500 Homeworks 3 and 4 Answer Sketch

Thomas E. Love

Version: 2020-02-27.

Contents

1 Homework 3 Tasks **1**

1.1 Task 1 5
1.2 Task 2 7
1.3 Task 3 9
1.4 Task 4 12
1.5 Task 5 13
1.6 Task 6 17
1.7 Task 7 25

2 Homework 4 Tasks **27**

2.1 Task 1. 27
2.2 Task 2. 28
2.3 Task 3. 28
2.4 Task 4. 31
2.5 Task 5. 32
2.6 Task 6. 35
2.7 Task 7. 37
2.8 Task 8. 39

1 Homework 3 Tasks

1.0.1 Preliminaries

```r
knitr::opts_chunk$set(comment=NA)
```

```r
library(Hmisc)
library(tableone)
library(Matching)
library(twang)
library(cobalt)
library(broom)
library(survival)
library(arm)
library(tidyverse)
```

```r
canc3 <- read.csv("canc3.csv") %>% tbl_df
```

1.0.2 The Data

The `canc3.csv` data file is available at our Assignments page.

We have completed the data collection in a study of 400 subjects with cancer, where 150 have received an intervention, while the remaining 250 received usual care control. The primary aims of the study are to learn about the impact of the intervention on patient survival and whether or not the patient enters hospice.

1.0.3 The Codebook

The data file includes 400 observations, on 12 variables. All values are measured at baseline except for the two outcomes: `alive` and `hospice`.

Variable	Description	Notes
`id`	Study ID #	1-250 control, 251-400 intervention
`treated`	Treatment status	1 = intervention, 0 = control
`age`	Age at study entry	range is 34-93 years
`female`	Sex	1 = female ($n = 258$), 0 = male ($n = 142$)
`race`	Race	1 = caucasian/white ($n = 317$), 0 = not ($n = 83$)
`married`	Marital status	1 = married ($n = 245$), 0 = not ($n = 155$)
`typeca`	Cancer type	1 = GI/colorectal ($n = 177$), 2 = Lung ($n = 129$), 3 = GYN ($n = 94$)
`stprob`	5-year survival probability	Model probability, based on type and stage of cancer (range: 0.01, 0.72)
`charlson`	Charlson Comorbidity index	Total score: higher indicates greater comorbidity (observed range: 0-7)
`ecog`	ECOG functional status	0 = fully active ($n = 155$), 1 = restricted re: physically strenuous activity ($n = 201$), 2 = ambulatory, can self-care, otherwise limited ($n = 31$), 3 = capable of only limited self-care ($n = 13$)
`alive`	Alive at study conclusion	1 = alive ($n = 245$), 0 = dead ($n = 155$)
`hospice`	Entered hospice during study	1 = hospice ($n = 143$), 0 = no hospice ($n = 257$)

Note: In the answer sketch, I plan to treat `ecog` and `typeca` as categorical and `charlson` as quantitative.

1.0.4 Data Management and Creation of New Formats

- For **binary** outcomes and treatments, we want both numerical (0/1) and factor (with meaningful names) versions, so that includes treatment status [in `canc3`, this is `treated`] or binary outcomes [in `canc3`, this includes `alive` and `hospice`]. For other binary variables (for instance, representing covariates), all we really need are the numeric (0/1) variables we already have, although I'll use a better name for `race`, so I can indicate what 1 means there.
- For **categorical variables with more than two categories**, we want factor (with meaningful names, especially for unordered categories) versions of the variable [in `canc3`, these are `typeca` and `ecog`], and we may also eventually need a series of numeric (0/1) indicators to represent the individual categories.
- For **quantitative** variables [in `canc3`, these will be `age`, `stprob` and `charlson` assuming that you, like me, are willing to treat `charlson` as quantitative], we just want the numerical representations we already have.

```
canc3
```

```
# A tibble: 400 x 12
   subject treated   age female  race married typeca stprob charlson  ecog alive
     <int>   <int> <int>  <int> <int>   <int>  <int>  <dbl>    <int> <int> <int>
 1       1       0    59      1     1       1      1   0.07        1     3     1
 2       2       0    58      1     1       1      3   0.16        0     0     1
 3       3       0    71      1     1       1      3   0.72        1     0     1
 4       4       0    64      0     1       1      2   0.16        1     2     1
 5       5       0    56      1     1       1      1   0.07        0     1     1
 6       6       0    79      0     1       1      1   0.09        0     0     1
 7       7       0    69      0     1       1      2   0.16        1     0     1
 8       8       0    62      0     1       1      1   0.07        0     0     1
 9       9       0    71      0     1       1      1   0.07        1     1     1
10      10       0    52      1     0       0      1   0.07        1     0     0
# ... with 390 more rows, and 1 more variable: hospice <int>
```

So, our primary cleanup task will be to create factor versions of five of the variables (specifically, `treated`, `alive` and `hospice` on the binary side and `typeca` and `ecog` on the multi-categorical side), and numeric indicator variables for the multi-categorical variables, while the remaining variables can stay as they are.

```
canc3.original <- canc3 # save original version in case of catastrophe

canc3 <- canc3 %>%
    mutate(treated_f = factor(treated, levels = c(0,1),
                              labels = c("Control", "Intervention")),
           treatment_group = fct_relevel(treated_f, "Intervention"),
           alive_f = factor(alive, levels = c(0,1),
                            labels = c("Dead", "Alive")),
           hospice_f = factor(hospice, levels = c(0, 1),
                              labels = c("No Hospice", "Hospice")),
           caucasian = race,
           typeca_GI = as.numeric(typeca == 1),
           typeca_Lung = as.numeric(typeca == 2),
           typeca_GYN = as.numeric(typeca == 3),
           ecog = factor(ecog),
           ecog_0 = as.numeric(ecog == 0),
           ecog_1 = as.numeric(ecog == 1),
           ecog_2 = as.numeric(ecog == 2),
           ecog_3 = as.numeric(ecog == 3),
           typeca = factor(typeca, levels = c(1, 2, 3),
                           labels = c("GI/colorectal", "Lung", "GYN"))
           )
```

1.0.5 Table 1 to Check Results

I'll build a simple Table 1, without *p* values, to look over the results. We could easily leave off the two outcomes, but I'll keep them in for now.

```
varlist = c("age", "female", "caucasian", "married", "typeca", "ecog",
            "alive_f", "hospice_f")
factorlist = c("female", "caucasian", "married", "typeca", "ecog",
            "alive_f", "hospice_f")
CreateTableOne(vars = varlist, strata = "treatment_group",
               data = canc3, factorVars = factorlist, test = FALSE)
```

```
                         Stratified by treatment_group
                          Intervention  Control
  n                         150           250
  age (mean (SD))         63.76 (10.87) 62.56 (11.26)
  female = 1 (%)             93 (62.0)    165 (66.0)
  caucasian = 1 (%)         109 (72.7)    208 (83.2)
  married = 1 (%)            83 (55.3)    162 (64.8)
  typeca (%)
     GI/colorectal           68 (45.3)    109 (43.6)
     Lung                    64 (42.7)     65 (26.0)
     GYN                     18 (12.0)     76 (30.4)
  ecog (%)
     0                       52 (34.7)    103 (41.2)
     1                       85 (56.7)    116 (46.4)
```

```
    2                           9 ( 6.0)     22 ( 8.8)
    3                           4 ( 2.7)      9 ( 3.6)
  alive_f = Alive (%)          82 (54.7)    163 (65.2)
  hospice_f = Hospice (%)      62 (41.3)     81 (32.4)
```

```
rm(varlist, factorlist)
```

Everything looks reasonable to me.

1.1 Task 1

> Ignoring the covariate information, provide an appropriate (unadjusted) estimate (with point estimate and 95% confidence interval) of the effect of the intervention on each of the two binary outcomes; first survival, and then hospice entry. Be sure to describe the effect in English sentences, so that both the direction and magnitude are clear, and also be sure to specify the method you used in generating your estimates.

1.1.1 Unadjusted Logistic Regression Model for Survival

We can obtain the odds ratio estimate uses logistic regression:

```
unadj.alive <- glm(alive ~ treated_f, data=canc3, family=binomial)

display(unadj.alive)
```

```
glm(formula = alive ~ treated_f, family = binomial, data = canc3)
                      coef.est coef.se
(Intercept)            0.63     0.13  
treated_fIntervention -0.44     0.21  
---
  n = 400, k = 2
  residual deviance = 529.7, null deviance = 534.1 (difference = 4.4)
```

```
exp(coef(unadj.alive)) # odds ratio estimate
```

```
          (Intercept) treated_fIntervention 
            1.8735632             0.6436305 
```

```
exp(confint(unadj.alive)) # 95% CI for odds ratio
```

```
Waiting for profiling to be done...

                          2.5 %    97.5 %
(Intercept)           1.4487048 2.4397580
treated_fIntervention 0.4252576 0.9733951
```

We have an odds ratio estimate for the intervention's impact on survival of **0.64** with 95% CI of (0.43, 0.97). This is just barely statistically significant at the 5% level.

1.1.2 Unadjusted logistic regression model for the hospice outcome

```
unadj.hospice <- glm(hospice ~ treated_f, data=canc3, family=binomial)

display(unadj.hospice)
```

```
glm(formula = hospice ~ treated_f, family = binomial, data = canc3)
                      coef.est coef.se
(Intercept)           -0.74     0.14  
treated_fIntervention  0.39     0.21  
---
  n = 400, k = 2
  residual deviance = 518.3, null deviance = 521.6 (difference = 3.2)
```

```
exp(coef(unadj.hospice)) # odds ratio estimate
```

```
          (Intercept) treated_fIntervention 
            0.4792899             1.4699776 
```

```
exp(confint(unadj.hospice)) # 95% CI for odds ratio
```

```
Waiting for profiling to be done...

                          2.5 %    97.5 %
(Intercept)           0.3660984 0.6223179
treated_fIntervention 0.9658958 2.2362360
```

Either way, we have an odds ratio estimate for the intervention's impact on `hospice` of 1.47 with 95% CI of (0.97, 2.24). So the odds of going to hospice are 1.47 times as large for intervention patients as compared to control patients. The confidence interval does include 1, so we cannot claim statistical significance (at a 5% significance level) in this unadjusted analysis for `hospice`.

1.1.3 Final Answers for Task 1

Unadjusted Analyses Comparing the Intervention Group to the Control Group…

Outcome	Odds Ratio	95% CI
`alive`	0.64	(0.43, 0.97)
`hospice`	1.47	(0.97, 2.24)

1.2 Task 2

Next, fit a propensity score model to the data, using the eight pieces of covariate information, including age, gender, race, marital status, cancer type (which must be treated in R as a factor rather than just a continuous predictor) the model survival probability, Charlson index and ECOG. Do not include interactions between terms.

