500 Homework 5 Answer Sketch

Thomas E. Love

Due 2019-04-04 at noon. Version: 2019-03-18 13:27:49

Contents


```
library(Epi)
library(Matching)
library(tableone)
```

```
library(cobalt)
library(rbounds)
library(survival)
library(twang)
library(survey)
library(skimr)
library(tidyverse)
```

1 Selecting an Exposure, Outcome and Population

1.1 What the Class Did (2018)

Students in 2018 selected all kinds of things as treatments and outcomes, and some also restricted the sample to look at particular subpopulations of interest. Thanks for doing that.

- Everyone selected a binary outcome
  - Death
  - Angina
  - Hospitalization due to respiratory infection
  - Respiratory infection
  - MI
  - Stroke
- Treatments people considered were:
  - Digoxin was the most common choice
  - Diabetes
  - Hydralazine
  - Pulmonary congestion
  - Rales
  - ACE-inhibitor
  - Nitrates
  - Hypertension

As a result, pretty much whatever I choose, I'm going to miss something that people were interested in. Rather than write many answer sketches, I'll just select something that doesn't actually match any of those folks, and we'll see how that works out.

1.2 My Choices

- I will study the subpopulation of patients who have no prior MI (`PREVMI` == 0).
- The exposure of interest is NYHA Functional Class (`FUNCTCLS`) of III or IV, as compared to I or II.
- The outcome I'll study is all-cause hospitalization (`HOSP`).

I am anticipating that among the patients without a prior myocardial infarction, those with baseline NYHA Class III or IV will be hospitalized more freqently than those with NYHA Class I or II.

The covariates I'll study (selected at least in part to try to cover each kind of variable people had trouble with) are:

1. ejection fraction (`EJF_PER`)
2. sex (`SEX`)
3. age (`AGE`)
4. race (`RACE`)
5. body-mass index (`BMI`)
6. serum creatinine level (`CREAT`)
7. heart rate (`HEARTRTE`)
8. systolic blood pressure (`SYSBP`)
9. diastolic blood pressure (`DIABP`)
10. baseline angina (`ANGINA`)
11. history of diabetes (`DIABETES`)
12. history of hypertension (`HYPERTEN`)
13. use of potassium-sparing diuretics (`DIURETK`)
14. present and past status of pulmonary edema (`PEDEMA`)
15. present and past status of rales (`RALES`)

These covariates include: - quantities measured as continuous (`EJF_PER`, `AGE`, `BMI`, `CREAT`, `HEARTRTE`, `SYSBP`, `DIABP`) - binary variables (`SEX`, `RACE`, `ANGINA`, `DIABETES`, `HYPERTEN`, `DIURETK`) - multi-categorical variables (`PEDEMA`, `RALES`)

2 Cleaning the Data

The code below reads in the data, and selects my population and the variables I'll use. Among the 2,380 subjects with `PREVMI` of 0 in the main data set, all but 11 have complete data on my selected variables[1] so I'll just simply my life by dropping all 11 cases with missing values.

```
dig <- read.csv("dig1.csv") %>% tbl_df %>%
    filter(PREVMI == 0) %>%
    select(subjectid, PREVMI, FUNCTCLS, HOSP, EJF_PER,
           SEX, AGE, RACE, BMI, CREAT, HEARTRTE, SYSBP,
           DIABP, ANGINA, DIABETES, HYPERTEN, DIURETK,
           PEDEMA, RALES) %>%
    drop_na ## only 11 subjects have any missing values
```

[1] I won't lie. That's part of the reason I selected them.

2.1 Exposure

Next, I'll create my exposure variable.

- I'll call the 1/0 version `badNYHA`, which is 1 if the patient's NYHA functional class is III or IV, and 0 otherwise. This is the version I'll use in modeling.
- I'll call the factor version `NYHA_f` which takes the values "III or IV" or "I or II" corresponding directly to the NYHA functional class. This is the version I'll use in summary tables.

```
dig <- dig %>%
    mutate(badNYHA = ifelse(FUNCTCLS %in% c(3, 4), 1, 0),
           NYHA_f = fct_recode(factor(badNYHA),
                               "III or IV" = "1", "I or II" = "0"),
           NYHA_f = fct_relevel(NYHA_f, "III or IV"))

dig %>% count(badNYHA, NYHA_f)
```

```
# A tibble: 2 x 3
  badNYHA NYHA_f        n
    <dbl> <fct>     <int>
1       0 I or II    1609
2       1 III or IV   760
```

Of the 2,369 subjects in my subpopulation, 760 (32%) have the exposure of interest (NYHA class III or IV.)

2.2 Outcome

Next, I'll set up a factor version of my outcome variable.

```
dig <- dig %>%
    mutate(hosp_f = fct_recode(factor(HOSP),
            "Hospitalized" = "1", "Not Hosp." = "0"),
           hosp_f = fct_relevel(hosp_f, "Hospitalized"))

dig %>% count(hosp_f, HOSP)
```

```
# A tibble: 2 x 3
  hosp_f        HOSP     n
  <fct>        <int> <int>
1 Hospitalized     1  1558
2 Not Hosp.        0   811
```

Of the 2,369 subjects in my subpopulation, 1558 (66%) were hospitalized.

2.3 Binary Covariates: Cleanup

```
dig <- dig %>%
    mutate(female = ifelse(SEX == 2, 1, 0),
     sex_f = fct_recode(factor(SEX), "Male" = "1",
                        "Female" = "2"),
     white = ifelse(RACE == 1, 1, 0),
     race_f = fct_recode(factor(RACE), "White" = "1",
                         "Non-White" = "2"),
     angina_f = fct_recode(factor(ANGINA),
                           "Yes" = "1", "No" = "0"),
     dm_f = fct_recode(factor(DIABETES),
                       "Yes" = "1", "No" = "0"),
     htn_f = fct_recode(factor(HYPERTEN),
                        "Yes" = "1", "No" = "0"),
     diurk_f = fct_recode(factor(DIURETK),
                          "Yes" = "1", "No" = "0"))
```

