500 Assignment 3 Answer Sketch

Thomas E. Love

due 2020-02-20 Sketch generated 2020-02-20.

Contents

Preliminaries

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

```
library(here); library(janitor); library(magrittr)
library(Hmisc)
library(knitr)
library(tableone)
library(arm)
library(Matching)
library(cobalt)
library(broom)
library(survival)
library(tidyverse)

theme_set(theme_bw())
```

```
canc3 <- read_csv(here("data/canc3.csv")) %>%
    mutate(subject = as.character(subject))
```

Data

We have completed the data collection in a simulated 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. The `canc3.csv` data file is available above.

The Codebook

The data file includes 400 observations, on 12 variables.

Variable	Description	Notes
`subject`	Study ID number	1-250 are control, 251-400 are intervention
`treated`	Treatment status	1 = intervention (150), 0 = control (250)
`age`	Patient age	At study entry, Observed range: 34, 93 years
`female`	Patient sex	1 = female (n = 258), 0 = male (n = 142)
`race`	Patient's race	1 = Caucasian / White (n = 317), 0 = not (n = 83)
`married`	Marital status	At study entry: 1 = Married (n = 245), 0 = not (n = 155)
`typeca`	Type of cancer	3 categories: 1 = GI/colorectal (n = 177), 2 = Lung (n = 129), 3 = GYN (n = 94).
`stprob`	5-year survival	Model probability of 5-year survival, based on type and stage of cancer. Observed range: 0.01, 0.72
`charlson`	Charlson score	Comorbidity index at baseline: higher scores indicate greater comorbidity. Observed range: 0-7.
`ecog`	ECOG score	0 = fully active, 1 = restricted regarding physically strenuous activity, 2 = ambulatory, can self-care, otherwise limited, 3 = capable of only limited self-care.
`alive`	Mortality Status	Alive at study conclusion & 1 = alive (n = 245), 0 = dead (n = 155)
`hospice`	Hospice Status	Entered hospice before death or study end & 1 = hospice (n = 143), 0 = no (n = 257)

- Note: You are welcome to treat `ecog` and `charlson` as either quantitative or categorical variables in developing your response. In this sketch, I will treat `ecog` (and `typeca`) as categorical and `charlson` as quantitative.

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.

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"))
           )
```

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 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 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)

unadj_alive_tidy <- tidy(unadj_alive, exponentiate = TRUE,
     conf.int = TRUE, conf.level = 0.95) %>%
    select(term, estimate, std.error, conf.low, conf.high)

unadj_alive_tidy
```

```
# A tibble: 2 x 5
  term                  estimate std.error conf.low conf.high
  <chr>                    <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)              1.87      0.133    1.45      2.44
2 treated_fIntervention    0.644     0.211    0.425     0.973
```

And so our odds ratio estimate for the intervention's impact on survival (with a 95% confidence interval) is just ...

```
unadj_alive_tidy %>%
  filter(term == "treated_fIntervention") %>%
  select(estimate, conf.low, conf.high) %>%
  kable(digits = 2)
```

estimate	conf.low	conf.high
0.64	0.43	0.97

1.2 Unadjusted logistic regression model for the `hospice` outcome

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

```
unadj_hospice_tidy <- tidy(unadj_hospice, exponentiate = TRUE,
     conf.int = TRUE, conf.level = 0.95) %>%
    select(term, estimate, std.error, conf.low, conf.high)

unadj_hospice_tidy
```

```
# A tibble: 2 x 5
  term                  estimate std.error conf.low conf.high
  <chr>                    <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)              0.479     0.135    0.366     0.622
2 treated_fIntervention    1.47      0.214    0.966     2.24
```

And so our odds ratio estimate for the intervention's impact on going to hospice (with a 95% confidence interval) is . . .

```
unadj_hospice_tidy %>%
  filter(term == "treated_fIntervention") %>%
  select(estimate, conf.low, conf.high) %>%
  kable(digits = 2)
```

estimate	conf.low	conf.high
1.47	0.97	2.24

The odds of going to hospice are higher, but not statistically detectably higher (at a 95% confidence level) for intervention patients as compared to control patients.

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)

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.