1.2.1 Fitting the Model and Saving Raw and Linear Propensity Scores

```
psmodel <- glm(treated_f ~ age + female + caucasian + married + typeca +
                 stprob + charlson + ecog, family=binomial, data=canc3)
canc3$ps <- psmodel$fitted
canc3$linps <- psmodel$linear.predictors
```

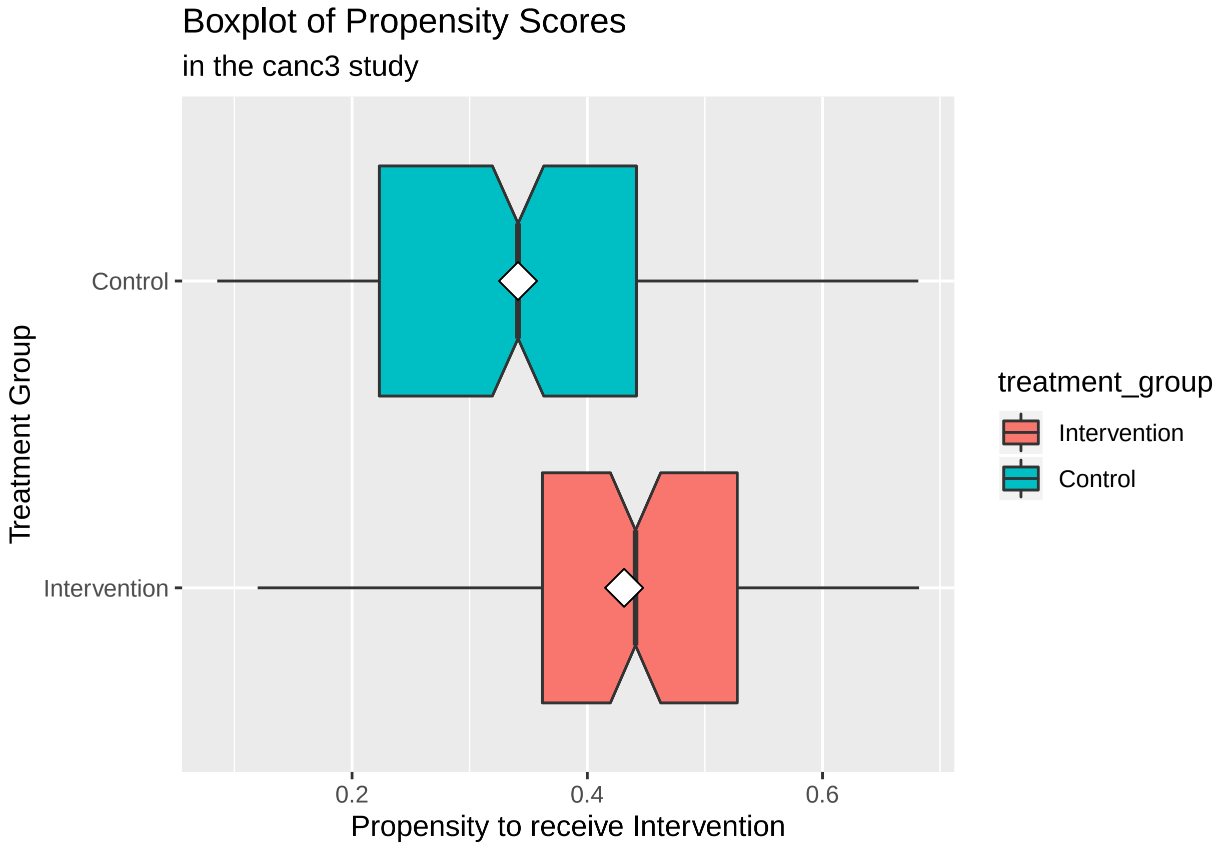
1.2.2 Looking at the Overlap Numerically

```
canc3 %>%
  group_by(treated_f) %>%
  summarise(mean.ps = mean(ps), sd.ps = sd(ps), median.ps = median(ps),
            mean.linps = mean(linps), sd.linps = sd(linps))
```

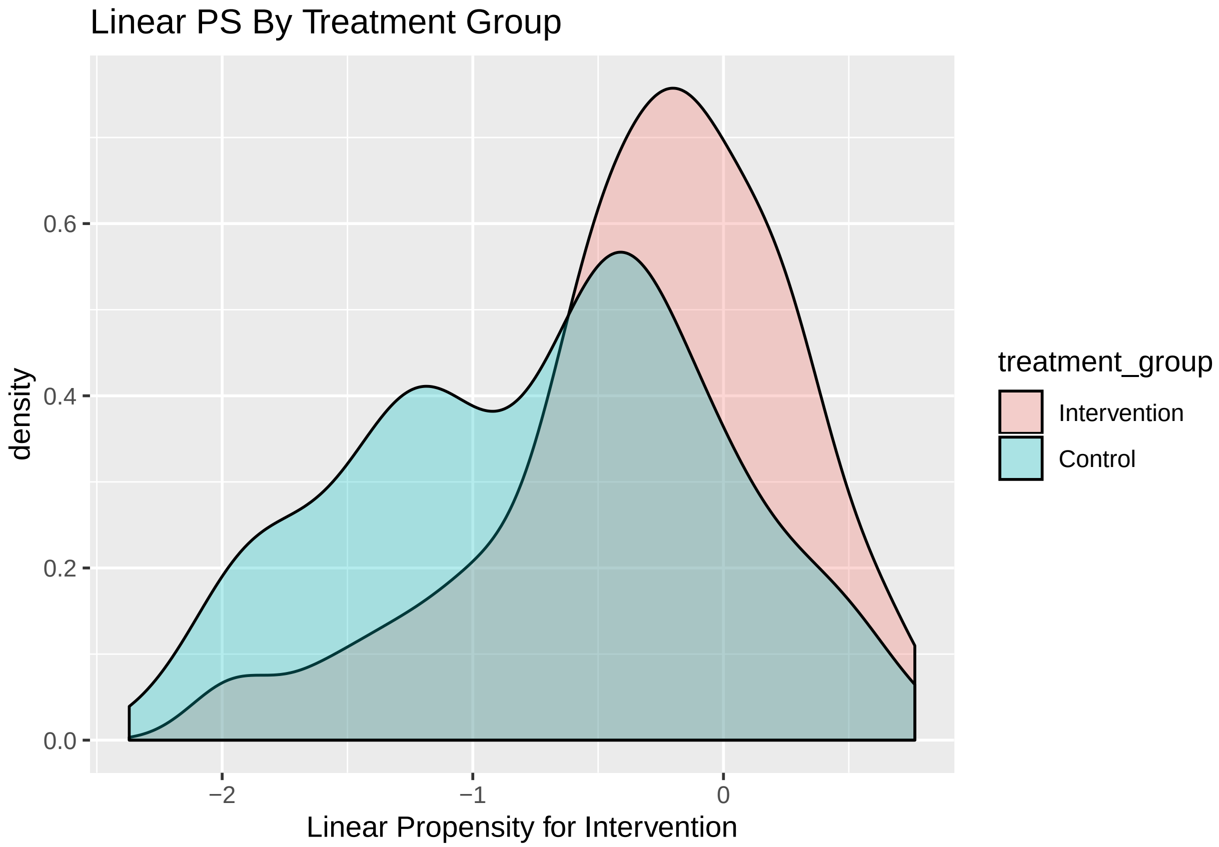
```
# A tibble: 2 x 6
  treated_f    mean.ps sd.ps median.ps mean.linps sd.linps
  <fct>          <dbl> <dbl>     <dbl>      <dbl>    <dbl>
1 Control        0.341 0.145     0.341     -0.735    0.705
2 Intervention   0.431 0.129     0.441     -0.308    0.585
```

1.2.3 Looking at the Overlap Graphically

```r
# requires ggplot2 library
ggplot(canc3, aes(x = treatment_group, y = ps, fill = treatment_group)) +
  geom_boxplot(notch=TRUE) +
  stat_summary(fun.y="mean", geom="point", shape=23, size = 5, fill = "white") +
  coord_flip() +
    labs(x = "Treatment Group",
         y = "Propensity to receive Intervention",
         title = "Boxplot of Propensity Scores",
         subtitle = "in the canc3 study")
```



```
ggplot(canc3, aes(x=linps, fill=treatment_group)) +
  geom_density(alpha=0.3) +
  labs(x="Linear Propensity for Intervention",
       title="Linear PS By Treatment Group")
```



1.3 Task 3

Evaluate Rubin’s Rule 1 and Rubin’s Rule 2 for the data taken as a whole. What can you conclude about the balance across the two exposure groups prior to using the propensity score? What do these results suggest about your model in Task 1?

1.3.1 Rubin’s Rule 1

First, the absolute value of the standardized difference of the linear propensity score, comparing the intervention group to the control group, should be close to 0, ideally below 10%, and in any case less than 50%. If so, we may move on to Rubin’s Rule 2.

To evaluate this here, I’ll use :

```
rubin1.unadj <- with(canc3,
            abs(100*(mean(linps[treated==1])-mean(linps[treated==0])) /
                           sd(linps)))
rubin1.unadj
```

```
[1] 61.59673
```

Here, with a value of 62%, we cannott justify simply running an unadjusted regression model, be it a linear, logistic or Cox model. We have substantial observed selection bias, and need to further adjust for this using our propensity score before trusting that our comparisons will be fair. But we'll check Rule 2 anyway, as instructed.

1.3.2 Rubin's Rule 2

Second, the ratio of the variance of the linear propensity score in the intervention group to the variance of the linear propensity score in the control group should be close to 1, ideally between 4/5 and 5/4, but certainly between 1/2 and 2. If so, we may move on to Rule 3.

To evaluate this here, I'll use:

```
rubin2.unadj <-with(canc3,
                    var(linps[treated==1]) / var(linps[treated==0]))
rubin2.unadj
```

```
[1] 0.6883501
```

Again, this is the ratio of variances of the linear propensity score comparing intervention patients to control patients. We want this value to be close to 1, and certainly between 0.5 and 2. In this case, we pass Rule 2, though not by much.