2.4 Multi-Categorical Variables

```
dig <- dig %>%
    mutate(
      ped_f = fct_recode(factor(PEDEMA),
                         "None" = "0",
                         "Present" = "1",
                         "Past" = "2",
                         "Present & Past" = "3"),
      ped_pres = ifelse(PEDEMA %in% c(1, 3), 1, 0),
      ped_past = ifelse(PEDEMA %in% c(2, 3), 1, 0),
      rales_f = fct_recode(factor(RALES),
                         "None" = "0",
                         "Present" = "1",
                         "Past" = "2",
                         "Present & Past" = "3"),
      rales_pres = ifelse(RALES %in% c(1, 3), 1, 0),
      rales_past = ifelse(RALES %in% c(2, 3), 1, 0)
    )
```

3 Task 1. Build a Table 1

```
v1 <- c("EJF_PER", "sex_f", "AGE", "race_f", "BMI", "CREAT",
       "HEARTRTE", "SYSBP", "DIABP", "angina_f", "dm_f",
       "htn_f", "diurk_f", "ped_f", "rales_f")
fv1 <- c("sex_f", "race_f", "angina_f", "dm_f", "htn_f",
        "diurk_f", "ped_f", "rales_f")

CreateTableOne(vars = v1, strata = "NYHA_f",
               factorVars = fv1, data = dig)
```

```
                       Stratified by NYHA_f
                        III or IV      I or II        p      test
  n                        760           1609
  EJF_PER (mean (SD))    25.87 (8.99)  29.58 (8.52)  <0.001
  sex_f = Female (%)       218 (28.7)    325 (20.2)  <0.001
  AGE (mean (SD))        63.97 (10.58) 63.33 (11.02)  0.179
  race_f = Non-White (%)    95 (12.5)    251 (15.6)   0.053
  BMI (mean (SD))        27.24 (5.58)  27.12 (5.07)   0.602
  CREAT (mean (SD))       1.31 (0.38)   1.29 (0.37)   0.235
  HEARTRTE (mean (SD))   78.46 (12.48) 78.27 (12.86)  0.740
  SYSBP (mean (SD))     125.44 (19.74) 125.76 (19.67) 0.707
  DIABP (mean (SD))      75.26 (12.26) 75.14 (10.96)  0.806
  angina_f = Yes (%)       203 (26.7)    429 (26.7)   1.000
  dm_f = Yes (%)           187 (24.6)    474 (29.5)   0.016
  htn_f = Yes (%)          367 (48.3)    741 (46.1)   0.330
  diurk_f = Yes (%)         75 ( 9.9)    117 ( 7.3)   0.037
  ped_f (%)                                           0.925
     None                  370 (48.7)    794 (49.3)
     Present                46 ( 6.1)     89 ( 5.5)
     Past                  238 (31.3)    493 (30.6)
     Present & Past        106 (13.9)    233 (14.5)
  rales_f (%)                                         0.735
     None                  215 (28.3)    479 (29.8)
     Present                48 ( 6.3)    114 ( 7.1)
     Past                  422 (55.5)    859 (53.4)
     Present & Past         75 ( 9.9)    157 ( 9.8)
```

Four or five variables show a large and significant difference here (for instance, ejection fraction, sex, diabetes, potassium-sparing diuretics and perhaps race), but most of the others show only tiny distinctions between the I-II vs. III-IV groups. The angina rate, coincidentally, is exactly the same in each exposure group in this sample. We'd expect that the results will look a bit different after propensity adjustment.

4 Task 2. Unadjusted Analysis of Exposure on Outcome

Before any sort of propensity adjustment, the effect of NYHA functional class (III or IV vs. I or II) on hospitalization rates looks highly significant[2]

```
twoby2(dig$NYHA_f, dig$hosp_f)
```

```
2 by 2 table analysis:
------------------------------------------------------
Outcome   : Hospitalized
Comparing : III or IV vs. I or II

          Hospitalized Not Hosp.    P(Hospitalized) 95% conf. interval
III or IV          547       213             0.7197    0.6867   0.7505
I or II           1011       598             0.6283    0.6044   0.6516

                                    95% conf. interval
            Relative Risk: 1.1455    1.0808   1.2140
         Sample Odds Ratio: 1.5190    1.2589   1.8329
Conditional MLE Odds Ratio: 1.5187    1.2542   1.8427
    Probability difference: 0.0914    0.0510   0.1303

             Exact P-value: 0
        Asymptotic P-value: 0
------------------------------------------------------
```

The unadjusted estimate of the odds ratio for bad NYHA class vs. good NYHA class on hospitalization is 1.52, with 95% CI (1.26, 1.83).

5 Fit the Propensity Score Model

```
psmod <- glm(badNYHA ~ EJF_PER + sex_f + AGE + race_f +
                 BMI + CREAT + HEARTRTE + SYSBP + DIABP +
                 ANGINA + DIABETES + HYPERTEN + DIURETK +
                 ped_f + rales_f, family = binomial(),
             data = dig)
summary(psmod)
```

```
Call:
glm(formula = badNYHA ~ EJF_PER + sex_f + AGE + race_f + BMI +
```

