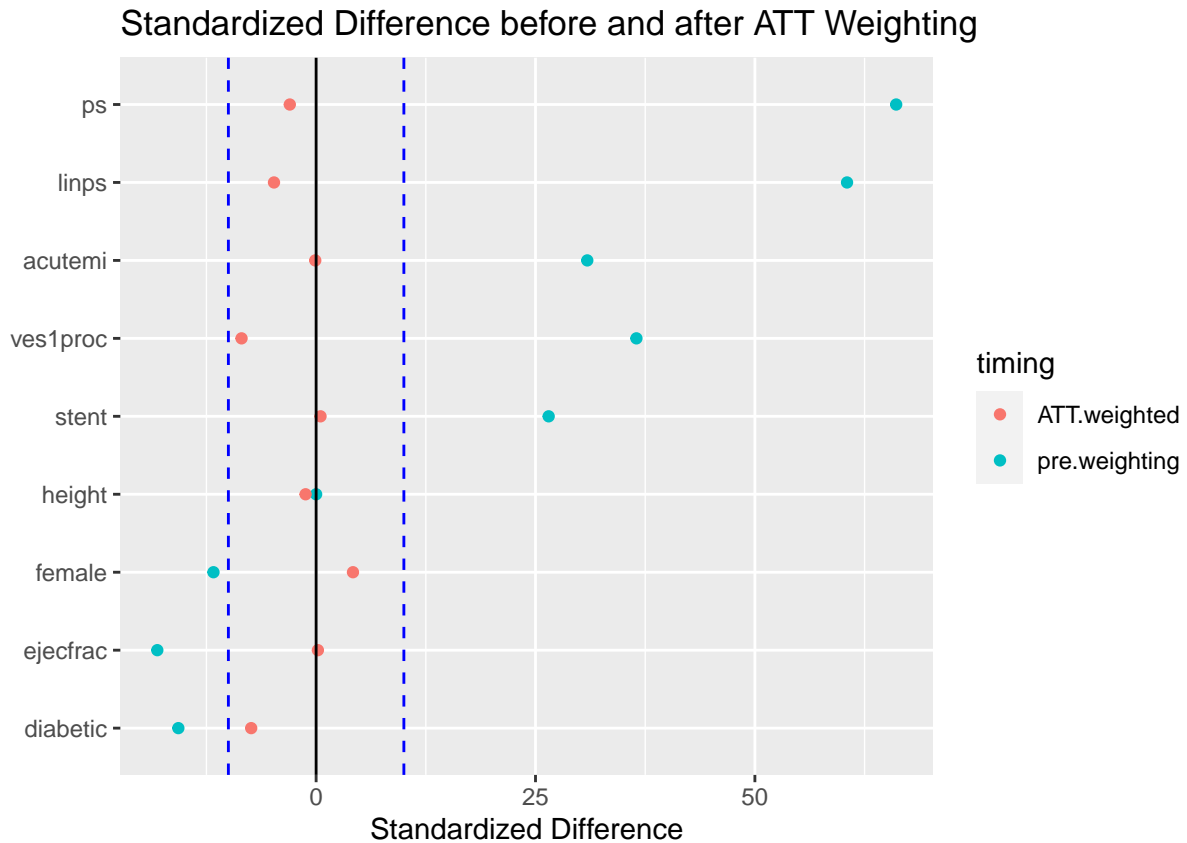
```
[[2]]
```

	tx.mn	tx.sd	ct.mn	ct.sd	std.eff.sz	stat	p	ks	ks.pval
stent	0.705	0.456	0.702	0.458	0.005	0.065	0.948	0.002	1.000
height	171.443	10.695	171.568	11.934	-0.012	-0.102	0.919	0.042	0.974
female	0.331	0.471	0.311	0.464	0.042	0.497	0.620	0.020	1.000
diabetic	0.205	0.404	0.235	0.425	-0.074	-0.716	0.474	0.030	1.000
acutemi	0.179	0.384	0.180	0.385	-0.001	-0.011	0.991	0.001	1.000
ejecfrac	50.403	10.419	50.384	10.358	0.002	0.019	0.985	0.032	0.999
veslproc	1.463	0.706	1.523	0.749	-0.085	-0.647	0.518	0.038	0.990
ps	0.727	0.130	0.730	0.134	-0.030	-0.273	0.785	0.061	0.725
linps	1.115	0.796	1.153	0.839	-0.048	-0.360	0.719	0.061	0.725

```
bal.before.wts1 <- bal.table(bal.wts1)[1]
bal.after.wts1 <- bal.table(bal.wts1)[2]
balance.att.weights <- data_frame(names = rownames(bal.before.wts1$unw),
pre.weighting = 100*bal.before.wts1$unw$std.eff.sz,
ATT.weighted = 100*bal.after.wts1[[1]]$std.eff.sz)
balance.att.weights <- gather(balance.att.weights, timing, szd, 2:3)
```

Now we can plot the standardized differences after ATT weighting.

```
ggplot(balance.att.weights, aes(x = szd, y = reorder(names, szd), color = timing)) +
  geom_point() +
  geom_vline(xintercept = 0) +
  geom_vline(xintercept = c(-10,10), linetype = "dashed", col = "blue") +
  labs(x = "Standardized Difference",
       y = "",
       title = "Standardized Difference before and after ATT Weighting")
```



The standardized differences look much better here in this approach.