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 <- canc3 %>%
  mutate(ps = psmodel$fitted,
         linps = psmodel$linear.predictors)
```

2.2 Describing the Overlap Numerically

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

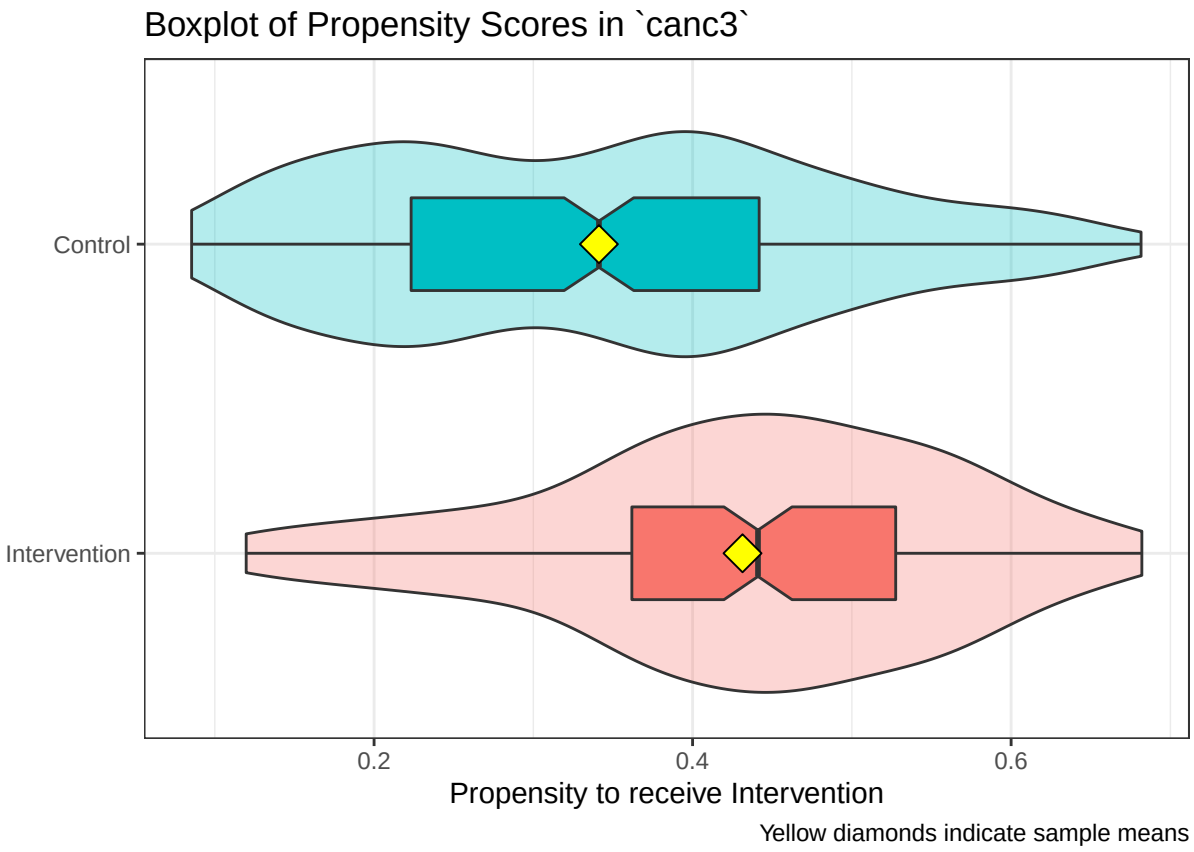
```
# A tibble: 2 x 8
  treated_f    mean.ps sd.ps median.ps min.ps max.ps mean.linps sd.linps
  <fct>          <dbl> <dbl>     <dbl>  <dbl>  <dbl>      <dbl>    <dbl>
1 Control        0.341 0.145     0.341 0.0854  0.682     -0.735    0.705
2 Intervention   0.431 0.129     0.441 0.120   0.682     -0.308    0.585
```

- All of our propensity scores are between 0.09 and 0.68, so that's well within the range of (0.01, 0.99) that we're hoping to see.
- The average propensity score is larger in the Intervention group than the Control, as we'd planned.

2.3 Describing the Overlap Graphically

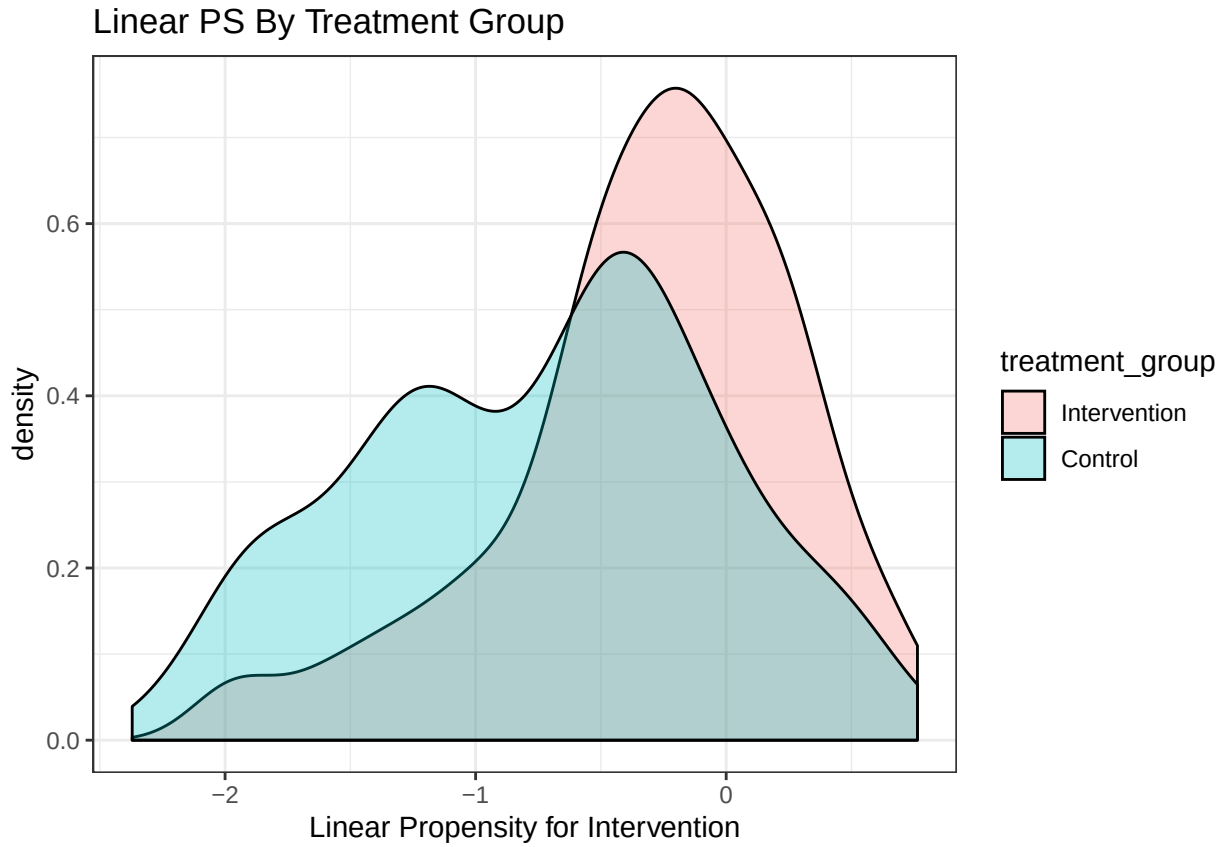
First, we'll produce a boxplot with a violin plot, and the means superimposed, for the raw propensity scores.