1.3.3 Rubin's Rule 3 (not part of the assignment)

I didn't ask you to do this, but finding the Rubin's Rule 3 results prior to adjustment looks like this:

```
## General function rubin3 to help calculate Rubin's Rule 3
decim <- function(x, k) format(round(x, k), nsmall=k)
rubin3 <- function(data, covlist, linps) {
  covlist2 <- as.matrix(covlist)
  res <- NA
  for(i in 1:ncol(covlist2)) {
    cov <- as.numeric(covlist2[,i])
    num <- var(resid(lm(cov ~ data$linps))[data$exposure == 1])
    den <- var(resid(lm(cov ~ data$linps))[data$exposure == 0])
    res[i] <- decim(num/den, 3)
  }
  final <- data_frame(name = names(covlist), resid.var.ratio = as.numeric(res))
  return(final)
}
```

Now, then, applying the rule to our sample prior to propensity score adjustment, we get …

```
cov.sub <- canc3 %>% select(age, female, caucasian, married,
                            stprob, charlson, typeca_GI,
                            typeca_Lung, typeca_GYN, ecog_0,
                            ecog_1, ecog_2, ecog_3)

canc3$exposure <- canc3$treated

rubin3.unadj <- rubin3(data=canc3, covlist = cov.sub, linps = linps)
```

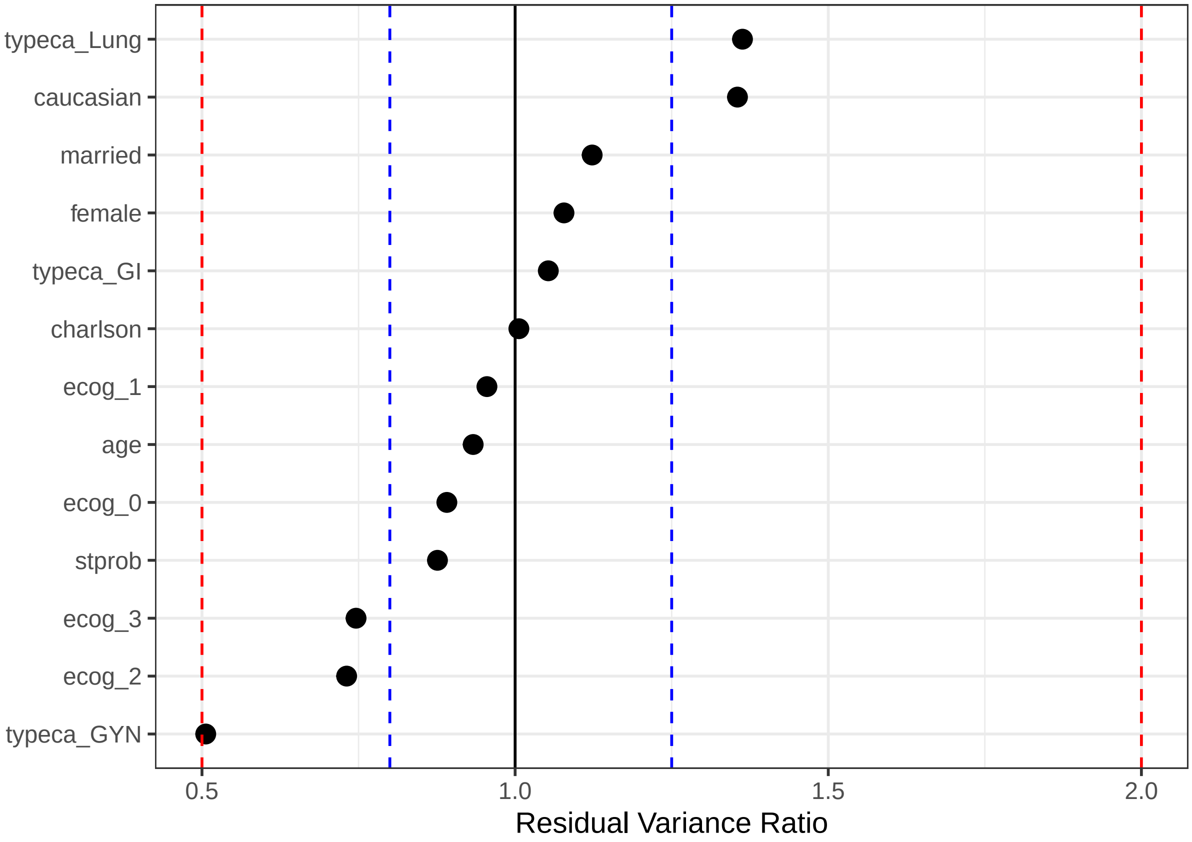
```
Warning: `data_frame()` is deprecated, use `tibble()`.
This warning is displayed once per session.
```

```
rubin3.unadj
```

```
# A tibble: 13 x 2
   name        resid.var.ratio
   <chr>                 <dbl>
 1 age                   0.933
 2 female                1.08
 3 caucasian             1.36
 4 married               1.12
 5 stprob                0.876
 6 charlson              1.01
 7 typeca_GI             1.05
 8 typeca_Lung           1.36
 9 typeca_GYN            0.506
10 ecog_0                0.891
11 ecog_1                0.955
12 ecog_2                0.731
13 ecog_3                0.746
```

Some of these covariates look to have residual variance ratios near 1, while others are further away, but all are within the (0.5, 2.0) range. So we would pass Rule 3 here, although we would clearly like to see some covariates (`typeca_GYN`, in particular) with ratios closer to 1. Here's a dotplot.

```
ggplot(rubin3.unadj, aes(x = resid.var.ratio, y = reorder(name, resid.var.ratio))) +
    geom_point(size = 3) +
    theme_bw() +
    xlim(0.5, 2.0) +
    geom_vline(xintercept = 1) +
    geom_vline(xintercept = c(4/5,5/4), lty = "dashed", col = "blue") +
    geom_vline(xintercept = c(0.5,2), lty = "dashed", col = "red") +
  labs(x = "Residual Variance Ratio", y = "")
```



1.4 Task 4

Use direct adjustment for the (logit of) the propensity score in a logistic regression model for the `hospice` outcome to evaluate the intervention's effect on hospice entry, developing a point estimate (this should be an odds ratio) and a 95% confidence interval.

1.4.1 Fitting the Model

Recall that the unadjusted logistic regression model for the `hospice` outcome was:

```
unadj.hospice <- glm(hospice ~ treated, data=canc3, family=binomial)
```

This led to an unadjusted odds ratio estimate for the intervention's effect on `hospice` of 1.47 with 95% CI of (0.97, 2.24).

Our new model will add the linear propensity score on the right hand side…

```
adj.hospice <- glm(hospice ~ treated + linps, data=canc3, family=binomial)
display(adj.hospice)
```

```
glm(formula = hospice ~ treated + linps, family = binomial, data = canc3)
            coef.est coef.se
(Intercept) -0.20     0.17
treated      0.07     0.23
linps        0.81     0.18
---
  n = 400, k = 3
  residual deviance = 495.0, null deviance = 521.6 (difference = 26.6)
```

```
exp(coef(adj.hospice))
```

```
(Intercept)     treated       linps
  0.8227802   1.0736976   2.2530282
```

```
exp(confint(adj.hospice))
```

```
Waiting for profiling to be done...

                2.5 %   97.5 %
(Intercept) 0.5831353 1.159700
treated     0.6841785 1.676902
linps       1.6072543 3.211896
```

So, after direct adjustment for the linear propensity score, the odds ratio estimate for the impact of the intervention on hospice is 1.07 with 95% CI of (0.68, 1.68). In other words, we still see no significant treatment effect on the hospice outcome.

1.4.2 Our results so far, for the `hospice` outcome

Estimating the **intervention effect** on the `hospice` outcome…

Analytic Approach	Odds Ratio	95% CI
Unadjusted	1.47	(0.97, 2.24)
Direct PS adjustment	1.07	(0.68, 1.68)

1.5 Task 5

> Use subclassification by quintile of the propensity score to estimate the effect of the intervention on hospice entry. Specifically, first report an odds ratio estimate (and confidence interval) for each individual stratum, then demonstrate a pooled estimate across all five strata, being sure to indicate whether you believe pooling

to be appropriate in this setting.

1.5.1 Subclassifying by Propensity Score Quintile

```
## cut2 requires the Hmisc library
canc3$stratum <- cut2(canc3$ps, g=5)
canc3$quintile <- factor(canc3$stratum, labels=1:5)

table(canc3$stratum, canc3$quintile) ## sanity check
```

```
                  1  2  3  4  5
  [0.0854,0.229) 80  0  0  0  0
  [0.2294,0.349)  0 80  0  0  0
  [0.3493,0.419)  0  0 80  0  0
  [0.4192,0.505)  0  0  0 80  0
  [0.5046,0.682]  0  0  0  0 80
```

```
## semi-fancy summaries of PS by stratum using dplyr
canc3 %>% group_by(stratum) %>%
  summarise(n = length(ps), mean = mean(ps), sd = sd(ps),
            min=min(ps), max=max(ps))
```

```
# A tibble: 5 x 6
  stratum            n  mean     sd    min   max
  <fct>          <int> <dbl>  <dbl>  <dbl> <dbl>
1 [0.0854,0.229)    80 0.167 0.0382 0.0854 0.229
2 [0.2294,0.349)    80 0.284 0.0388 0.229  0.348
3 [0.3493,0.419)    80 0.387 0.0188 0.349  0.419
4 [0.4192,0.505)    80 0.461 0.0257 0.419  0.505
5 [0.5046,0.682]    80 0.576 0.0482 0.505  0.682
```

Next, I'll create a separate subset of the data for each of the five quintiles.

```
quin1 <- subset(canc3, quintile==1)
quin2 <- subset(canc3, quintile==2)
quin3 <- subset(canc3, quintile==3)
quin4 <- subset(canc3, quintile==4)
quin5 <- subset(canc3, quintile==5)
```

1.5.2 Fitting Logistic Regression Models

Given that we want an odds ratio estimate, we can focus on logistic regression modeling.

```
quin1.hospice <- glm(hospice ~ treated_f, data=quin1, family=binomial)
quin2.hospice <- glm(hospice ~ treated_f, data=quin2, family=binomial)
quin3.hospice <- glm(hospice ~ treated_f, data=quin3, family=binomial)
quin4.hospice <- glm(hospice ~ treated_f, data=quin4, family=binomial)
quin5.hospice <- glm(hospice ~ treated_f, data=quin5, family=binomial)
```

Let's start by looking closely at Quintile 1

```
display(quin1.hospice)
```

```
glm(formula = hospice ~ treated_f, family = binomial, data = quin1)
                      coef.est coef.se
(Intercept)           -1.33     0.30  
treated_fIntervention -0.37     0.83  
---
  n = 80, k = 2
  residual deviance = 79.8, null deviance = 80.1 (difference = 0.2)
```

```
exp(coef(quin1.hospice)[2]) # odds ratio estimate: Quintile 1
```

```
treated_fIntervention 
            0.6883117 
```

```
exp(confint(quin1.hospice)[c(2,4)]) # 95% CI for OR in Quintile 1
```

```
[1] 0.09912283 2.96273415
```

1.5.3 Quintile-Specific Logistic Regression Coefficients and Standard Errors

Here are the results for each Quintile…

```
coef(quin1.hospice)
```

```
          (Intercept) treated_fIntervention 
           -1.3312346            -0.3735135 
```

```
coef(quin2.hospice)
```

```
          (Intercept) treated_fIntervention 
           -1.6094379             0.7621401 
```

```
coef(quin3.hospice)
```

```
          (Intercept) treated_fIntervention 
           -0.3227734            -0.2237703 
```

```
coef(quin4.hospice)
```

```
          (Intercept) treated_fIntervention 
          0.3184537            -0.5095090
```

```r
coef(quin5.hospice)
```

```
          (Intercept) treated_fIntervention
           -0.4054651             0.5389965
```

Quintile	Coefficient = $log(\hat{OR})$	Associated Standard Error
1	-0.374	0.825
2	0.762	0.598
3	-0.224	0.475
4	-0.51	0.452
5	0.539	0.456

1.5.4 Odds Ratio Estimates and 95% CI within Quintiles

Quintile	Odds Ratio	95% CI
1	0.69	(0.1, 2.96)
2	2.14	(0.64, 6.87)
3	0.8	(0.31, 2.01)
4	0.6	(0.24, 1.45)
5	1.71	(0.71, 4.26)

Pooling doesn't look like a good idea here. The individual odds ratios vary substantially from quintile to quintile, even though none are statistically significantly different from 1.