13.2.1 Rubin's Rules

13.2.1.1 Rule 1 Numbers from balance table above: $(-0.048 * 100) = 4.8\%$. So passes Rule 1.

13.2.1.2 Rule 2 Numbers from balance table above: $(0.796^2)/(0.839^2) = 0.9001237$. Passes Rule 2

13.2.2 Estimated effect on outcomes after ATT weighting

13.2.2.1 Quantitative outcome To estimate the effect of the treatment on `cardbill`, we'll use `svyglm` from the `survey` package to apply the ATT weights in a linear model.

```
lindnerwt1.design <- svydesign(ids=~1, weights=~wts1, data=lindner_clean) # using ATT weights
adjout1.wt1 <- svyglm(cardbill ~ treated, design=lindnerwt1.design)
wt_att_results1 <- tidy(adjout1.wt1, conf.int = TRUE) %>% filter(term == "treated")
wt_att_results1
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value  conf.low  conf.high
<chr>      <dbl>    <dbl>    <dbl>   <dbl>   <dbl>    <dbl>
1 treated    -239.    1417.    -0.169  0.866  -3017.   2538.
```

Estimate (95%CI) -239.28 (-3016.54, 2537.99)

13.2.2.2 Binary outcome We'll do similar coding for the binary outcome.

```
adjout2.wt1 <- svyglm(sixMonthSurvive ~ treated, design=lindnerwt1.design, family=quasibinomial())
```

```
wt_att_results2 <- tidy(adjout2.wt1, conf.int = TRUE, exponentiate = TRUE) %>%  
filter(term == "treated")  
wt_att_results2
```

```
# A tibble: 1 x 7  
  term      estimate std.error statistic  p.value conf.low conf.high  
  <chr>      <dbl>    <dbl>    <dbl>   <dbl>   <dbl>   <dbl>  
1 treated      6.50     0.537      3.49 0.000509      2.27     18.6
```

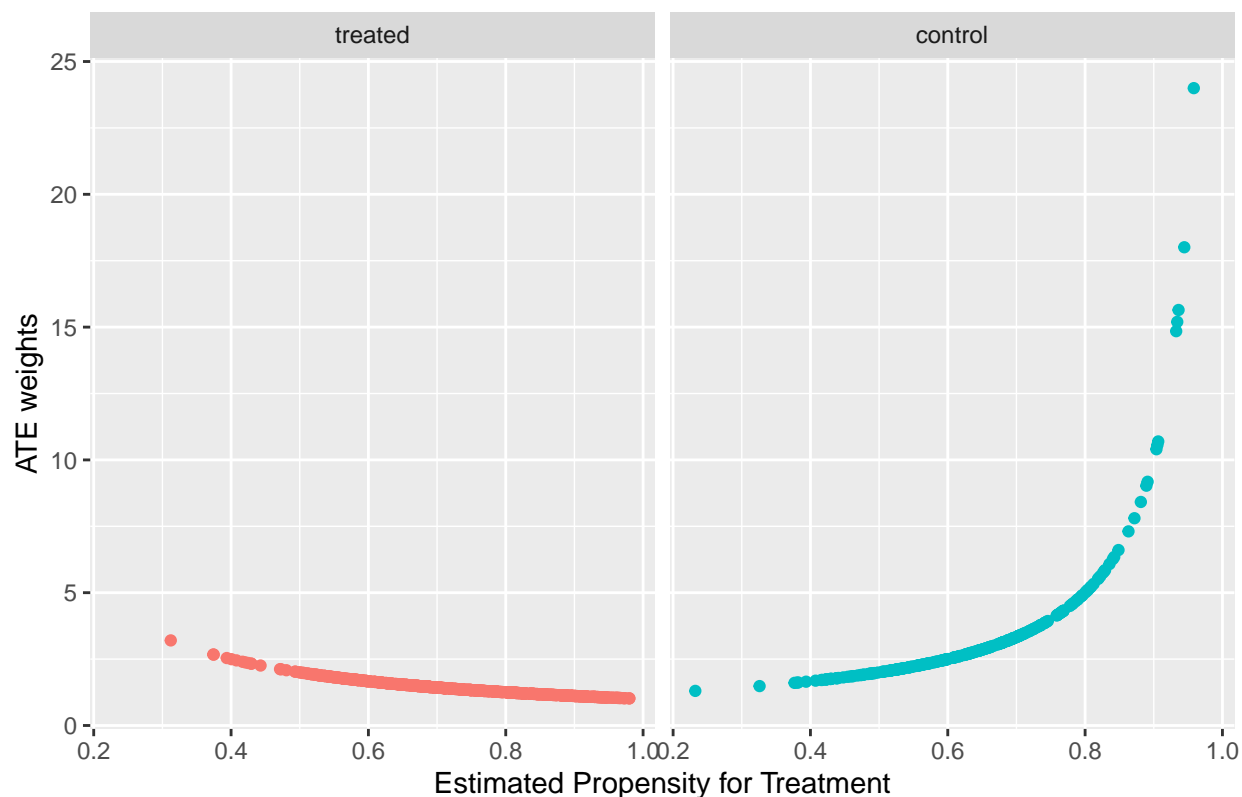
Estimate (95%CI) 6.5 (2.27, 18.63)