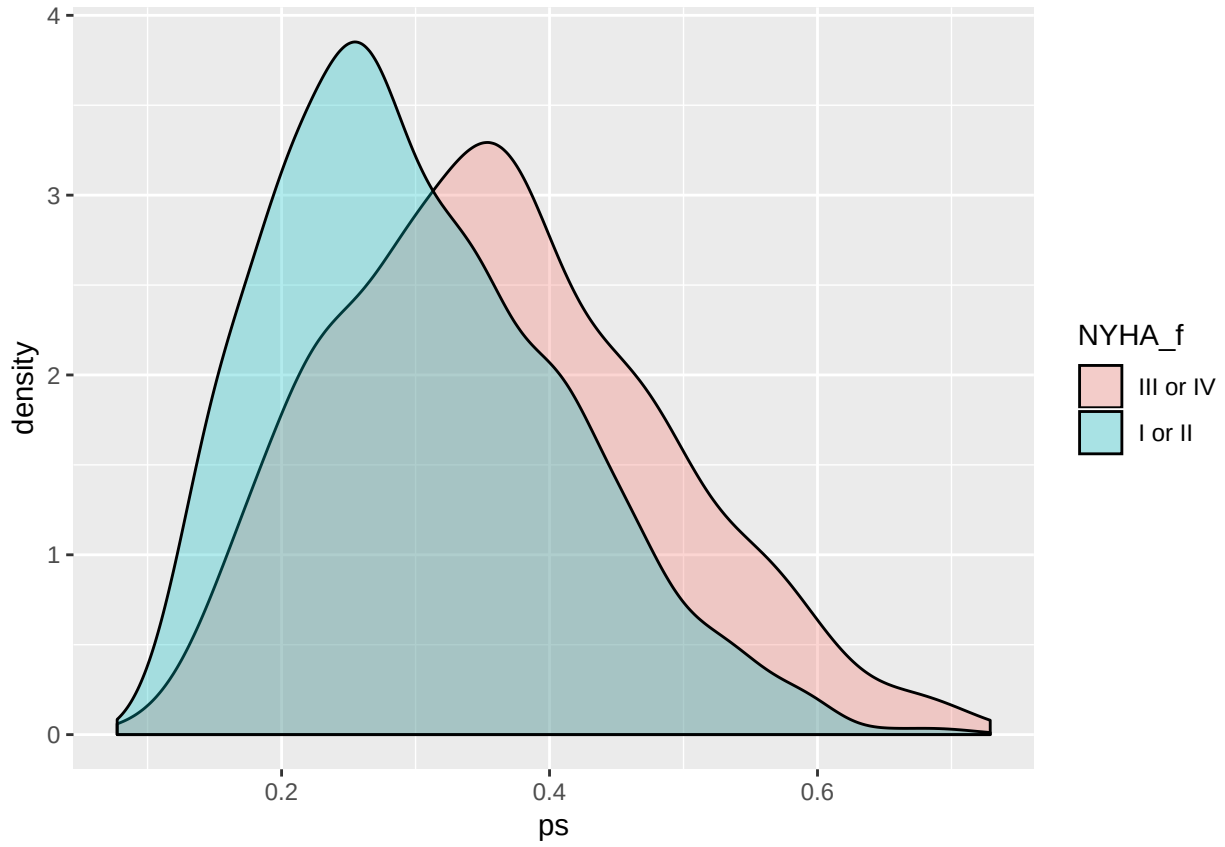
[2]This is another reason why I selected the setup I did.

```
    CREAT + HEARTRTE + SYSBP + DIABP + ANGINA + DIABETES + HYPERTEN +
    DIURETK + ped_f + rales_f, family = binomial(), data = dig)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5755  -0.8918  -0.7126   1.2494   2.2628

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)           -0.0751881  0.6486954  -0.116  0.90773
EJF_PER               -0.0527486  0.0053209  -9.913  < 2e-16 ***
sex_fFemale            0.5984494  0.1060298   5.644 1.66e-08 ***
AGE                    0.0042810  0.0042131   1.016  0.30957
race_fNon-White       -0.3743204  0.1346764  -2.779  0.00545 **
BMI                    0.0011836  0.0086954   0.136  0.89173
CREAT                  0.1752097  0.1212783   1.445  0.14854
HEARTRTE               0.0009948  0.0035933   0.277  0.78191
SYSBP                 -0.0003159  0.0023222  -0.136  0.89179
DIABP                  0.0013091  0.0040037   0.327  0.74370
ANGINA                 0.0286657  0.1031219   0.278  0.78103
DIABETES              -0.2743120  0.1039463  -2.639  0.00832 **
HYPERTEN               0.0848974  0.0912568   0.930  0.35221
DIURETK                0.3559687  0.1620173   2.197  0.02801 *
ped_fPresent           0.1728499  0.2134553   0.810  0.41807
ped_fPast             -0.0346594  0.1079195  -0.321  0.74809
ped_fPresent & Past   -0.0729451  0.1415190  -0.515  0.60624
rales_fPresent        -0.1398178  0.2086093  -0.670  0.50271
rales_fPast            0.0655406  0.1083128   0.605  0.54511
rales_fPresent & Past  0.0759205  0.1729542   0.439  0.66069
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2973.0  on 2368  degrees of freedom
Residual deviance: 2825.7  on 2349  degrees of freedom
AIC: 2865.7

Number of Fisher Scoring iterations: 4
```

```
dig$ps <- fitted(psmod) # propensity score
dig$linps <- psmod$linear.predictors # linear PS
```

5.1 Overlap of the propensity scores by exposure group

```
ggplot(dig, aes(x = ps, fill = NYHA_f)) +
    geom_density(alpha = 0.3)
```



5.2 Rubin's Rule 1 for the unadjusted comparison

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

```
[1] 52.86382
```

5.3 Rubin's Rule 2 for the unadjusted comparison

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

```
[1] 1.102277
```

6 Task 3: Analysis using Propensity Score Matching

I'll do a 1:1 greedy match.

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

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

Original number of observations..............  2369
Original number of treated obs...............  760
Matched number of observations...............  760
Matched number of observations  (unweighted).  760
```

6.1 Using `cobalt` to build a "Love Plot" after Matching

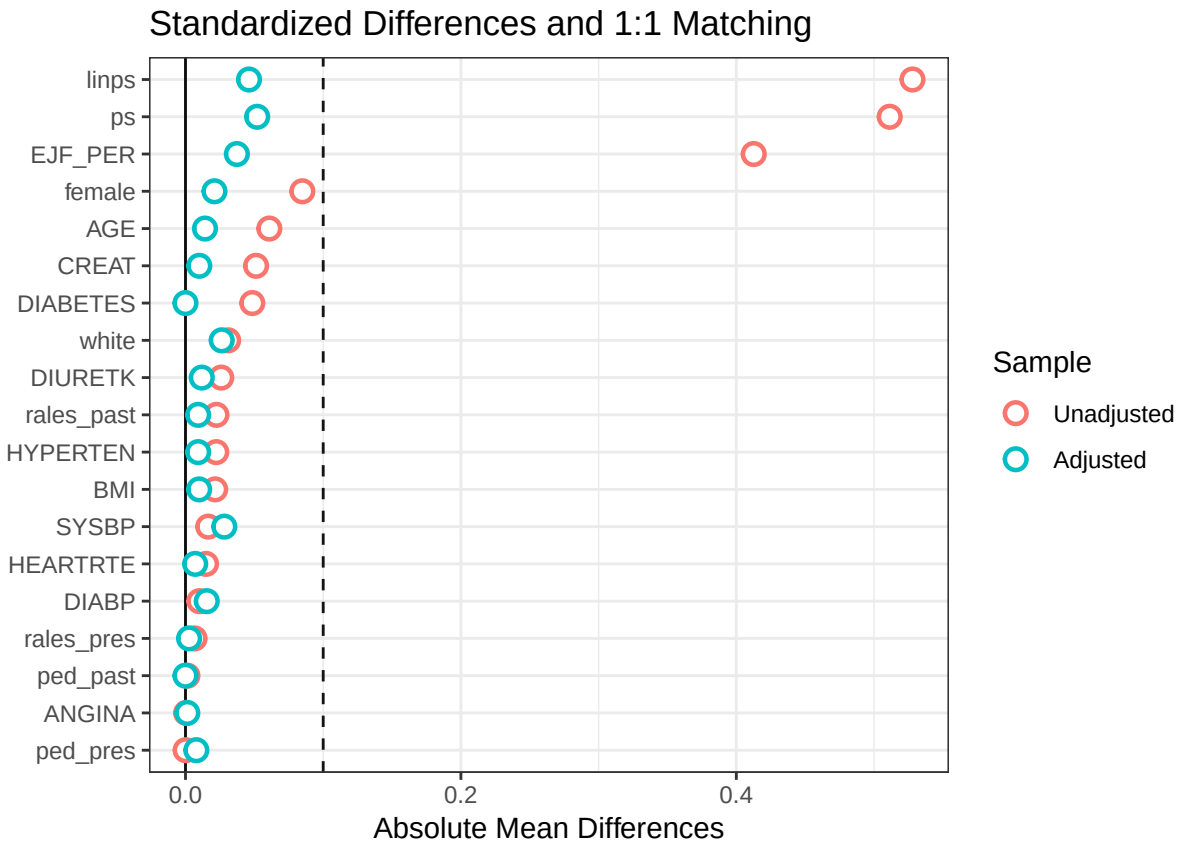
```
b <- bal.tab(match1,
             badNYHA ~ EJF_PER + female + AGE + white +
                 BMI + CREAT + HEARTRTE + SYSBP + DIABP +
                 ANGINA + DIABETES + HYPERTEN + DIURETK +
                 ped_pres + ped_past + rales_pres +
                 rales_past + ps + linps,
             data=dig, un = TRUE)
b
```