1.5.5 Producing a Pooled Estimate

That said, I asked you to produce a pooled estimate anyway. To do so, we first estimate the pooled log odds ratio, across the five quintiles:

```r
## Next, we find the mean of the five
## quintile-specific estimated logistic regression coefficients
est.st <- (coef(quin1.hospice)[2] + coef(quin2.hospice)[2] +
             coef(quin3.hospice)[2] + coef(quin4.hospice)[2] +
             coef(quin5.hospice)[2]) / 5
round(est.st,3) ## this is the estimated log odds ratio
```

```
treated_fIntervention
                0.039
```

```r
## And we exponentiate this to get the overall odds ratio estimate
round(exp(est.st),3)
```

```
treated_fIntervention
                 1.04
```

To get the combined standard error estimate, we have:

```
## Pooling the quintile-specific standard errors
se.q1 <- summary(quin1.hospice)$coefficients[2,2]
se.q2 <- summary(quin2.hospice)$coefficients[2,2]
se.q3 <- summary(quin3.hospice)$coefficients[2,2]
se.q4 <- summary(quin4.hospice)$coefficients[2,2]
se.q5 <- summary(quin5.hospice)$coefficients[2,2]
se.st <- sqrt((se.q1^2 + se.q2^2 + se.q3^2 + se.q4^2 + se.q5^2)*(1/25))
```

Of course, this standard error is also on the log odds ratio scale.

So the 95% Confidence Interval for the effect of the intervention on hospice (as an Odds Ratio) requires us to exponentiate again…

```
subclass.res <- c(exp(est.st), exp(est.st - 1.96*se.st), exp(est.st + 1.96*se.st))
names(subclass.res) <- c("Estimate", "Low 95% CI", "High 95% CI")
round(subclass.res,3)
```

```
   Estimate  Low 95% CI High 95% CI 
      1.040       0.626       1.727 
```

1.5.6 Our Results So Far, for the `hospice` Outcome

Estimating the **intervention effect** on the `hospice` outcome…

Analytic Approach	Odds Ratio	95% CI
Unadjusted	1.47	(0.97, 2.24)
Direct PS adjustment	1.07	(0.68, 1.68)
PS quintile subclassification	1.04	(0.63, 1.73)

1.6 Task 6

> In our first propensity score matching attempt with the `canc3` data, we'll apply a 1:1 match without replacement. Do the matching, and then evaluate the balance associated with this approach, as follows.

1.6.1 Do the matching

We'll do 1:1 greedy matching, without replacement.

```
## Use 1:1 greedy matching to match all treated to unique control patients
## on the linear propensity scores. We'll break ties at random, as well.

## requires Matching library

X <- psmodel$linear.predictors ## matching on the linear propensity score

Tr <- as.logical(canc3$treated)

set.seed(432)
# if we rerun Match, we want to get the same answer
# since we're breaking ties at random, we should set a seed
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..............  400
Original number of treated obs...............  150
Matched number of observations...............  150
Matched number of observations  (unweighted).  150
```

```
## Finally, we'll create a new data frame, containing only the matched sample
matches <- factor(rep(match1$index.treated, 2))
canc3.matchedsample <-
  cbind(matches, canc3[c(match1$index.control, match1$index.treated),])

## Sanity Check
table(canc3.matchedsample$treated_f)
```

1.6.1.1 Create Data Frame with Matched Sample After 1:1 Matching

```
     Control Intervention 
         150          150 
```

```
## should be 150 treated and 150 control patients

head(canc3.matchedsample,5)
```

```
  matches subject treated age female race married        typeca stprob charlson
1     251     215       0  67      0    0       1 GI/colorectal   0.02        0
2     252     191       0  51      0    0       0 GI/colorectal   0.18        0
3     253       6       0  79      0    1       1 GI/colorectal   0.09        0
4     254      40       0  73      0    1       1 GI/colorectal   0.04        0
5     255      44       0  52      1    1       1          Lung   0.16        6
  ecog alive hospice treated_f treatment_group alive_f  hospice_f caucasian
1    1     0       1   Control         Control    Dead    Hospice         0
2    0     1       0   Control         Control   Alive No Hospice         0
3    0     1       0   Control         Control   Alive No Hospice         1
4    1     0       1   Control         Control    Dead    Hospice         1
5    1     1       0   Control         Control   Alive No Hospice         1
  typeca_GI typeca_Lung typeca_GYN ecog_0 ecog_1 ecog_2 ecog_3        ps
1         1           0          0      0      1      0      0 0.4915717
2         1           0          0      1      0      0      0 0.4607890
3         1           0          0      1      0      0      0 0.3519827
4         1           0          0      0      1      0      0 0.4036331
5         0           1          0      0      1      0      0 0.5403399
        linps exposure        stratum quintile
1 -0.03371622        0 [0.4192,0.505)        4
2 -0.15716682        0 [0.4192,0.505)        4
3 -0.61033523        0 [0.3493,0.419)        3
4 -0.39034963        0 [0.3493,0.419)        3
5  0.16171127        0 [0.5046,0.682]        5
```

1.6.2 Task 6a.

> Evaluate the degree of covariate imbalance before and after propensity score matching for each of the eight covariates and for the (linear *and* raw) propensity score. Do so by plotting the standardized differences. Your plot should include standardized differences that identify the three cancer types (one remaining as baseline) individually, one each for any other covariates you treat as quantitative, and an appropriate set of indicators for any others you treat as categorical, plus one for the linear propensity score, and one for the raw propensity score.

```r
b <- bal.tab(match1, treated ~ age + female + caucasian +
                 married + typeca + stprob + charlson +
                 ecog + ps + linps, data = canc3,
             disp.v.ratio = TRUE, quick = FALSE, un = TRUE)

b
```

1.6.2.1 Performing Task 6a with cobalt

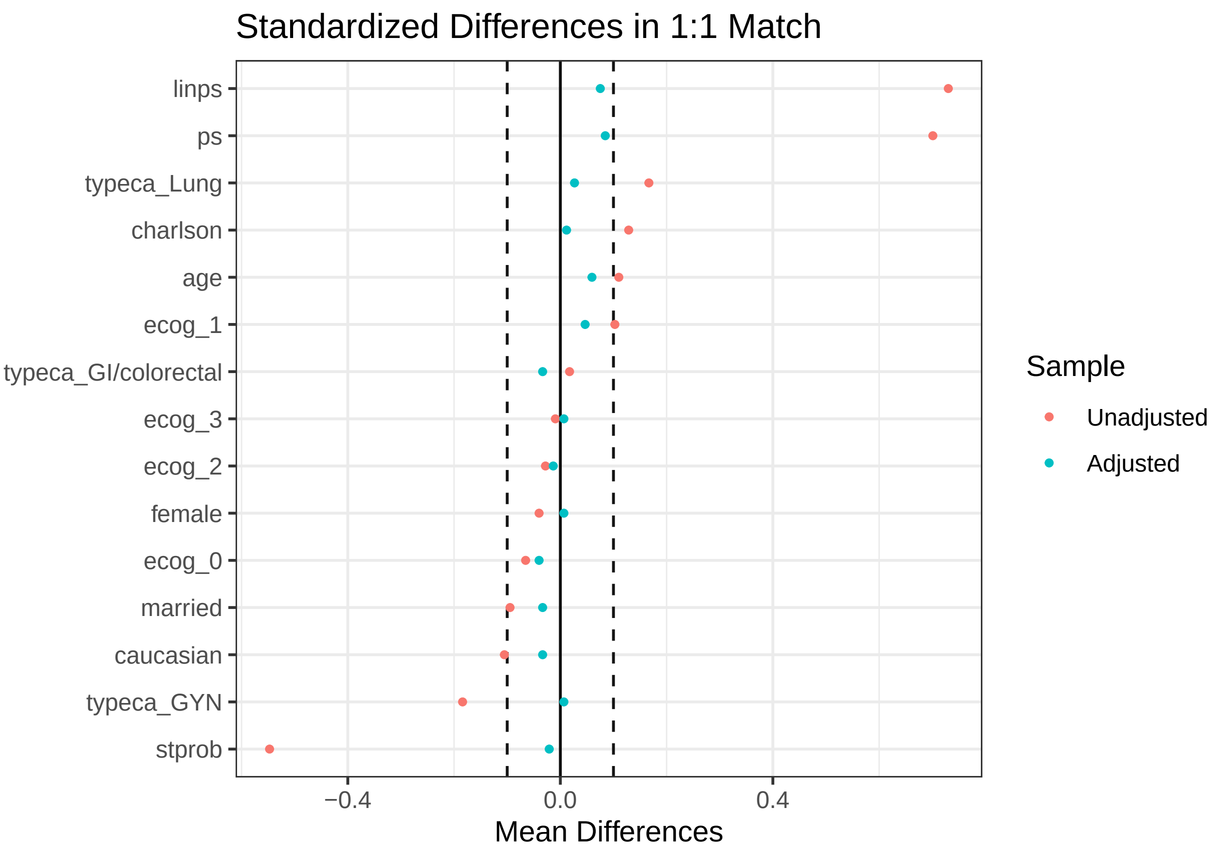
```
Balance Measures
                        Type Diff.Un V.Ratio.Un Diff.Adj V.Ratio.Adj
age                  Contin.  0.1101     0.9309   0.0595      0.9691
female                Binary -0.0400              0.0067
caucasian             Binary -0.1053             -0.0333
married               Binary -0.0947             -0.0333
typeca_GI/colorectal  Binary  0.0173             -0.0333
typeca_Lung           Binary  0.1667              0.0267
typeca_GYN            Binary -0.1840              0.0067
stprob               Contin. -0.5473     0.6889  -0.0209      1.1205
charlson             Contin.  0.1286     0.9968   0.0117      0.7609
ecog_0                Binary -0.0653             -0.0400
ecog_1                Binary  0.1027              0.0467
ecog_2                Binary -0.0280             -0.0133
ecog_3                Binary -0.0093              0.0067
ps                   Contin.  0.7011     0.7859   0.0846      1.0746
linps                Contin.  0.7302     0.6884   0.0752      1.0687

Sample sizes
          Control Treated
All           250     150
Matched       150     150
Unmatched     100       0
```

```r
p <- love.plot(b, threshold = .1, size = 1.5,
               var.order = "unadjusted",
               title = "Standardized Differences in 1:1 Match")
```