13.3 Working with the ATE weights

Now, we'll go through the same steps with the ATE weights.

```
ggplot(lindner_clean, aes(x = ps, y = wts2, color = treated_f)) +  
geom_point() +  
guides(color = FALSE) +  
facet_wrap(~ treated_f) +  
labs(x = "Estimated Propensity for Treatment",  
y = "ATE weights",  
title = "ATE weighting structure")
```

ATE weighting structure



```
bal.wts2 <- dx.wts(x=lindner_clean_df$wts2, data=lindner_clean_df, vars=covlist,
  treat.var="treated", estimand="ATE")
```

```
bal.wts2
```

	type	n.treat	n.ctrl	ess.treat	ess.ctrl	max.es	mean.es	max.ks
1	unw	698	298	698.000	298.0000	0.64205075	0.29974928	0.27599469
2		698	298	671.093	199.6805	0.06759172	0.02390944	0.04595042

	mean.ks	iter
1	0.13749095	NA
2	0.02622715	NA

```
bal.table(bal.wts2)
```

\$unw	tx.mn	tx.sd	ct.mn	ct.sd	std.eff.sz	stat	p	ks	ks.pval
stent	0.705	0.456	0.584	0.494	0.257	3.624	0.000	0.121	0.004
height	171.443	10.695	171.446	10.589	0.000	-0.005	0.996	0.025	0.999
female	0.331	0.471	0.386	0.488	-0.115	-1.647	0.100	0.055	0.531
diabetic	0.205	0.404	0.268	0.444	-0.152	-2.127	0.034	0.064	0.349
acutemi	0.179	0.384	0.060	0.239	0.338	5.923	0.000	0.119	0.005
ejecfrac	50.403	10.419	52.289	10.297	-0.181	-2.640	0.008	0.114	0.008
veslproc	1.463	0.706	1.205	0.480	0.393	6.693	0.000	0.188	0.000
ps	0.727	0.130	0.641	0.123	0.642	9.928	0.000	0.276	0.000
linps	1.115	0.796	0.633	0.616	0.619	10.321	0.000	0.276	0.000

```
[[2]]
```

	tx.mn	tx.sd	ct.mn	ct.sd	std.eff.sz	stat	p	ks	ks.pval
stent	0.670	0.470	0.667	0.472	0.006	0.081	0.936	0.003	1.000
height	171.404	10.602	171.532	11.552	-0.012	-0.124	0.902	0.038	0.974
female	0.344	0.475	0.333	0.472	0.022	0.283	0.777	0.010	1.000
diabetic	0.223	0.416	0.245	0.431	-0.052	-0.601	0.548	0.022	1.000
acutemi	0.143	0.351	0.144	0.352	-0.003	-0.026	0.979	0.001	1.000
ejecfrac	50.943	10.109	50.948	10.377	0.000	-0.006	0.995	0.042	0.934
ves1proc	1.384	0.663	1.428	0.696	-0.068	-0.586	0.558	0.028	0.999
ps	0.701	0.133	0.704	0.137	-0.018	-0.185	0.853	0.046	0.884
linps	0.973	0.774	0.999	0.815	-0.034	-0.292	0.771	0.046	0.884

```

bal.before.wts2 <- bal.table(bal.wts2)[1]
bal.after.wts2 <- bal.table(bal.wts2)[2]
balance.ate.weights <- data_frame(names = rownames(bal.before.wts2$unw),
pre.weighting = 100*bal.before.wts2$unw$std.eff.sz,

ATE.weighted = 100*bal.after.wts2[[1]]$std.eff.sz)
balance.ate.weights <- gather(balance.ate.weights, timing, szd, 2:3)

ggplot(balance.ate.weights, aes(x = szd, y = reorder(names, szd), color = timing)) +
  geom_point() +
  geom_vline(xintercept = 0) +
  geom_vline(xintercept = c(-10,10), linetype = "dashed", col = "blue") +
  labs(x = "Standardized Difference", y = "",
title = "Standardized Difference before and after ATE Weighting")

```



Again, the standardized differences look good here.

13.3.1 Rubin's Rules

13.3.1.1 Rule 1 $-0.033 \times 100 = 3.3\%$. Passes Rule 1 (numbers from ATE weight balance table above).

13.3.1.2 Rule 2 $(0.774^2)/(0.815^2) = 0.9019173$. Passes Rule 2 (numbers from ATE weight balance table above).

13.3.2 Estimated effect on outcomes after ATE weighting

```
lindnerwt2.design <- svydesign(ids=~1, weights=~wts2, data=lindner_clean) # using ATE weights
adjout1.wt2 <- svyglm(cardbill ~ treated, design=lindnerwt2.design)

wt_ate_results1 <- tidy(adjout1.wt2, conf.int = TRUE) %>% filter(term == "treated")
wt_ate_results1
```

13.3.2.1 Quantitative outcome

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1 treated    147.    1192.     0.124    0.902   -2190.    2484.
```

- **Estimate** 147.26 (95% CI: -2189.63, 2484.15)

```
adjout2.wt2 <- svyglm(sixMonthSurvive ~ treated, design=lindnerwt2.design, family=quasibinomial())

wt_ate_results2 <- tidy(adjout2.wt2, conf.int = TRUE, exponentiate = TRUE) %>%
  filter(term == "treated")
wt_ate_results2
```

13.3.2.2 Binary outcome

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
1 treated     5.74     0.503     3.47 0.000538     2.14    15.4
```