```
Balance Measures
              Type Diff.Un Diff.Adj
EJF_PER    Contin. -0.4126  -0.0373
female      Binary  0.0849   0.0211
AGE        Contin.  0.0609   0.0142
white       Binary  0.0310  -0.0263
BMI        Contin.  0.0215  -0.0099
CREAT      Contin.  0.0513   0.0100
HEARTRTE   Contin.  0.0149  -0.0071
SYSBP      Contin. -0.0165   0.0282
DIABP      Contin.  0.0101   0.0155
ANGINA      Binary  0.0005   0.0013
DIABETES    Binary -0.0485   0.0000
```

```
HYPERTEN    Binary  0.0224  -0.0092
DIURETK     Binary  0.0260   0.0118
ped_pres    Binary -0.0001   0.0079
ped_past    Binary  0.0014   0.0000
rales_pres  Binary -0.0066   0.0026
rales_past  Binary  0.0225   0.0092
ps         Contin.  0.5114   0.0521
linps      Contin.  0.5279   0.0461

Sample sizes
          Control Treated
All          1609     760
Matched       760     760
Unmatched     849       0
```

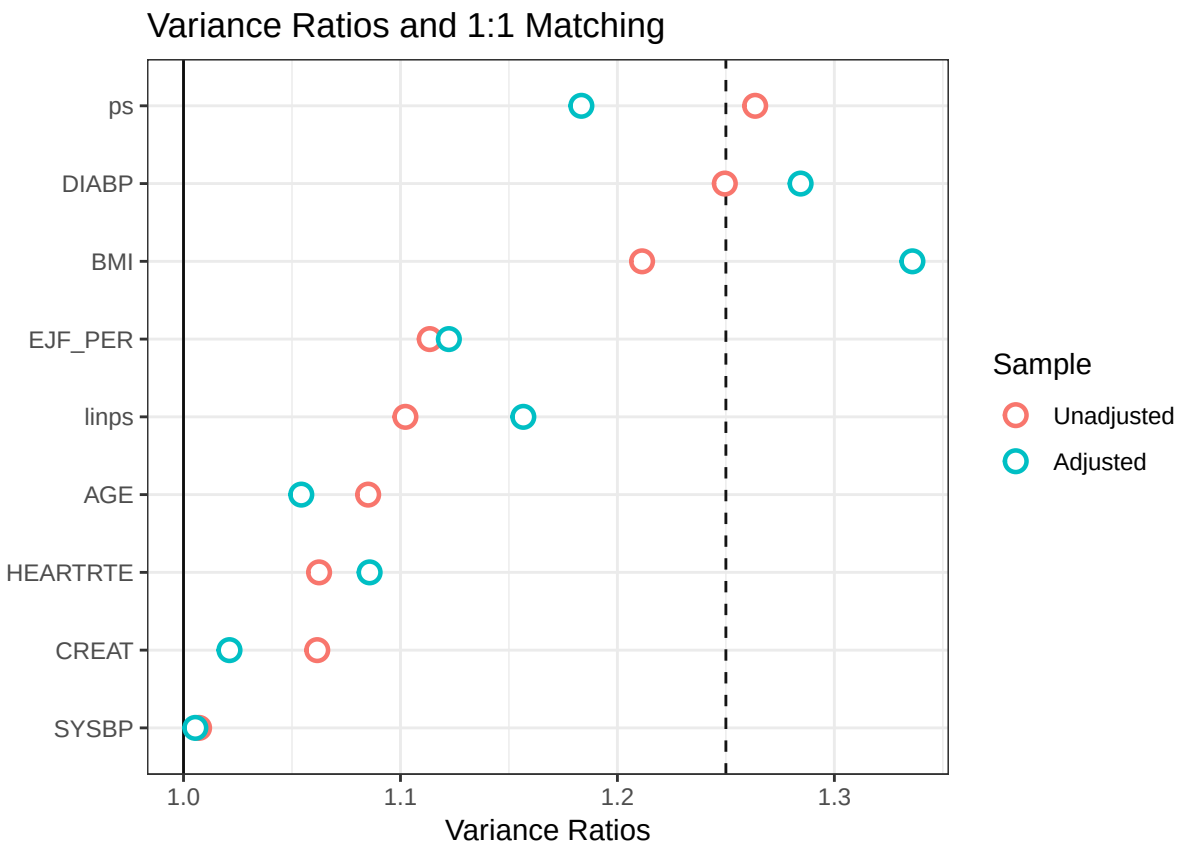
6.1.1 Building a Plot of Standardized Differences, with `cobalt`