1.6.2.1.1 Love Plot of Standardized Differences, via cobalt

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

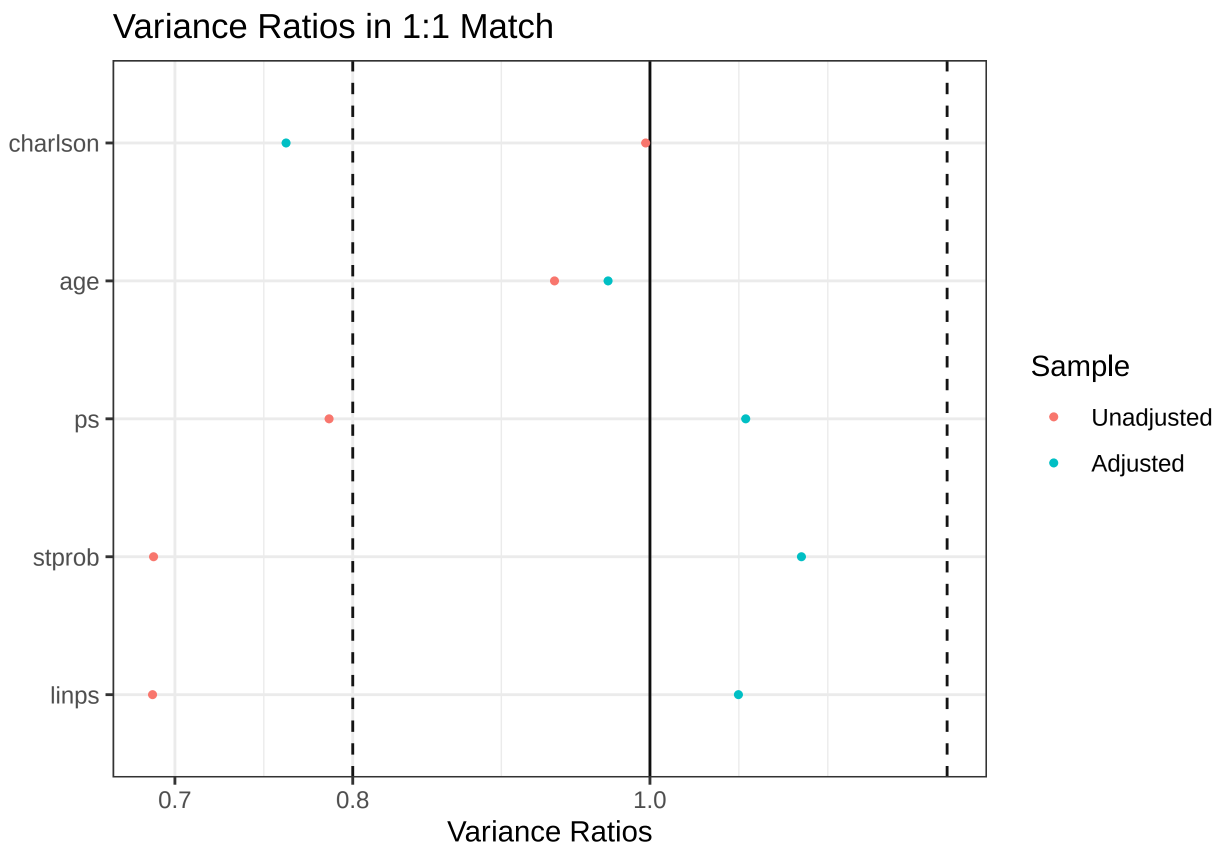
```r
p + theme_bw()
```



1.6.2.1.2 Plot of Variance Ratios, via `cobalt`

Note that by default in `cobalt`, this plot only compares variances for continuous predictors, and the linear and raw propensity scores.

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



1.6.3 Task 6b.

Evaluate the balance imposed by your 1:1 match via calculation of Rubin's Rule 1 and Rule 2 results, and comparing them to our results obtained prior to propensity adjustment in Task 3.

1.6.4 Evaluate the balance using Rubin's Rules after Matching

```
rubin1.match <- with(canc3.matchedsample,
      abs(100*(mean(linps[treated==1]) -
                 mean(linps[treated==0])) /
                    sd(linps)))
rubin1.match
```

```
[1] 7.652265
```

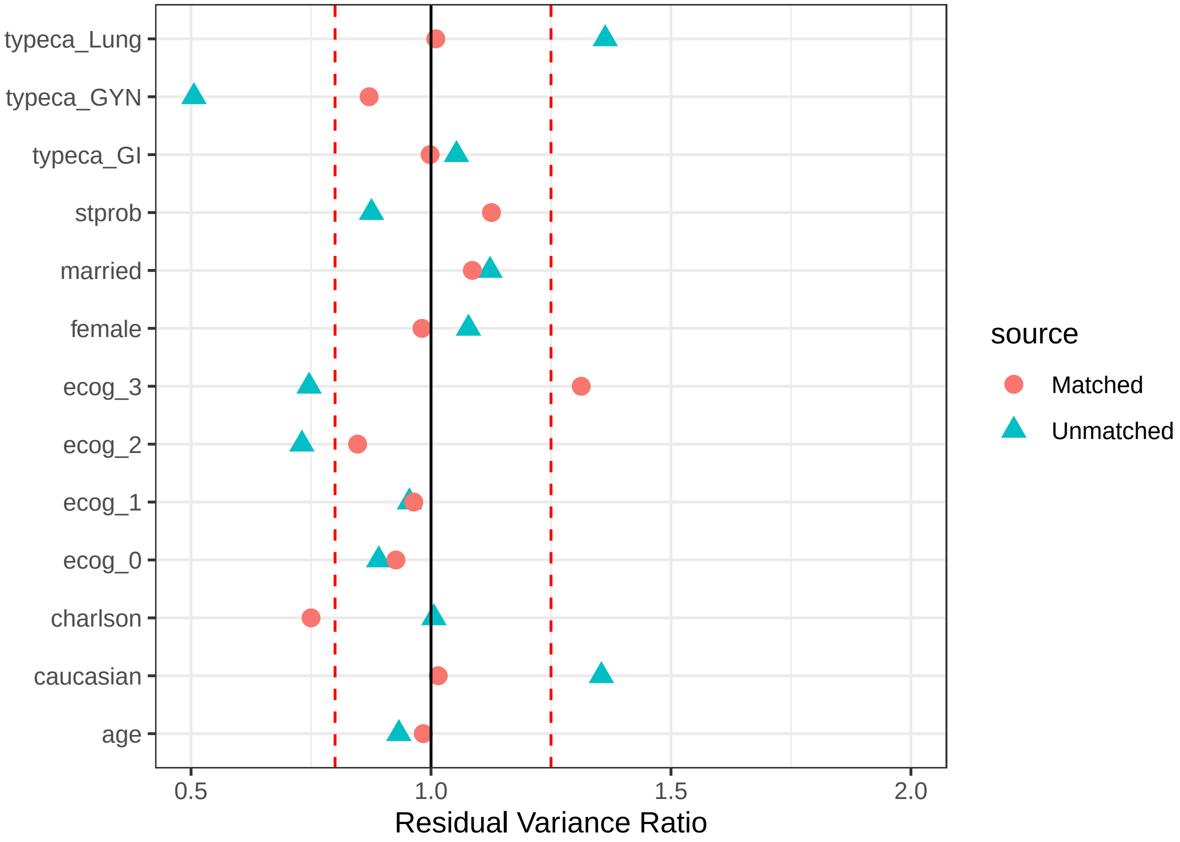
```
rubin2.match <-with(canc3.matchedsample,
      var(linps[treated==1])/
        var(linps[treated==0]))
rubin2.match
```

```
[1] 1.068696
```

```
cov.sub <- canc3.matchedsample %>% select(age, female, caucasian, married,
                            stprob, charlson, typeca_GI,
                            typeca_Lung, typeca_GYN, ecog_0,
                            ecog_1, ecog_2, ecog_3)


canc3.matchedsample$exposure <- canc3.matchedsample$treated

rubin3.match <- rubin3(data = canc3.matchedsample,
                      covlist = cov.sub, linps = linps)

rubin3.match
```

```
# A tibble: 13 x 2
   name        resid.var.ratio
   <chr>                 <dbl>
 1 age                   0.984
 2 female                0.981
 3 caucasian             1.01
 4 married               1.09
 5 stprob                1.13
 6 charlson              0.75
 7 typeca_GI             0.998
 8 typeca_Lung           1.01
 9 typeca_GYN            0.871
10 ecog_0                0.927
11 ecog_1                0.964
12 ecog_2                0.847
13 ecog_3                1.31
```

```
rubin3.match$source <- "Matched"
rubin3.unadj$source <- "Unmatched"

rubin3.both <- bind_rows(rubin3.unadj, rubin3.match)

ggplot(rubin3.both, aes(x = resid.var.ratio, y = name,
                       col = source, pch = source)) +
    geom_point(size = 3) +
    theme_bw() +
    xlim(0.5, 2.0) +
    geom_vline(aes(xintercept = 1)) +
    geom_vline(aes(xintercept = 4/5), linetype = "dashed", col = "red") +
    geom_vline(aes(xintercept = 5/4), linetype = "dashed", col = "red") +
  labs(x = "Residual Variance Ratio", y = "")
```



1.6.4.1 Comparison of Results: Rubin's Rules

Setting	Rubin's Rule 1	Rubin's Rule 2	Rubin's Rule 3 Range
GOAL	0	near 1 (4/5, 5/4)	near 1 (4/5, 5/4)
PASS if…	below 50	(1/2, 2)	(1/2, 2)
Prior to Matching	58.48	0.67	(0.53, 1.41)
After 1:1 Matching	7.65	1.07	(0.75, 1.31)

1.6.5 Task 6c.

> Finally, find a point estimate (and 95% confidence interval) for the effect of the treatment on the `hospice` outcome, based on your 1:1 match on the propensity score. Since the outcomes are binary, you should be using a conditional logistic regression to establish odds ratio estimates, while accounting for the pairs.

We'll run a conditional logistic regression (using the `survival` package) to estimate the intervention effect.

```
model.hospice <- clogit(hospice ~ treated + strata(matches),
                        data=canc3.matchedsample)
```

```
summary(model.hospice)
```

```
Call:
coxph(formula = Surv(rep(1, 300L), hospice) ~ treated + strata(matches),
    data = canc3.matchedsample, method = "exact")

  n= 300, number of events= 123

          coef exp(coef) se(coef)    z Pr(>|z|)
treated 0.0339    1.0345   0.2604 0.13    0.896

        exp(coef) exp(-coef) lower .95 upper .95
treated     1.034     0.9667     0.621     1.723

Concordance= 0.508  (se = 0.092 )
Likelihood ratio test= 0.02  on 1 df,   p=0.9
Wald test            = 0.02  on 1 df,   p=0.9
Score (logrank) test = 0.02  on 1 df,   p=0.9
```

This model estimates the Odds Ratio as OR = 1.03, with 95% CI (0.62, 1.72).

1.7 Task 7

> Compare your unadjusted (Task 1), propensity score-adjusted (by regression: Task 4), propensity score subclassification (Task 5) and propensity matching (Task 6) estimates of the effect of the intervention on the `hospice` outcome in a table (or better, graph.) What can you conclude?