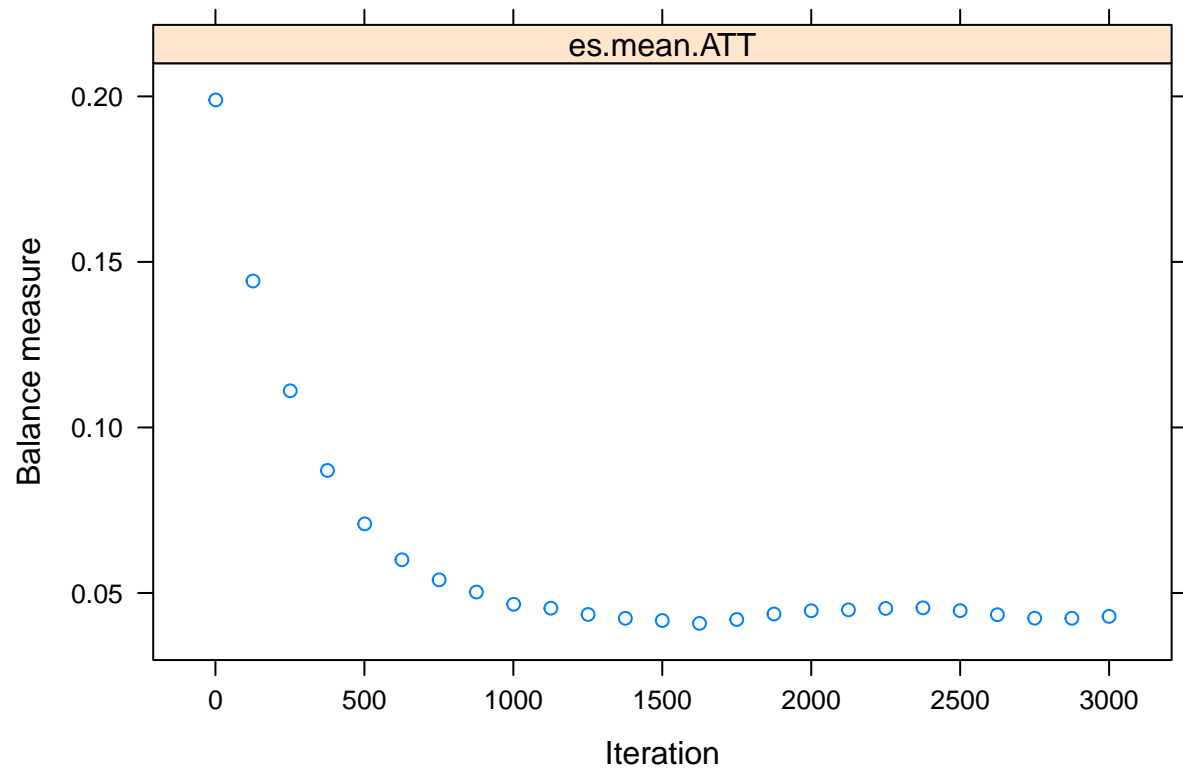
- **Estimate** 5.74 (95% CI: 2.14, 15.38)

14 Task 10: Using TWANG for propensity score estimation and ATT weighting

```
ps.toy <- ps(treated ~ stent + height + female + diabetic + acutemi + ejecfrac + ves1proc,
  data = lindner_clean_df,
  n.trees = 3000,
  interaction.depth = 2,
  stop.method = c("es.mean"),
  estimand = "ATT",
  verbose = FALSE)
```