```
p <- love.plot(b, threshold = .1, size = 1.5,
        var.order = "unadjusted",
        title = "Standardized Differences and 1:1 Matching")
p + theme_bw()
```



6.1.2 Building a Plot of Variance Ratios, with `cobalt`

```
p <- love.plot(b, stat = "v",
               threshold = 1.25, size = 1.5,
               var.order = "unadjusted",
               title = "Variance Ratios and 1:1 Matching")
p + theme_bw()
```



7 Creating a New Data Frame, Containing the Matched Sample

Now, we build a new matched sample data frame with 760 subjects in each group.

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

Some sanity checks:

```
dig.matched %>% count(NYHA_f)
```

```
# A tibble: 2 x 2
  NYHA_f        n
  <fct>     <int>
1 III or IV   760
2 I or II     760
```

```
head(dig.matched)
```

```
  matches subjectid PREVMI FUNCTCLS HOSP EJF_PER SEX AGE RACE    BMI CREAT
```

```
1       2         5      0        1    1      15   1  74    1 25.654 1.466
2       6      5373      0        2    0      25   1  63    2 25.060 1.011
3      19      4335      0        1    0      17   1  70    2 32.079 1.700
4      21      3493      0        2    1      24   1  74    1 26.651 0.989
5      26       287      0        2    1      38   1  63    1 29.159 1.700
6      28      4255      0        2    0      33   2  70    1 22.503 1.400
  HEARTRTE SYSBP DIABP ANGINA DIABETES HYPERTEN DIURETK PEDEMA RALES
1       84   120    60      0        0        0       0      3     2
2       76   132    95      0        1        0       0      2     3
3       86   108    70      0        0        0       0      2     1
4       82   120    70      0        0        1       0      0     0
5       86   120    78      0        0        0       0      2     2
6       80   120    70      0        0        0       0      3     0
  badNYHA  NYHA_f       hosp_f female  sex_f white    race_f angina_f dm_f
1       0 I or II Hospitalized      0   Male     1     White       No   No
2       0 I or II    Not Hosp.      0   Male     0 Non-White       No  Yes
3       0 I or II    Not Hosp.      0   Male     0 Non-White       No   No
4       0 I or II Hospitalized      0   Male     1     White       No   No
5       0 I or II Hospitalized      0   Male     1     White       No   No
6       0 I or II    Not Hosp.      1 Female     1     White       No   No
  htn_f diurk_f          ped_f ped_pres ped_past        rales_f rales_pres
1    No      No Present & Past        1        1           Past          0
2    No      No           Past        0        1 Present & Past          1
3    No      No           Past        0        1        Present          1
4   Yes      No           None        0        0           None          0
5    No      No           Past        0        1           Past          0
6    No      No Present & Past        1        1           None          0
  rales_past        ps      linps
1          1 0.4636611 -0.1456124
2          1 0.2031840 -1.3665117
3          0 0.3225336 -0.7421525
4          0 0.3544943 -0.5993419
5          1 0.2146811 -1.2969360
6          0 0.3576716 -0.5854844
```

7.1 Rubin's Rule 1 Before and After Matching

Recall that our result without propensity matching (or any other adjustment) was

```
rubin1.unadj
```

```
[1] 52.86382
```

To run this for our matched sample, we use:

```
rubin1.match <- with(dig.matched,
      abs(100*(mean(linps[badNYHA==1]) -
                  mean(linps[badNYHA==0])) /
               sd(linps)))
rubin1.match
```

```
[1] 4.778138
```

An enormous improvement.

7.1.1 Rubin's Rule 2 Before and After Matching

Recall that our result without propensity matching (or any other adjustment) was

```
rubin2.unadj
```

```
[1] 1.102277
```

To run this for our matched sample, we use:

```
rubin2.match <- with(dig.matched,
                     var(linps[badNYHA==1]) /
                         var(linps[badNYHA==0]))
rubin2.match
```

```
[1] 1.156628
```

Still within our desired range of (4/5, 5/4). Looks good.

8 Task 3 Propensity-Matched Analysis

We'll use the matched sample to perform a conditional logistic regression.

```
matched.a <- clogit(HOSP ~ badNYHA + strata(matches),
                    data=dig.matched)
summary(matched.a)
```

```
Call:
coxph(formula = Surv(rep(1, 1520L), HOSP) ~ badNYHA + strata(matches),
    data = dig.matched, method = "exact")

  n= 1520, number of events= 1021

          coef exp(coef) se(coef)     z Pr(>|z|)
badNYHA 0.4349    1.5448   0.1109 3.922 8.77e-05 ***
---
```

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

        exp(coef) exp(-coef) lower .95 upper .95
badNYHA     1.545     0.6473     1.243      1.92

Concordance= 0.607  (se = 0.037 )
Rsquare= 0.01   (max possible= 0.267 )
Likelihood ratio test= 15.75  on 1 df,   p=7e-05
Wald test            = 15.38  on 1 df,   p=9e-05
Score (logrank) test = 15.63  on 1 df,   p=8e-05
```

The odds ratio in the `exp(coef)` section above is the average causal effect estimate - it describes the odds of being hospitalized if you are a subject with a bad NYHA functional class as compared to the odds of hospitalization if you do not have a bad NYHA class.

- Again, the result is highly statistically significant, according to our 95% confidence interval.
- Our estimate, after matching, is 1.51, with 95% CI (1.22, 1.88).

8.1 Sensitivity Analysis

Since we have a significant result, I'll run a sensitivity analysis. We have already used the Match function from the Matching package to develop a matched sample. We can do the analysis in two ways:

8.1.1 Rerun the match, including the outcome

```
X <- dig$linps ## matching on the linear propensity score
Y <- dig$HOSP
Tr <- as.logical(dig$badNYHA)
match1_withY <- Match(Y = Y, Tr=Tr, X=X, M = 1, replace=FALSE, ties=FALSE)
```

Once we've done this, we need only run the `binarysens` function from the `rbounds` package to obtain sensitivity results.

```
binarysens(match1_withY, Gamma = 1.5, GammaInc = 0.05)
```

```
 Rosenbaum Sensitivity Test

Unconfounded estimate ....  0

 Gamma Lower bound Upper bound
  1.00       3e-05     0.00003
```

```
  1.05       0e+00     0.00019
  1.10       0e+00     0.00093
  1.15       0e+00     0.00355
  1.20       0e+00     0.01097
  1.25       0e+00     0.02826
  1.30       0e+00     0.06196
  1.35       0e+00     0.11809
  1.40       0e+00     0.19926
  1.45       0e+00     0.30274
  1.50       0e+00     0.42049

 Note: Gamma is Odds of Differential Assignment To
 Treatment Due to Unobserved Factors
```

With a two-sided hypothesis test at $\alpha = 0.05$, we are insensitive to Γ values up to 1.30 or so. We can use the Table in Rosenbaum, Chapter 9 (and the formulas therein) to see that, for example, this is a little higher than Γ of 1.25, which corresponds to an unobserved covariate that doubles the odds of a bad NYHA class and doubles the odds of a positive difference in the hospitalization rates.