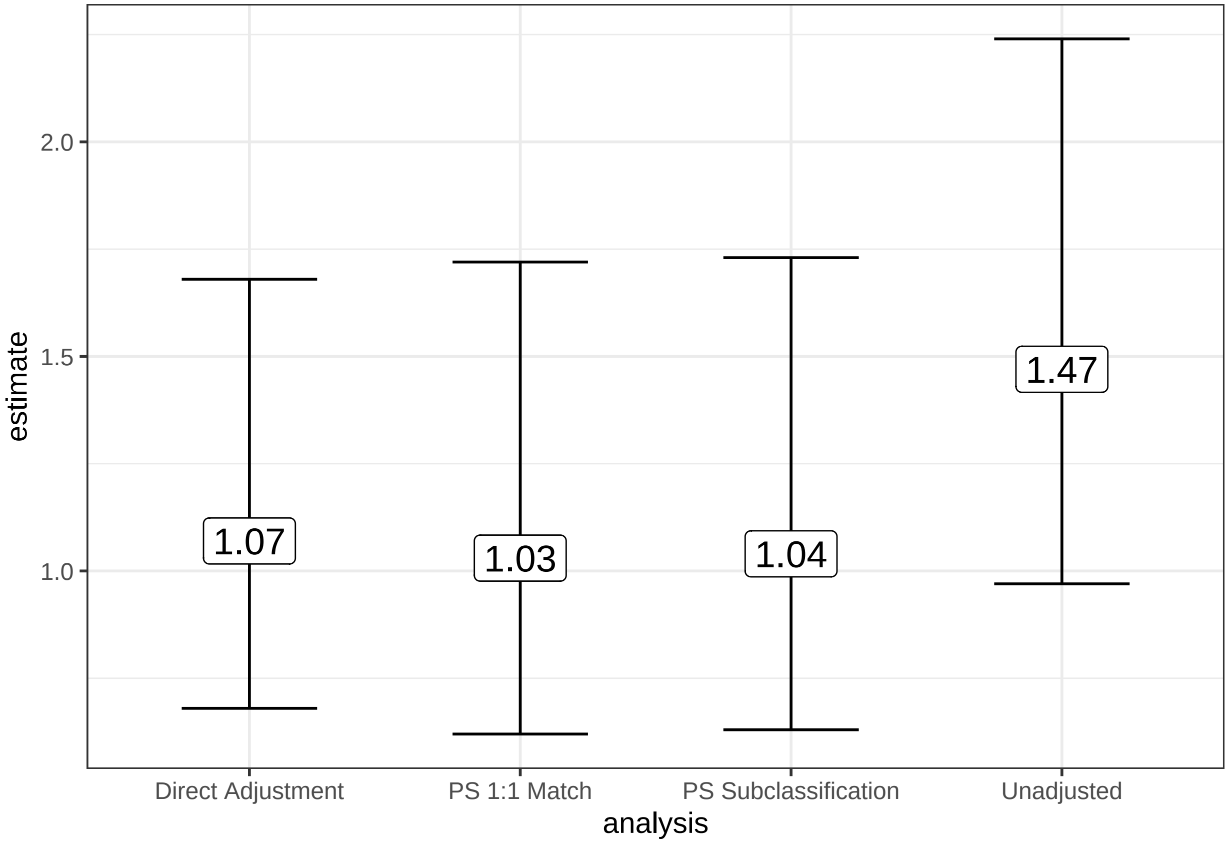
Estimating the **intervention effect** on the `hospice` outcome, we have yet to find a statistically significant result at the 5% significance level.

Analytic Approach	Odds Ratio	95% CI
Unadjusted	1.47	(0.97, 2.24)
Direct PS adjustment	1.07	(0.68, 1.68)
PS quintile subclassification	1.04	(0.63, 1.73)
1:1 propensity score matching	1.03	(0.62, 1.72)

1.7.1 Building a Data Frame of the Results

To make a nice plot, I'll want a data frame of the `hospice` results.

```
res_hospice <- data_frame(
    analysis = c("Unadjusted", "Direct Adjustment",
                 "PS Subclassification", "PS 1:1 Match"),
    estimate = c(1.47, 1.07, 1.04, 1.03),
    conf.low = c(0.97, 0.68, 0.63, 0.62),
    conf.high = c(2.24, 1.68, 1.73, 1.72))

ggplot(res_hospice, aes(x = analysis, y = estimate)) +
    geom_errorbar(aes(ymax = conf.high, ymin = conf.low), width = 0.5) +
    geom_label(aes(label = estimate), size = 5) +
    theme_bw()
```



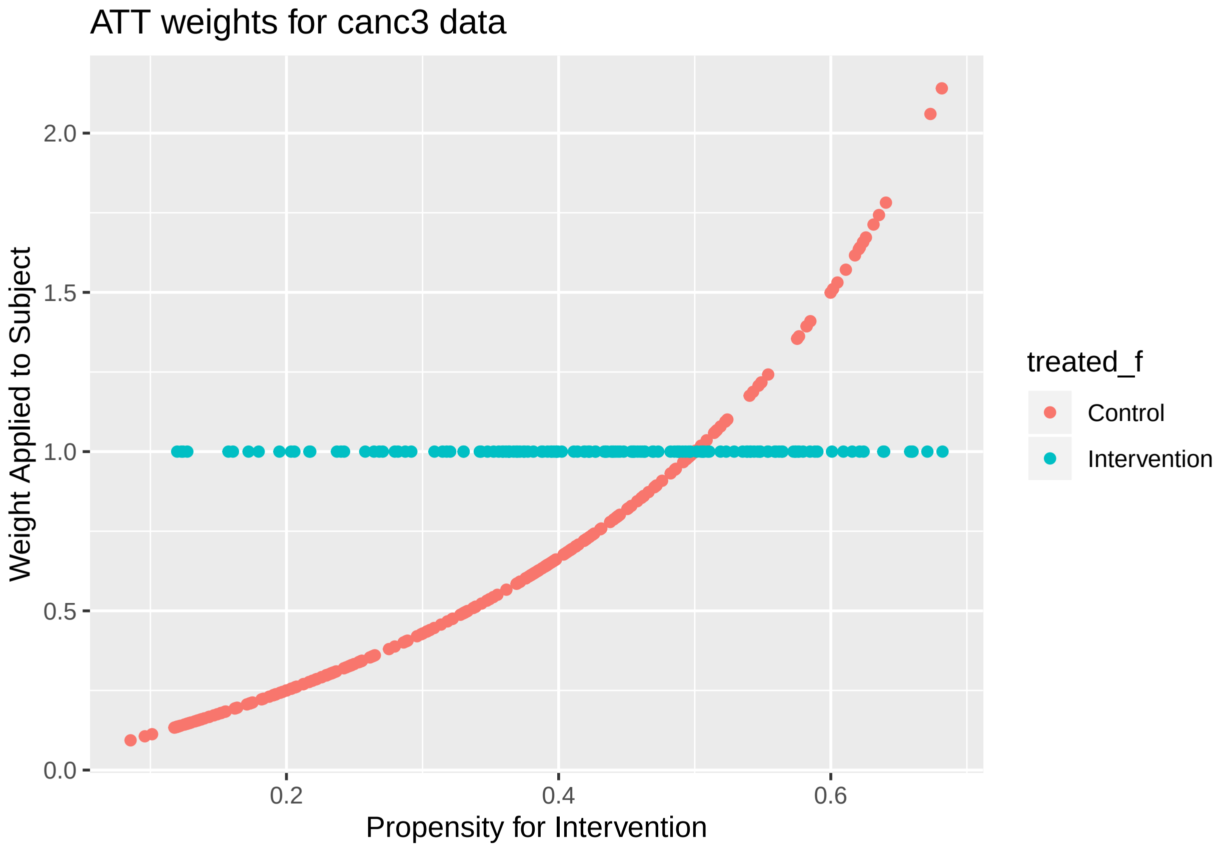
2 Homework 4 Tasks

2.1 Task 1.

Execute weighting by the inverse propensity score, using the ATT approach (weight 1 for all intervention patients and weight `ps/(1-ps)` for all controls.) Plot the weights you applied within the intervention and control groups. Briefly explain what's happening.

```
canc3$wts <- ifelse(canc3$treated==1, 1, canc3$ps/(1-canc3$ps))
```

```
ggplot(canc3, aes(x = ps, y = wts, colour=treated_f)) +
    geom_point() +
    labs(x = "Propensity for Intervention",
         y = "Weight Applied to Subject",
         title = "ATT weights for canc3 data")
```



The intervention patients are each weighted at 1, while the control patients weights vary, as a function of their propensity score. Control patients with unusual combinations of predictors among the controls (and thus relatively high propensity for the intervention) are weighted more than more typical controls (with low propensity scores.)

2.2 Task 2.

Use the `twang` package's `dx.wts` function to start assessing balance after weighting. What is the effective sample size within the control group after weighting? Can you explain what this value means, briefly?

```
canc3_df <- data.frame(canc3) ## twang doesn't play well with tibbles

covlist <- c("ps", "linps", "age", "female", "caucasian",
             "married", "typeca_Lung", "typeca_GYN",
             "ecog_1", "ecog_2", "ecog_3", "stprob",
             "charlson")

bal.wts <- dx.wts(x=canc3$wts, data=canc3_df, vars=covlist,
                  treat.var="treated", estimand="ATT")

bal.wts
```

```
  type n.treat n.ctrl ess.treat ess.ctrl     max.es    mean.es    max.ks
1  unw     150    250       150 250.0000 0.73020532 0.30820291 0.3120000
2          150    250       150 170.3926 0.05904512 0.02398123 0.1712779
     mean.ks iter
1 0.14225641   NA
2 0.04613121   NA
```

The effective sample size within the control group after ATT weighting by the inverse propensity score is 170.3926248, which implies that about 170 of the 250 control patients are comparable to the treatment group. This implies that the results we'll see will have similar power to an observational comparative effectiveness study done with 150 treated and 170 unweighted control subjects.

Quoting the TWANG vignette:

> The ESS is approximately the number of observations from a simple random sample that yields an estimate with sampling variation equal to the sampling variation obtained with the weighted comparison observations. Therefore, the ESS will give an estimate of the number of comparison participants that are comparable to the treatment group when estimand = "ATT".

2.3 Task 3.

Use the `bal.table` function to list (among other things) the standardized effect sizes for your covariate list. What can you conclude about the standardized differences (i.e. 100* the standardized effect sizes) across our covariates? Plot these standardized differences in a Love plot, along with the standardized differences prior to propensity adjustment that you developed in Assignment 3. Are you satisfied with the balance after weighting here?

```
bal.table(bal.wts)
```

```
$unw
             tx.mn  tx.sd  ct.mn  ct.sd std.eff.sz   stat     p    ks ks.pval
ps           0.431  0.129  0.341  0.145      0.701  6.476 0.000 0.300   0.000
linps       -0.308  0.585 -0.735  0.705      0.730  6.547 0.000 0.300   0.000
age         63.760 10.866 62.564 11.262      0.110  1.053 0.293 0.097   0.313
female       0.620  0.487  0.660  0.475     -0.082 -0.804 0.422 0.040   0.996
caucasian    0.727  0.447  0.832  0.375     -0.236 -2.424 0.016 0.105   0.230
married      0.553  0.499  0.648  0.479     -0.190 -1.869 0.062 0.095   0.345
typeca_Lung  0.427  0.496  0.260  0.440      0.336  3.398 0.001 0.167   0.010
typeca_GYN   0.120  0.326  0.304  0.461     -0.564 -4.667 0.000 0.184   0.003
ecog_1       0.567  0.497  0.464  0.500      0.206  1.999 0.046 0.103   0.256
ecog_2       0.060  0.238  0.088  0.284     -0.118 -1.059 0.290 0.028   1.000
ecog_3       0.027  0.162  0.036  0.187     -0.058 -0.528 0.598 0.009   1.000
stprob       0.160  0.220  0.280  0.265     -0.547 -4.907 0.000 0.312   0.000
charlson     0.727  1.140  0.580  1.142      0.129  1.246 0.213 0.109   0.196

[[2]]
             tx.mn  tx.sd  ct.mn  ct.sd std.eff.sz   stat     p    ks ks.pval
ps           0.431  0.129  0.436  0.138     -0.040 -0.341 0.733 0.103   0.342
linps       -0.308  0.585 -0.288  0.621     -0.035 -0.309 0.757 0.103   0.342
age         63.760 10.866 63.909 11.734     -0.014 -0.115 0.908 0.086   0.556
female       0.620  0.487  0.642  0.480     -0.046 -0.419 0.675 0.022   1.000
caucasian    0.727  0.447  0.718  0.451      0.019  0.160 0.873 0.009   1.000
married      0.553  0.499  0.524  0.500      0.059  0.524 0.601 0.029   1.000
typeca_Lung  0.427  0.496  0.438  0.497     -0.024 -0.208 0.835 0.012   1.000
typeca_GYN   0.120  0.326  0.122  0.328     -0.007 -0.077 0.939 0.002   1.000
ecog_1       0.567  0.497  0.561  0.497      0.010  0.094 0.925 0.005   1.000
ecog_2       0.060  0.238  0.054  0.227      0.024  0.238 0.812 0.006   1.000
ecog_3       0.027  0.162  0.025  0.158      0.008  0.076 0.939 0.001   1.000
stprob       0.160  0.220  0.161  0.201     -0.006 -0.058 0.954 0.171   0.016
charlson     0.727  1.140  0.751  1.349     -0.021 -0.156 0.876 0.050   0.979
```

To build a standardized difference plot, we'll first collect the standardized effect sizes, and multiply them by 100 to produce estimated standardized differences. Then, we'll sort the results by the pre-weight standardized differences, to yield the table we'll need.