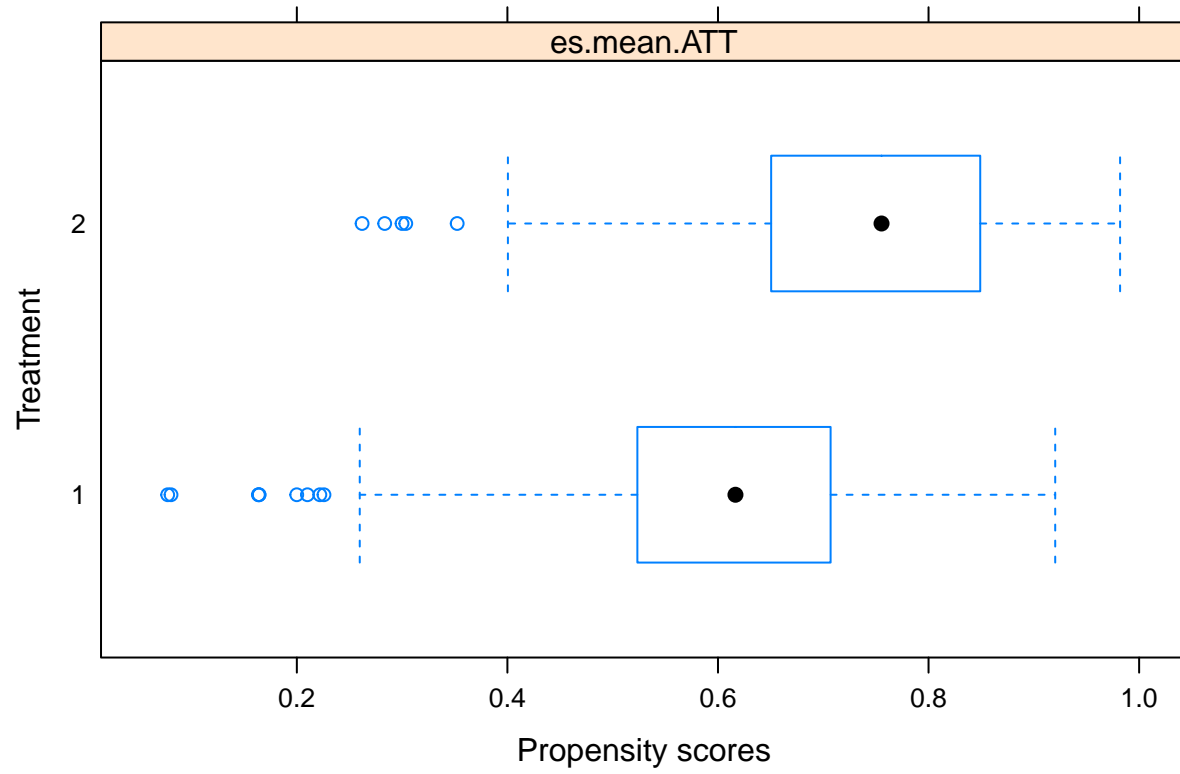
```
plot(ps.toy)
```



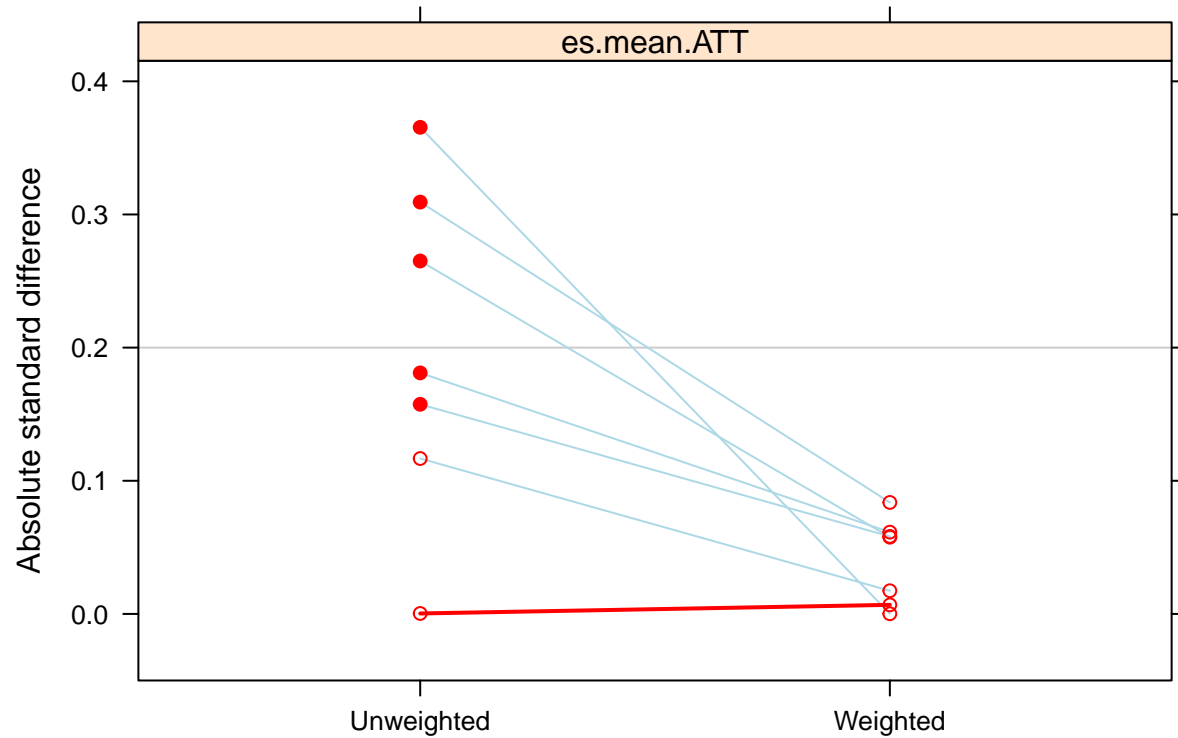
```
summary(ps.toy)
```

	n.treat	n.ctrl	ess.treat	ess.ctrl	max.es	mean.es	max.ks
unw	698	298	698	298.00	0.36544982	0.19933096	0.1884195
es.mean.ATT	698	298	698	172.19	0.08373615	0.04075872	0.0388038
	max.ks.p	mean.ks	iter				
unw	NA	0.09791845	NA				
es.mean.ATT	NA	0.02469335	1628				

```
plot(ps.toy, plots = 2)
```



```
plot(ps.toy, plots = 3)
```



```
bal.tab(ps.toy, full.stop.method = "es.mean.att")
```

Call

```
ps(formula = treated ~ stent + height + female + diabetic + acutemi +
  ejecfrac + veslproc, data = lindner_clean_df, n.trees = 3000,
  interaction.depth = 2, verbose = FALSE, estimand = "ATT",
  stop.method = c("es.mean"))
```