8.1.2 Using the Matched Sample to fit McNemar's Test

The other approach we could take is to run McNemar's test on the matched sample.

```
dig_a <- dig.matched %>% select(matches, NYHA_f, hosp_f)

dig_a2 <- spread(dig_a, key = NYHA_f, value = hosp_f)

head(dig_a2)
```

```
  matches    III or IV      I or II
1       2 Hospitalized Hospitalized
2       6    Not Hosp.    Not Hosp.
3      19 Hospitalized    Not Hosp.
4      21 Hospitalized Hospitalized
5      26 Hospitalized Hospitalized
6      28 Hospitalized    Not Hosp.
```

```
addmargins(table(dig_a2$'III or IV', dig_a2$'I or II'))
```

```
              Hospitalized Not Hosp. Sum
  Hospitalized          340       207 547
  Not Hosp.             134        79 213
  Sum                   474       286 760
```

```
binarysens(x = 140, y = 212, Gamma = 1.30, GammaInc = 0.03)
```

```
 Rosenbaum Sensitivity Test

Unconfounded estimate ....  1e-04

 Gamma Lower bound Upper bound
  1.00       7e-05     0.00007
  1.03       2e-05     0.00022
  1.06       1e-05     0.00057
  1.09       0e+00     0.00139
  1.12       0e+00     0.00310
  1.15       0e+00     0.00637
  1.18       0e+00     0.01219
  1.21       0e+00     0.02183
  1.24       0e+00     0.03676
  1.27       0e+00     0.05850
  1.30       0e+00     0.08837

 Note: Gamma is Odds of Differential Assignment To
 Treatment Due to Unobserved Factors
```

With this setup, we are insensitive up to a Γ of 1.21. The two approaches use different assumptions about the outcome we're interested in, and there's probably some other issue, as well.

9 Task 4 Propensity Weighting Analysis

I'll perform an ATT weighting analysis

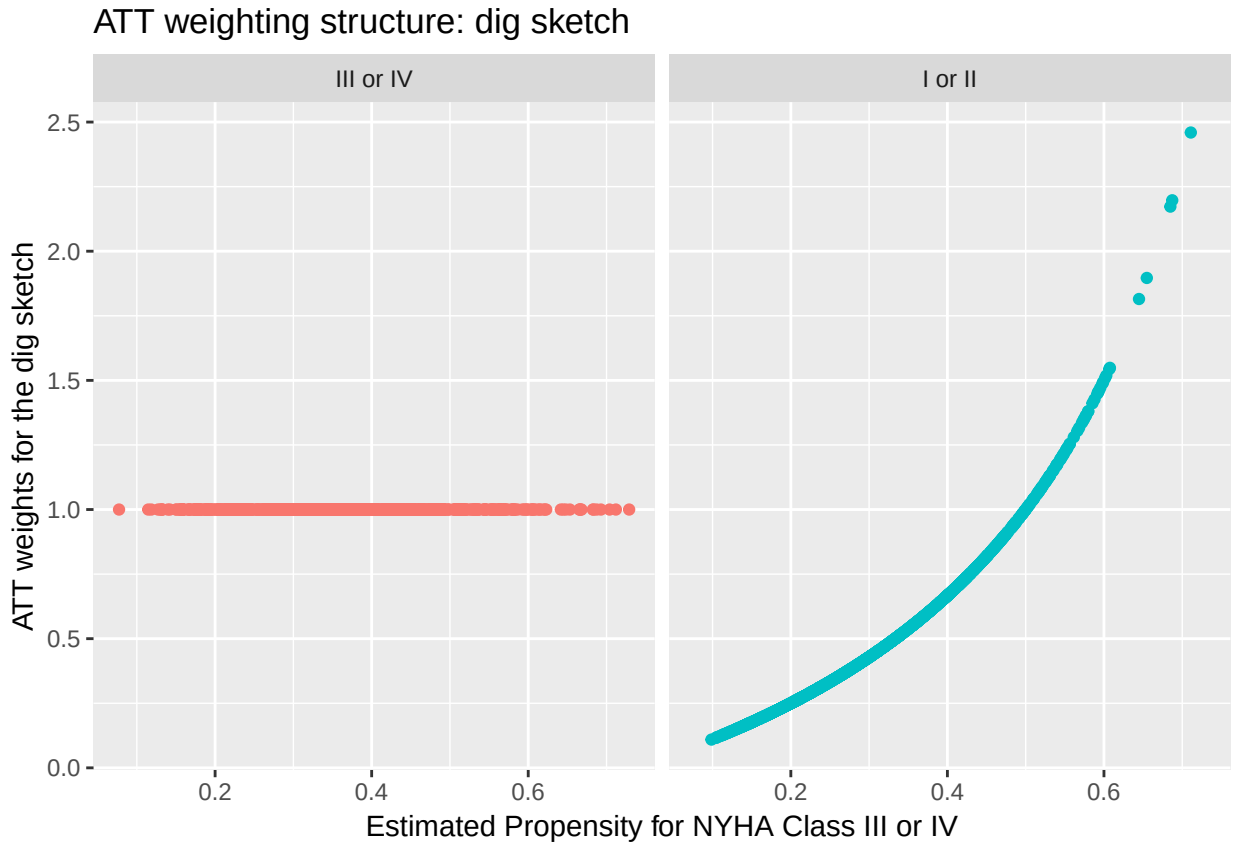
9.1 ATT approach: Weight treated subjects as 1; control subjects as ps/(1-ps)

```
dig$wts1 <- ifelse(dig$badNYHA==1, 1, dig$ps/(1-dig$ps))
```