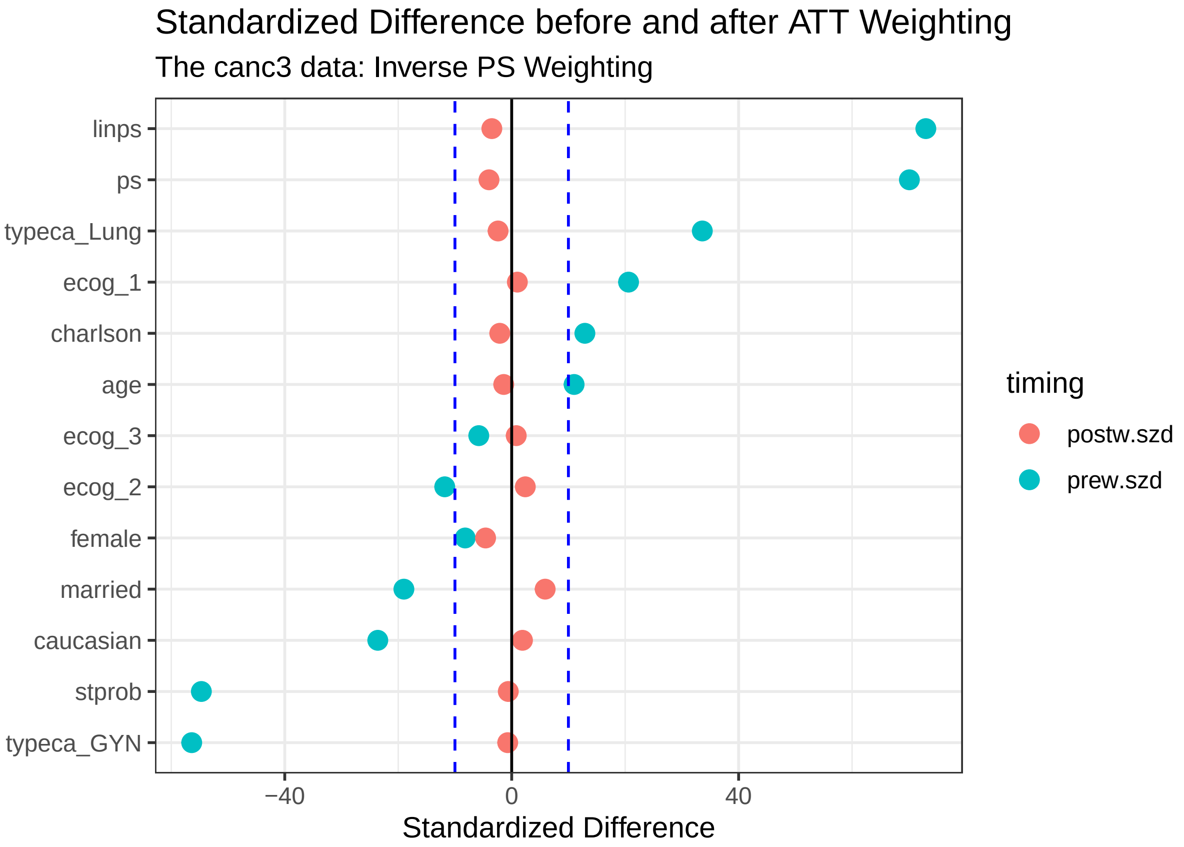
```
szd.weight1 <- data.frame(names=rownames(bal.table(bal.wts)[[2]]),
prew.szd = 100*bal.table(bal.wts)[[1]]$std.eff.sz,
postw.szd = 100*bal.table(bal.wts)[[2]]$std.eff.sz)
szd.weights <- szd.weight1[with(szd.weight1, order(prew.szd)),]
szd.weights
```

```
         names prew.szd postw.szd
8   typeca_GYN    -56.4      -0.7
```

```
12      stprob    -54.7      -0.6
5    caucasian    -23.6       1.9
6      married    -19.0       5.9
10      ecog_2    -11.8       2.4
4       female     -8.2      -4.6
11      ecog_3     -5.8       0.8
3          age     11.0      -1.4
13    charlson     12.9      -2.1
9       ecog_1     20.6       1.0
7  typeca_Lung     33.6      -2.4
1           ps     70.1      -4.0
2        linps     73.0      -3.5
```

And now, we can generate the plot, either through base graphics, or, as shown below, with ggplot2.

```r
szd.weight1_plot <- gather(data = szd.weights, key = timing, value = szd, 2:3)

ggplot(szd.weight1_plot, aes(x = szd, y = reorder(names, szd),
                             color = timing)) +
    geom_point(size = 3) +
    geom_vline(xintercept = 0) +
    geom_vline(xintercept = c(-10,10), linetype = "dashed", col = "blue") +
    theme_bw() +
    labs(x = "Standardized Difference", y = "",
         title = "Standardized Difference before and after ATT Weighting",
         subtitle = "The canc3 data: Inverse PS Weighting")
```



2.4 Task 4.

Evaluate Rubin's Rule 1 and Rule 2 for the post-weighting covariate distributions. Do the results seem satisfactory?

In a word, yes.

2.4.1 Rubin's Rule 1

From the `bal.table` output above, the `std.eff.sz` for `linps` after the weighting is -0.035, so that's a standardized difference of 100 x -0.035 = -3.5%, which is well below Rubin's maximum tolerable level in Rule 1 of 50%, so we pass Rule 1.

2.4.2 Rubin's Rule 2

From that same table, the post-weighting *treatment* standard deviation for `linps` is 0.585 and so, squaring the SD, we find the variance is 0.342225.

The post-weighting *control* standard deviation for `linps` is 0.621 and so the variance is 0.385641.

So that's a variance ratio for linps of 0.342225 / 0.385641 = 0.887, which is well within the maximum tolerable range of 0.5 to 2, and even within the tighter range we typically try to achieve of 0.8 to 1.25, so we also pass Rule 2.

2.5 Task 5.

Now use the twang package to create both the propensity scores (using generalized boosted modeling) and the ATT weights. Compare your results from 1-5 to your result here in terms of the following measures.

- effective sample size
- Love plot and standardized differences
- Rubin's first two rules

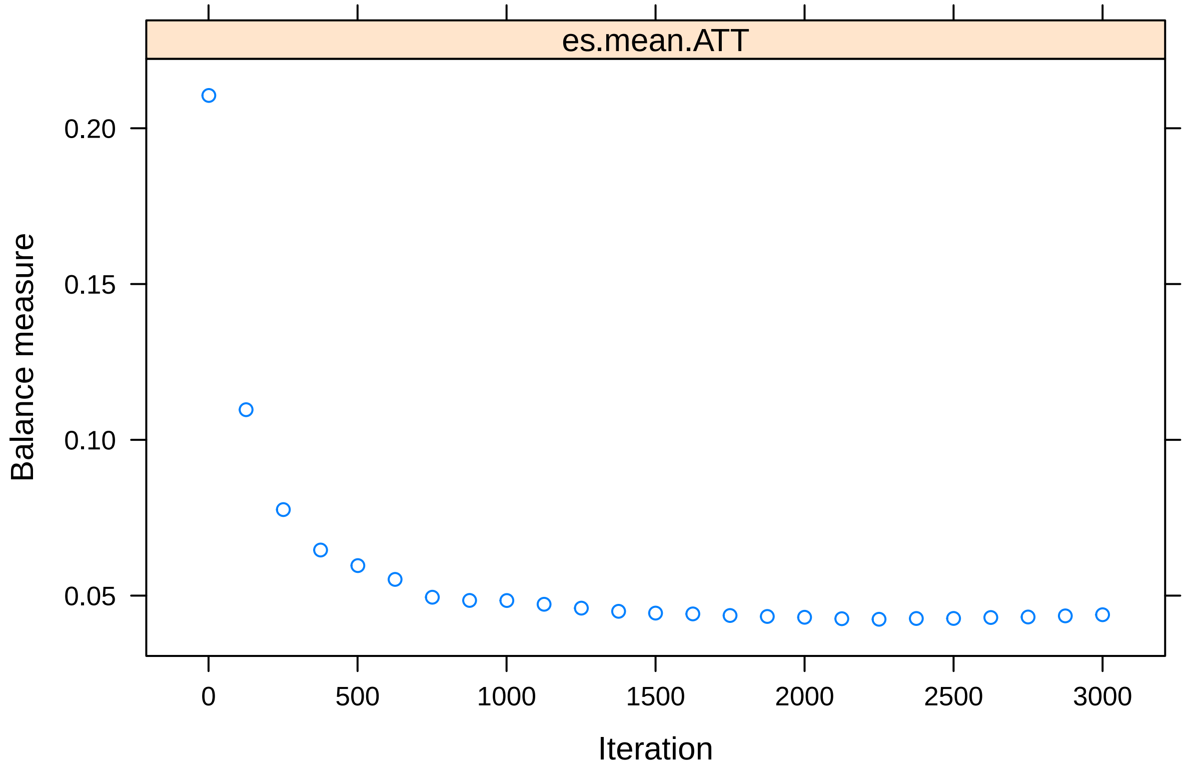
2.5.1 Creating the weights with twang

Start by estimating the propensity score using the twang function ps.

```
ps_canc3 <- ps(treated ~ age + female + caucasian + married + typeca +
                 stprob + charlson + ecog,
             data = canc3_df,
             n.trees = 3000,
             interaction.depth = 2,
             stop.method = c("es.mean"),
             estimand = "ATT",
             verbose = FALSE)
```

Does 3000 look like a long enough simulation run?

```
plot(ps_canc3)
```



2.5.2 Effective Sample Size

What is the effective sample size of these weighted results?

```
summary(ps_canc3)
```

```
             n.treat n.ctrl ess.treat ess.ctrl     max.es    mean.es     max.ks
unw              150    250       150 250.0000 0.56622086 0.21167255 0.31200000
es.mean.ATT      150    250       150 133.2961 0.08773735 0.04233414 0.05048702
             max.ks.p    mean.ks iter
unw                NA 0.10246154   NA
es.mean.ATT        NA 0.02469004 2291
```

The effective sample size in the control group is now 133.2961035, which is considerably smaller than we saw previously. Perhaps the balance of covariates will be better?