Balance Measures

	Type	Diff.Adj
prop.score	Distance	0.2497
stent	Binary	0.0263
height	Contin.	-0.0068
female	Binary	0.0082
diabetic	Binary	-0.0235
acutemi	Binary	0.0321
ejecfrac	Contin.	-0.0614
veslproc	Contin.	0.0001

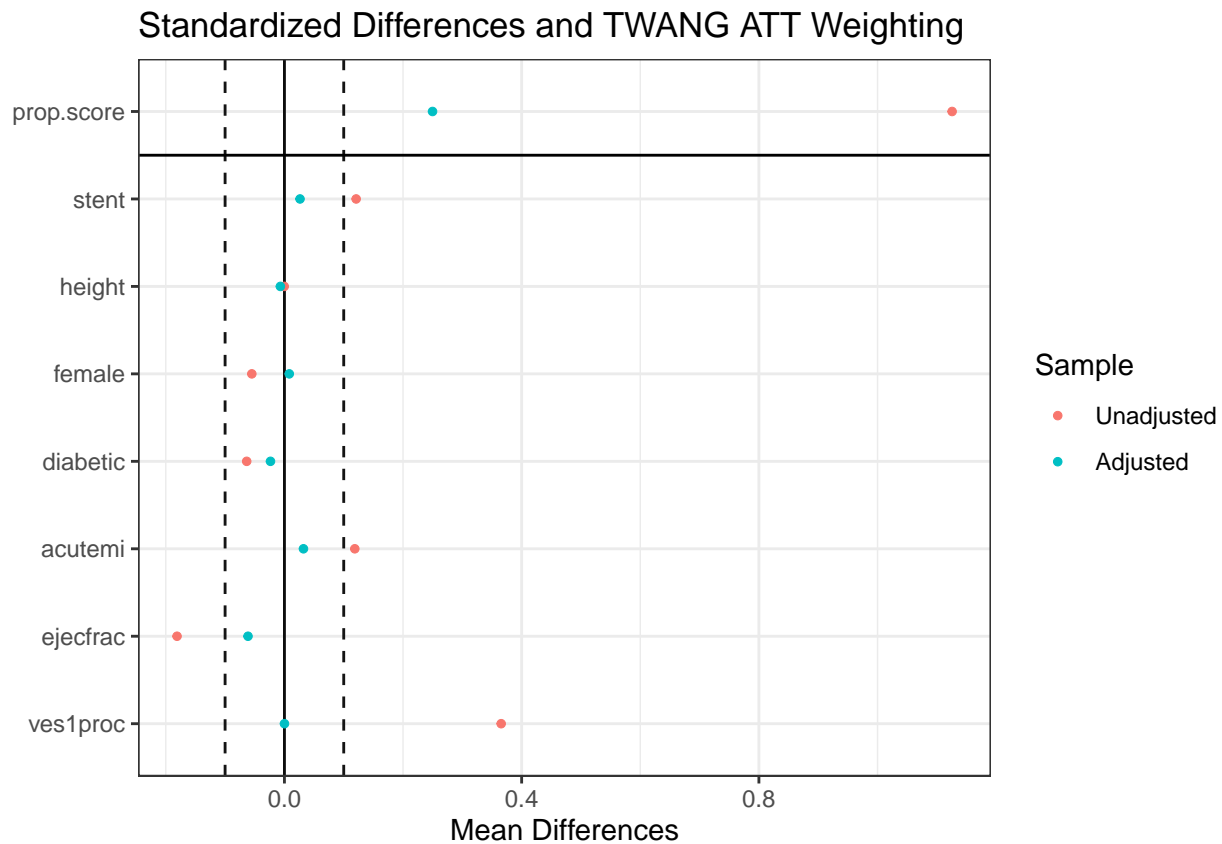
Effective sample sizes

	Control	Treated
Unadjusted	298.	698
Adjusted	172.19	698

```
p <- love.plot(bal.tab(ps.toy),
  threshold = .1, size = 1.5,
  title = "Standardized Differences and TWANG ATT Weighting")
```

Warning: Standardized mean differences and raw mean differences are present in the same plot. Use the 'stars' argument to distinguish between them and appropriately label the x-axis.

```
p + theme_bw()
```



Compared to the manual ATT/ATE weights, the standardized differences look a bit worse here.

14.1 Estimated effect on outcomes after TWANG ATT weighting

14.1.1 Quantitative outcome

```
toywt3.design <- svydesign(ids=~1,
  weights=~get.weights(ps.toy,
  stop.method = "es.mean"),
  data=lindner_clean) # using twang ATT weights

adjout1.wt3 <- svyglm(cardbill ~ treated, design=toywt3.design)
wt_twangatt_results1 <- tidy(adjout1.wt3, conf.int = TRUE) %>% filter(term == "treated")
wt_twangatt_results1
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>    <dbl>    <dbl>   <dbl>   <dbl>    <dbl>
1 treated    501.    1102.     0.454   0.650  -1660.   2661.
```

- **Estimate** 500.51 (95% CI: -1660.15, 2661.17)

14.1.2 Binary outcome

```
adjout2.wt3 <- svyglm(sixMonthSurvive ~ treated, design=toywt3.design,
family=quasibinomial())

wt_twangatt_results2 <- tidy(adjout2.wt3, conf.int = TRUE, exponentiate = TRUE) %>%
filter(term == "treated")
wt_twangatt_results2
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>    <dbl>    <dbl>   <dbl>   <dbl>    <dbl>
1 treated    4.02     0.487     2.86 0.00438    1.55    10.4
```

- Estimate 4.02 (95% CI: 1.55, 10.44)

15 Task 11: After direct adjustment with linear PS

Here we'll directly adjust for the linear propensity score by including it as a covariate in the model.