Here is a plot of the resulting ATT (average treatment effect on the treated) weights:

```
ggplot(dig, aes(x = ps, y = wts1, color = NYHA_f)) +
    geom_point() +
    guides(color = FALSE) +
```

```
    facet_wrap(~ NYHA_f) +
    labs(x = "Estimated Propensity for NYHA Class III or IV",
         y = "ATT weights for the dig sketch",
         title = "ATT weighting structure: dig sketch")
```



9.1.1 Balance Assessment before and after ATT weights

```
dig_df <- data.frame(dig) # twang doesn't react well to tibbles

covlist <- c("EJF_PER", "female", "AGE", "white", "BMI",
             "CREAT", "HEARTRTE", "SYSBP", "DIABP",
             "ANGINA", "DIABETES", "HYPERTEN", "DIURETK",
             "ped_pres", "ped_past", "rales_pres",
             "rales_past", "ps", "linps")

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

```
  type n.treat n.ctrl ess.treat ess.ctrl     max.es     mean.es     max.ks
```

```
1  unw     760   1609       760 1609.000 0.52793050 0.116952342 0.23404615
2          760   1609       760 1206.042 0.01977483 0.007508533 0.04459288
     mean.ks iter
1 0.05623686   NA
2 0.01609467   NA
```

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

```
$unw
              tx.mn  tx.sd   ct.mn  ct.sd std.eff.sz   stat     p    ks
EJF_PER      25.870  8.988  29.579  8.518     -0.413 -9.536 0.000 0.173
female        0.287  0.453   0.202  0.402      0.187  4.414 0.000 0.085
AGE          63.972 10.580  63.328 11.021      0.061  1.365 0.172 0.034
white         0.875  0.331   0.844  0.363      0.094  2.063 0.039 0.031
BMI          27.242  5.582  27.122  5.072      0.022  0.504 0.614 0.036
CREAT         1.306  0.379   1.287  0.368      0.051  1.176 0.240 0.052
HEARTRTE     78.455 12.476  78.269 12.859      0.015  0.336 0.737 0.030
SYSBP       125.436 19.742 125.762 19.672     -0.017 -0.376 0.707 0.015
DIABP        75.264 12.255  75.141 10.963      0.010  0.237 0.813 0.017
ANGINA        0.267  0.443   0.267  0.442      0.001  0.025 0.980 0.000
DIABETES      0.246  0.431   0.295  0.456     -0.113 -2.512 0.012 0.049
HYPERTEN      0.483  0.500   0.461  0.499      0.045  1.017 0.309 0.022
DIURETK       0.099  0.298   0.073  0.260      0.087  2.059 0.040 0.026
ped_pres      0.200  0.400   0.200  0.400      0.000 -0.007 0.994 0.000
ped_past      0.453  0.498   0.451  0.498      0.003  0.065 0.948 0.001
rales_pres    0.162  0.369   0.168  0.374     -0.018 -0.404 0.686 0.007
rales_past    0.654  0.476   0.631  0.483      0.047  1.069 0.285 0.022
ps            0.363  0.121   0.301  0.108      0.511 12.032 0.000 0.234
linps        -0.602  0.555  -0.895  0.528      0.528 12.181 0.000 0.234
           ks.pval
EJF_PER      0.000
female       0.001
AGE          0.590
white        0.689
BMI          0.516
CREAT        0.122
HEARTRTE     0.729
SYSBP        1.000
DIABP        0.997
ANGINA       1.000
DIABETES     0.169
HYPERTEN     0.952
DIURETK      0.866
ped_pres     1.000
ped_past     1.000
```

```
rales_pres   1.000
rales_past   0.950
ps           0.000
linps        0.000

[[2]]
             tx.mn  tx.sd   ct.mn  ct.sd std.eff.sz   stat     p    ks
EJF_PER     25.870  8.988  25.989  8.554     -0.013 -0.288 0.773 0.036
female       0.287  0.453   0.289  0.453     -0.004 -0.085 0.933 0.002
AGE         63.972 10.580  64.045 10.978     -0.007 -0.146 0.884 0.029
white        0.875  0.331   0.876  0.330     -0.002 -0.045 0.964 0.001
BMI         27.242  5.582  27.200  5.021      0.008  0.171 0.864 0.045
CREAT        1.306  0.379   1.306  0.378     -0.001 -0.023 0.982 0.031
HEARTRTE    78.455 12.476  78.479 12.746     -0.002 -0.041 0.968 0.024
SYSBP      125.436 19.742 125.190 19.366      0.012  0.273 0.785 0.025
DIABP       75.264 12.255  75.216 11.006      0.004  0.088 0.930 0.018
ANGINA       0.267  0.443   0.276  0.447     -0.020 -0.424 0.671 0.009
DIABETES     0.246  0.431   0.248  0.432     -0.004 -0.078 0.937 0.002
HYPERTEN     0.483  0.500   0.485  0.500     -0.004 -0.085 0.932 0.002
DIURETK      0.099  0.298   0.097  0.296      0.007  0.137 0.891 0.002
ped_pres     0.200  0.400   0.204  0.403     -0.010 -0.214 0.830 0.004
ped_past     0.453  0.498   0.461  0.499     -0.016 -0.355 0.723 0.008
rales_pres   0.162  0.369   0.161  0.368      0.001  0.020 0.984 0.000
rales_past   0.654  0.476   0.656  0.475     -0.004 -0.080 0.936 0.002
ps           0.363  0.121   0.361  0.118      0.014  0.281 0.779 0.033
linps       -0.602  0.555  -0.608  0.539      0.011  0.228 0.820 0.033
           ks.pval
EJF_PER      0.569
female       1.000
AGE          0.805
white        1.000
BMI          0.301
CREAT        0.731
HEARTRTE     0.935
SYSBP        0.928
DIABP        0.997
ANGINA       1.000
DIABETES     1.000
HYPERTEN     1.000
DIURETK      1.000
ped_pres     1.000
ped_past     1.000
rales_pres   1.000
rales_past   1.000
ps           0.673
```