2.5.3 Standardized Differences / Love Plot

```
bal.tab(ps_canc3, full.stop.method = "es.mean.att")
```

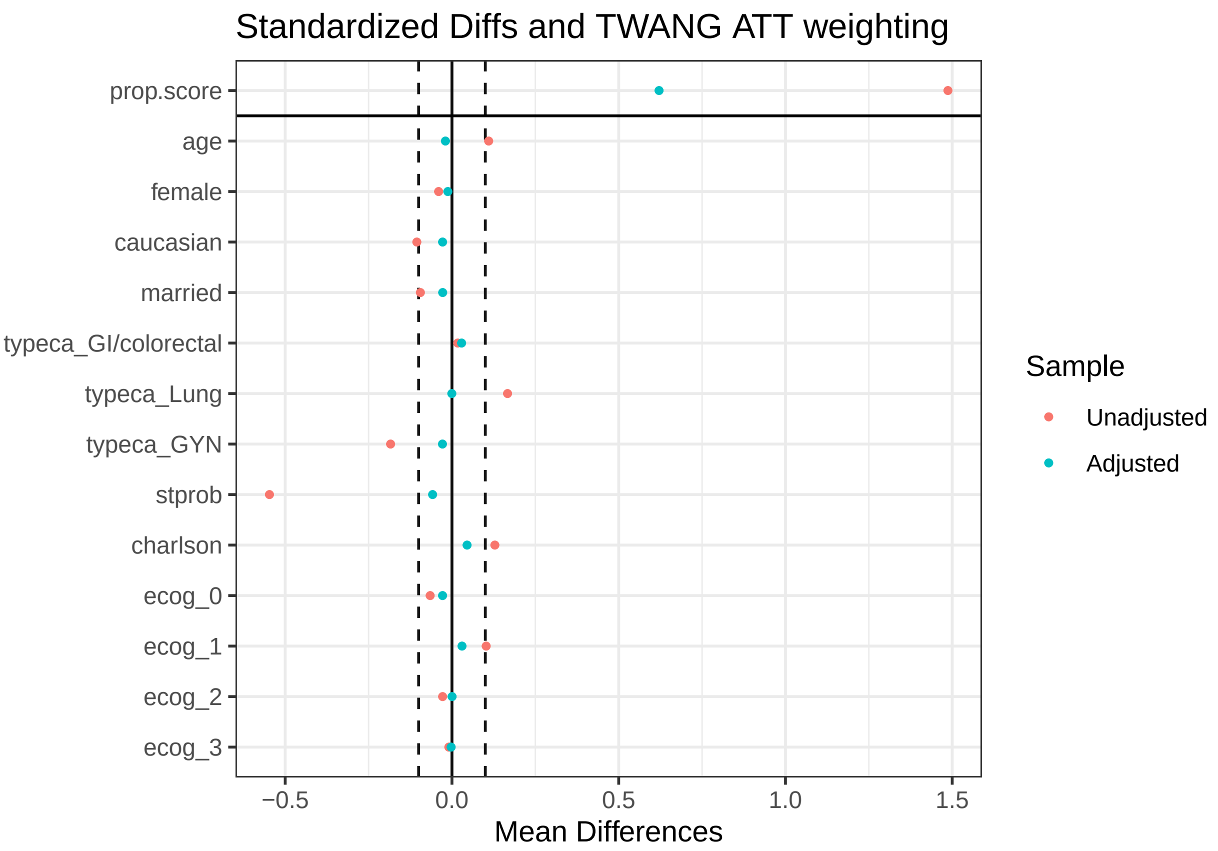
```
Call
```

```
 ps(formula = treated ~ age + female + caucasian + married + typeca +
    stprob + charlson + ecog, data = canc3_df, n.trees = 3000,
    interaction.depth = 2, verbose = FALSE, estimand = "ATT",
    stop.method = c("es.mean"))

Balance Measures
                         Type Diff.Adj
prop.score           Distance   0.6210
age                   Contin.  -0.0196
female                 Binary  -0.0124
caucasian              Binary  -0.0281
married                Binary  -0.0278
typeca_GI/colorectal   Binary   0.0289
typeca_Lung            Binary  -0.0004
typeca_GYN             Binary  -0.0285
stprob                Contin.  -0.0580
charlson              Contin.   0.0453
ecog_0                 Binary  -0.0281
ecog_1                 Binary   0.0301
ecog_2                 Binary   0.0004
ecog_3                 Binary  -0.0024

Effective sample sizes
           Control Treated
Unadjusted 250.000     150
Adjusted   133.296     150
```

```
p <- love.plot(bal.tab(ps_canc3),
               threshold = .1, size = 1.5,
               title = "Standardized Diffs and TWANG ATT weighting")
```

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

```
p + theme_bw()
```



The Love plot looks meaningfully worse on the propensity score in the ATT weights approach, and the two strategies also yield different effective sample sizes.

2.6 Task 6.

Select the weighting approach (of the two you have developed) that you prefer, and use it to find propensity-weighted estimates of the intervention effect on survival and on hospice. Your results should include a properly labeled point estimate and associated confidence interval for each outcome.

I'll go with the ATT weights I generated from the same propensity score model we've been using for matching, etc., because the balance of the propensity score is better, and the effective sample size is larger.

2.6.1 Analysis of survival using propensity score generated ATT weights

For *survival*, we fit a logistic regression model, and exponentiate the log odds ratio treatment effect estimate to obtain an odds ratio estimate of the average causal effect of treatment on the treated.

```r
canc3wt.design <- svydesign(ids=~1, weights=~wts, data=canc3)
survadj.wt <- svyglm(alive ~ treated, design=canc3wt.design,
                     family=quasibinomial())
summary(survadj.wt)
```

```
Call:
svyglm(formula = alive ~ treated, design = canc3wt.design, family = quasibinomial())

Survey design:
svydesign(ids = ~1, weights = ~wts, data = canc3)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.3560     0.1574   2.261   0.0243 *
treated      -0.1688     0.2275  -0.742   0.4586
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 1.002506)

Number of Fisher Scoring iterations: 4
```

```r
exp(summary(survadj.wt)$coef)
```

```
             Estimate Std. Error   t value Pr(>|t|)
(Intercept) 1.4275635   1.170469 9.5974418 1.024564
treated     0.8447136   1.255425 0.4762198 1.581858
```

```r
exp(confint(survadj.wt))
```

```
                2.5 %   97.5 %
(Intercept) 1.0486069 1.943471
treated     0.5408578 1.319277
```

Our odds ratio estimate for the intervention's effect on *survival* is 0.84 and our 95% CI is (0.54, 1.32).

2.6.2 Analysis of hospice using propensity score generated ATT weights

For *hospice*, we adopt the same approach…

```r
canc3wt.design <- svydesign(ids=~1, weights=~wts, data=canc3)
hospadj.wt <- svyglm(hospice ~ treated, design=canc3wt.design,
                     family=quasibinomial())
exp(summary(hospadj.wt)$coef)
```

```
             Estimate Std. Error   t value Pr(>|t|)
(Intercept) 0.6313001   1.172317 0.0553959 1.004030
treated     1.1160231   1.258427 1.6121247 1.883686
```

```
exp(confint(hospadj.wt))
```

```
                2.5 %    97.5 %
(Intercept) 0.4622855 0.8621074
treated     0.7112362 1.7511869
```

Our odds ratio estimate for the intervention's effect on *hospice* is 1.12 and our 95% CI is (0.71, 1.75).

2.7 Task 7.

Next, run an analysis that combines weighting (either approach is OK) with regression adjustment for the linear propensity score to obtain a "doubly robust" set of estimates. Use this approach to again find estimates of the intervention effect on survival and hospice.

2.7.1 Double Robust Analysis of survival via usual ATT weights

For *survival*, we simply fit the same logistic regression model but now add in the linear propensity score as a predictor, then exponentiate the log odds ratio treatment effect estimate to obtain an odds ratio estimate of the average causal effect of treatment on the treated.

```
canc3wt.design <- svydesign(ids=~1, weights=~wts, data=canc3)
survadj.dr <- svyglm(alive ~ treated + linps, design=canc3wt.design,
                     family=quasibinomial())
summary(survadj.dr)
```

```

Call:
svyglm(formula = alive ~ treated + linps, design = canc3wt.design,
    family = quasibinomial())

Survey design:
svydesign(ids = ~1, weights = ~wts, data = canc3)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.1591     0.1883   0.845 0.398438
treated      -0.1937     0.2330  -0.831 0.406481
linps        -0.7651     0.1955  -3.913 0.000107 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for quasibinomial family taken to be 0.9983068)

Number of Fisher Scoring iterations: 4
```

```
exp(summary(survadj.dr)$coef)
```

```
             Estimate Std. Error    t value Pr(>|t|)
(Intercept) 1.1725063   1.207151 2.32873880 1.489496
treated     0.8239451   1.262429 0.43561950 1.501525
linps       0.4653067   1.215945 0.01998238 1.000107
```

```
exp(confint(survadj.dr))
```

```
                2.5 %    97.5 %
(Intercept) 0.8107101 1.6957616
treated     0.5218384 1.3009499
linps       0.3171836 0.6826025
```

Our odds ratio estimate for the intervention's effect on *survival* is 0.82 and our 95% CI is (0.52, 1.3).

2.7.2 Analysis of hospice using propensity score generated ATT weights

For *hospice*, we adopt the same approach…

```
canc3wt.design <- svydesign(ids=~1, weights=~wts, data=canc3)
hospadj.dr <- svyglm(hospice ~ treated + linps, design=canc3wt.design,
                     family=quasibinomial())
exp(summary(hospadj.dr)$coef)
```

```
             Estimate Std. Error    t value Pr(>|t|)
(Intercept) 0.7574157   1.208647  0.2308066 1.154182
treated     1.1398428   1.265925  1.7420856 1.784522
linps       2.0549502   1.216629 39.3767866 1.000272
```

```
exp(confint(hospadj.dr))
```

```
                2.5 %   97.5 %
(Intercept) 0.5224323 1.098092
treated     0.7180074 1.809510
linps       1.3992464 3.017925
```

Our odds ratio estimate for the intervention's effect on *hospice* is 1.14 and our 95% CI is (0.72, 1.81).

2.8 Task 8.

Finally, compare your results in Tasks 6 and 7 here to those obtained in Assignment 3 for the `hospice` outcome. What conclusions can you draw?

Estimating the **intervention effect** on the `hospice` outcome, we have yet to find a statistically significant result at the 5% significance level.

Analytic Approach	Odds Ratio	95% CI
Unadjusted	1.47	(0.97, 2.24)
Direct PS adjustment	1.07	(0.68, 1.68)
PS quintile subclassification	1.04	(0.63, 1.73)
1:1 propensity score matching	1.03	(0.62, 1.72)
ATT weights from usual PS	1.12	(0.71, 1.75)
Double Robust from usual PS	1.14	(0.72, 1.81)