15.1 Quantitative outcome

```
direct_out1 <- lm(cardbill ~ treated + linps, data=lindner_clean)

adj_out1 <- tidy(direct_out1, conf.int = TRUE) %>% filter(term == "treated")
adj_out1
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>    <dbl>    <dbl>   <dbl>   <dbl>    <dbl>
1 treated   1168.     805.     1.45  0.147   -412.    2748.
```

- Estimate 1167.9 (95% CI: -412.22, 2748.02)

15.2 Binary outcome

```
direct_out2 <- glm(sixMonthSurvive ~ treated + linps, data=lindner_clean, family=binomial())

adj_out2 <- tidy(direct_out2, exponentiate = TRUE, conf.int = TRUE) %>%
filter(term == "treated")
adj_out2
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>    <dbl>    <dbl>   <dbl>   <dbl>    <dbl>
1 treated    4.64     0.438     3.50 0.000463    1.99    11.3
```

- Estimate 4.64 (95% CI: 1.99, 11.27)

16 Task 12: “Double Robust” Approach: Weighting + Direct Adjustment

Here we'll adjust for the linear propensity score and the ATT/ATE/TWANG weights when predicting the quantitative outcome.

16.1 Quantitative outcome

16.1.1 ATT weights

```
design_att <- svydesign(ids=~1, weights=~wts1, data=lindner_clean) # using ATT weights

dr.out1.wt1 <- svyglm(cardbill ~ treated + linps, design=design_att)
dr_att_out1 <- tidy(dr.out1.wt1, conf.int = TRUE) %>% filter(term == "treated")
dr_att_out1
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>    <dbl>    <dbl>   <dbl>   <dbl>    <dbl>
1 treated   -127.    1217.    -0.104  0.917   -2511.   2258.
```

- Estimate -126.72 (95% CI: -2511.33, 2257.89)

16.1.2 ATE weights

```
design_ate<- svydesign(ids=~1, weights=~wts2, data=lindner_clean) # using ATE weights

dr.out1.wt2 <- svyglm(cardbill ~ treated + linps, design=design_ate)
dr_ate_out1 <- tidy(dr.out1.wt2, conf.int = TRUE) %>% filter(term == "treated")
dr_ate_out1
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>    <dbl>    <dbl>   <dbl>   <dbl>    <dbl>
1 treated    217.    1069.     0.203  0.839   -1879.   2312.
```

- Estimate 216.77 (95% CI: -1878.59, 2312.13)

16.1.3 TWANG ATT weights

```
wts3 <- get.weights(ps.toy, stop.method = "es.mean")
twang.design <- svydesign(ids=~1, weights=~wts3, data=lindner_clean) # twang ATT weights

dr.out1.wt3 <- svyglm(cardbill ~ treated + linps, design=twang.design)
dr_twangatt_out1 <- tidy(dr.out1.wt3, conf.int = TRUE) %>% filter(term == "treated")
dr_twangatt_out1
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
<chr>      <dbl>    <dbl>    <dbl>   <dbl>   <dbl>    <dbl>
1 treated    375.    1103.     0.340  0.734   -1787.   2537.
```

- Estimate 375.05 (95% CI: -1787.05, 2537.16)

16.2 Binary outcome

Now we'll adjust for the linear propensity score and the ATT/ATE/TWANG weights when predicting the binary outcome.

16.2.1 ATT weights

```
dr.out2.wt1 <- svyglm(sixMonthSurvive ~ treated + linps, design=design_att,
family=quasibinomial())

dr_att_out2 <- tidy(dr.out2.wt1, exponentiate = TRUE, conf.int = TRUE) %>%
filter(term == "treated")
dr_att_out2
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic  p.value conf.low conf.high
<chr>      <dbl>     <dbl>     <dbl>    <dbl>   <dbl>   <dbl>
1 treated      6.90      0.563      3.43 0.000634    2.29    20.8

• Estimate 6.9 (95% CI: 2.29, 20.81)
```

16.2.2 ATE weights

```
dr.out2.wt2 <- svyglm(sixMonthSurvive ~ treated + linps, design=design_ate,
family=quasibinomial())

dr_ate_out2 <- tidy(dr.out2.wt2, exponentiate = TRUE, conf.int = TRUE) %>%
  filter(term == "treated")

dr_ate_out2
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic  p.value conf.low conf.high
<chr>      <dbl>     <dbl>     <dbl>    <dbl>   <dbl>   <dbl>
1 treated      5.95      0.517      3.45 0.000590    2.16    16.4

• Estimate 5.95 (95% CI: 2.16, 16.39)
```