```
linps        0.673
```

The `std.eff.sz` shows the standardized difference, but as a proportion, rather than as a percentage. We'll create a data frame (tibble) so we can plot the data more easily.

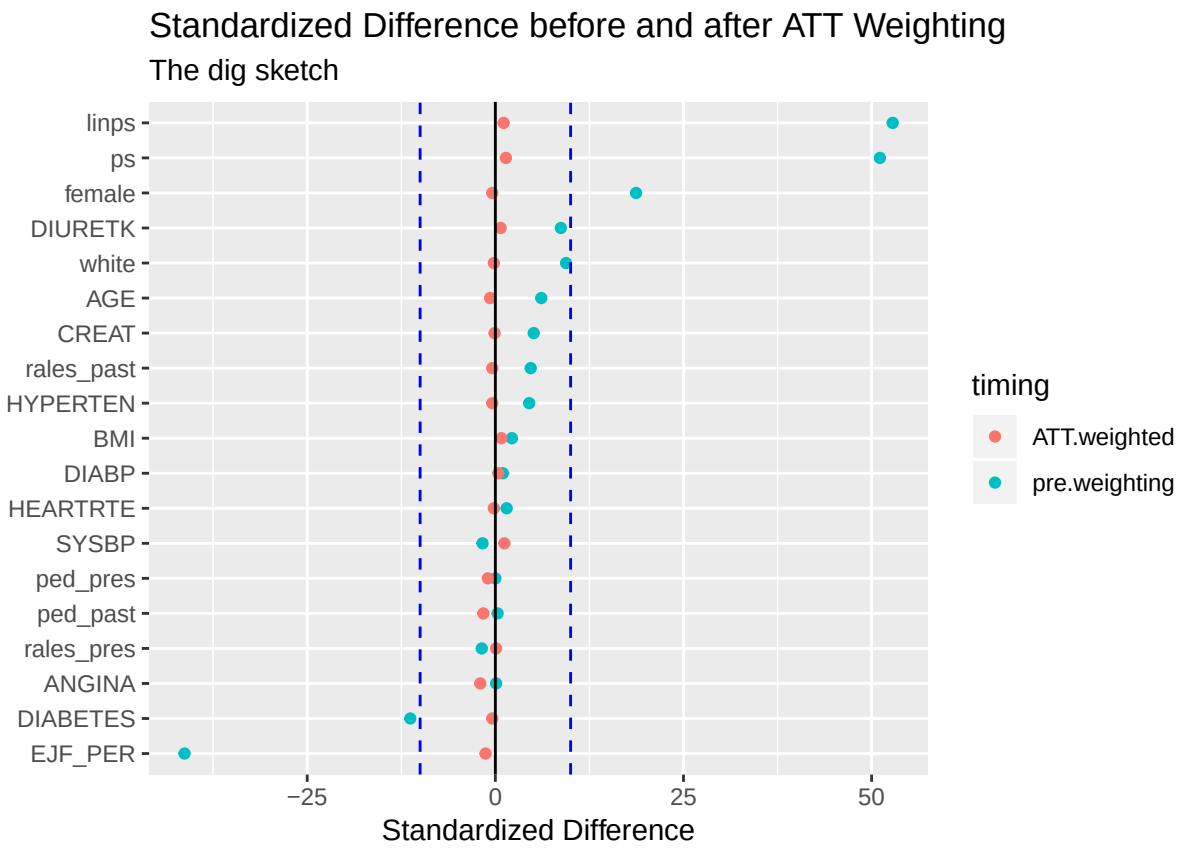
```
bal.before.wts1 <- bal.table(bal.wts1)[1]
bal.after.wts1 <- bal.table(bal.wts1)[2]

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)
```

```
Warning: `data_frame()` is deprecated, use `tibble()`.
This warning is displayed once per session.
```

```
balance.att.weights <- gather(balance.att.weights, timing, szd, 2:3)
```

OK - here is the plot of standardized differences before and 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",
         subtitle = "The dig sketch")
```



Looks great.

9.1.2 Rubin's Rule 1

```
balance.att.weights %>% filter(names == "linps")
```

```
# A tibble: 2 x 3
  names timing          szd
  <chr> <chr>         <dbl>
1 linps pre.weighting 52.8
2 linps ATT.weighted   1.10
```

The standardized difference of the linear propensity score is down (after weighting) from 52.8% to 1.1%. Excellent.

9.1.3 Rubin's Rule 2

We can read off the standard deviations within the treated and control groups. We can then square each, to get the relevant variances, then take the ratio of those variances.

```
bal.before.wts1$unw %>% select(tx.sd, ct.sd) %>% tail(1)
```

```
      tx.sd ct.sd
linps 0.555 0.528
```

Before weighting, we had a variance ratio of 0.555^2 / 0.528^2 = 1.105.

```
bal.after.wts1[[1]] %>% select(tx.sd, ct.sd) %>% tail(1)
```

```
      tx.sd ct.sd
linps 0.555 0.539
```

After weighting, the variance ratio us 0.555^2 / 0.539^2 = 1.060. Even better. We're well within the (4/5, 5/4) interval.

9.2 Build the Outcome Model using the weights

```
dig.design <- svydesign(ids=~1, weights=~wts1, data=dig)

wtd_model <- svyglm(HOSP ~ badNYHA, design=dig.design,
                    family=quasibinomial())
summary(wtd_model)
```

```
Call:
svyglm(formula = HOSP ~ badNYHA, design = dig.design, family = quasibinomial())

Survey design:
svydesign(ids = ~1, weights = ~wts1, data = dig)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.54723    0.05928   9.231   <2e-16 ***
badNYHA      0.39593    0.10020   3.951    8e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 1.000422)

Number of Fisher Scoring iterations: 4
```

```
exp(summary(wtd_model)$coef)
```

```
            Estimate Std. Error     t value Pr(>|t|)
(Intercept) 1.728452   1.061072 10210.95980  1.00000
badNYHA     1.485766   1.105391    52.00995  1.00008
```

```
exp(confint(wtd_model))
```

```
                2.5 %   97.5 %
(Intercept) 1.538857 1.941407
badNYHA     1.220845 1.808173
```

After weighting our odds ratio estimate is 1.49, with 95% CI (1.22, 1.81)

10 Task 5. Comparison of our Results

Approach	Odds Ratio	95% Confidence Interval
Unadjusted	1.52	(1.26, 1.83)
Matching	1.51	(1.22, 1.88)
Weighting	1.49	(1.22, 1.81)

The impact of the propensity score matching or weighting is pretty modest here in terms of this hospitalization outcome. Both the matching and the weighting do an excellent job of attending to the imbalances we see in the covariates for the unadjusted approach.