16.2.3 TWANG ATT weights

```
dr.out2.wt3 <- svyglm(sixMonthSurvive ~ treated + linps, design=twang.design,
family=quasibinomial())

dr_twangatt_out2 <- tidy(dr.out2.wt3, exponentiate = TRUE, conf.int = TRUE) %>%
filter(term == "treated")
dr_twangatt_out2
```

```
# A tibble: 1 x 7
  term      estimate std.error statistic p.value conf.low conf.high
```

	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	treated	4.87	0.554	2.86	0.00436	1.64	14.4

- Estimate 4.87 (95% CI: 1.64, 14.44)

16.3 Session Information

```
xfun::session_info()
```

```
R version 4.0.3 (2020-10-10)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19041)
```

Locale:

```
LC_COLLATE=English_United States.1252
LC_CTYPE=English_United States.1252
LC_MONETARY=English_United States.1252
LC_NUMERIC=C
LC_TIME=English_United States.1252
```

Package version:

askpass_1.1	assertthat_0.2.1	backports_1.2.1
base64enc_0.1-3	BH_1.75.0.0	blob_1.2.1
boot_1.3-27	brio_1.1.1	broom_0.7.5
broom.mixed_0.2.6	callr_3.5.1	cellranger_1.1.0
checkmate_2.0.0	class_7.3-17	cli_2.3.1
clipr_0.7.1	cluster_2.1.0	cmprsk_2.2-10
cobalt_4.3.0	coda_0.19-4	colorspace_2.0-0
compiler_4.0.3	cpp11_0.2.6	crayon_1.4.1
crosstalk_1.1.1	cubelyr_1.0.1	curl_4.3
data.table_1.14.0	DBI_1.1.1	dbplyr_2.1.0
desc_1.2.0	diffobj_0.3.3	digest_0.6.27
dplyr_1.0.4	e1071_1.7-4	ellipsis_0.3.1
Epi_2.43	etm_1.1.1	evaluate_0.14
fansi_0.4.2	farver_2.0.3	forcats_0.5.1
foreign_0.8-81	Formula_1.2-4	fs_1.5.0
gbm_2.1.8	gdata_2.18.0	generics_0.1.0
ggdendro_0.1.22	ggforce_0.3.2	ggformula_0.10.1
ggplot2_3.3.3	ggrepel_0.9.1	ggribes_0.5.3
ggstance_0.3.5	glue_1.4.2	gmodels_2.18.1
graphics_4.0.3	grDevices_4.0.3	grid_4.0.3
gridExtra_2.3	gtable_0.3.0	gtools_3.8.2
haven_2.3.1	here_1.0.1	highr_0.8
Hmisc_4.4-2	hms_1.0.0	htmlTable_2.1.0
htmltools_0.5.1.1	htmlwidgets_1.5.3	httr_1.4.2
isoband_0.2.3	janitor_2.1.0	jpeg_0.1-8.1
jsonlite_1.7.2	knitr_1.31	labeling_0.4.2
labelled_2.7.0	lattice_0.20-41	latticeExtra_0.6-29
lazyeval_0.2.2	leaflet_2.0.4.1	leaflet.providers_1.9.0
lifecycle_1.0.0	lme4_1.1-26	lubridate_1.7.9.2
magrittr_2.0.1	markdown_1.1	MASS_7.3-53.1
Matching_4.9-7	Matrix_1.2-18	methods_4.0.3
mgcv_1.8-33	mime_0.10	minqa_1.2.4

mitools_2.4	modelr_0.1.8	mosaic_1.8.3
mosaicCore_0.9.0	mosaicData_0.20.2	munsell_0.5.0
nlme_3.1-149	nloptr_1.2.2.2	nnet_7.3-15
numDeriv_2016.8-1.1	openssl_1.4.3	parallel_4.0.3
patchwork_1.1.1	pillar_1.5.0	pkgbuild_1.2.0
pkgconfig_2.0.3	pkgload_1.2.0	plyr_1.8.6
png_0.1-7	polyclip_1.10-0	praise_1.0.0
prettyunits_1.1.1	processx_3.4.5	progress_1.2.2
ps_1.5.0	purrr_0.3.4	R6_2.5.0
raster_3.4.5	RColorBrewer_1.1-2	Rcpp_1.0.6
RcppArmadillo_0.10.2.1.0	RcppEigen_0.3.3.9.1	readr_1.4.0
readxl_1.3.1	rematch_1.0.1	rematch2_2.1.2
reprex_1.0.0	reshape2_1.4.4	rlang_0.4.10
rmarkdown_2.7	rpart_4.1-15	rprojroot_2.0.2
rstudioapi_0.13	rvest_0.3.6	scales_1.1.1
selectr_0.4.2	snakecase_0.11.0	sp_1.4.5
splines_4.0.3	statmod_1.4.35	stats_4.0.3
stringi_1.5.3	stringr_1.4.0	survey_4.0
survival_3.2-7	sys_3.4	tableone_0.12.0
testthat_3.0.2	tibble_3.1.0	tidyr_1.1.2
tidyselect_1.1.0	tidyverse_1.3.0	tinytex_0.29
TMB_1.7.19	tools_4.0.3	twang_1.6
tweenr_1.0.1	utf8_1.1.4	utils_4.0.3
vctrs_0.3.6	viridis_0.5.1	viridisLite_0.3.0
waldo_0.2.4	withr_2.4.1	xfun_0.21
xml2_1.3.2	xtable_1.8-4	yaml_2.2.1
zoo_1.8-8		