```
ggplot(canc3, aes(x = treatment_group, y = ps,
                  fill = treatment_group)) +
  geom_violin(alpha = 0.3) +
  geom_boxplot(width = 0.3, notch=TRUE) +
  stat_summary(fun.y="mean", geom="point",
               shape=23, size = 5, fill = "yellow") +
  coord_flip() +
  guides(fill = FALSE) +
    labs(x = "",
         y = "Propensity to receive Intervention",
         title = "Boxplot of Propensity Scores in `canc3`",
         caption = "Yellow diamonds indicate sample means")
```



Next, we'll produce a density plot of the linear propensity scores.

```
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")
```



There are lots of other approaches we could take to visualize the overlap, of course.

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?

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

```
[1] 61.59673
```

Here, I've used the overall standard deviation of the linear propensity scores as my denominator. We could instead have restricted this to the standard deviation within the treatment group, yielding...

```
rubin1.unadj_ATT <- canc3 %$%
  abs(100*(mean(linps[treated==1]) -
             mean(linps[treated==0])) /
                          sd(linps[treated == 1]))
rubin1.unadj_ATT
```

```
[1] 73.02053
```

Either way, we cannot 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.

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 just barely.

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

I didn't ask you to do this, but one way of 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 <- tibble(name = names(covlist),
                  resid.var.ratio = as.numeric(res))
  return(final)
}
```

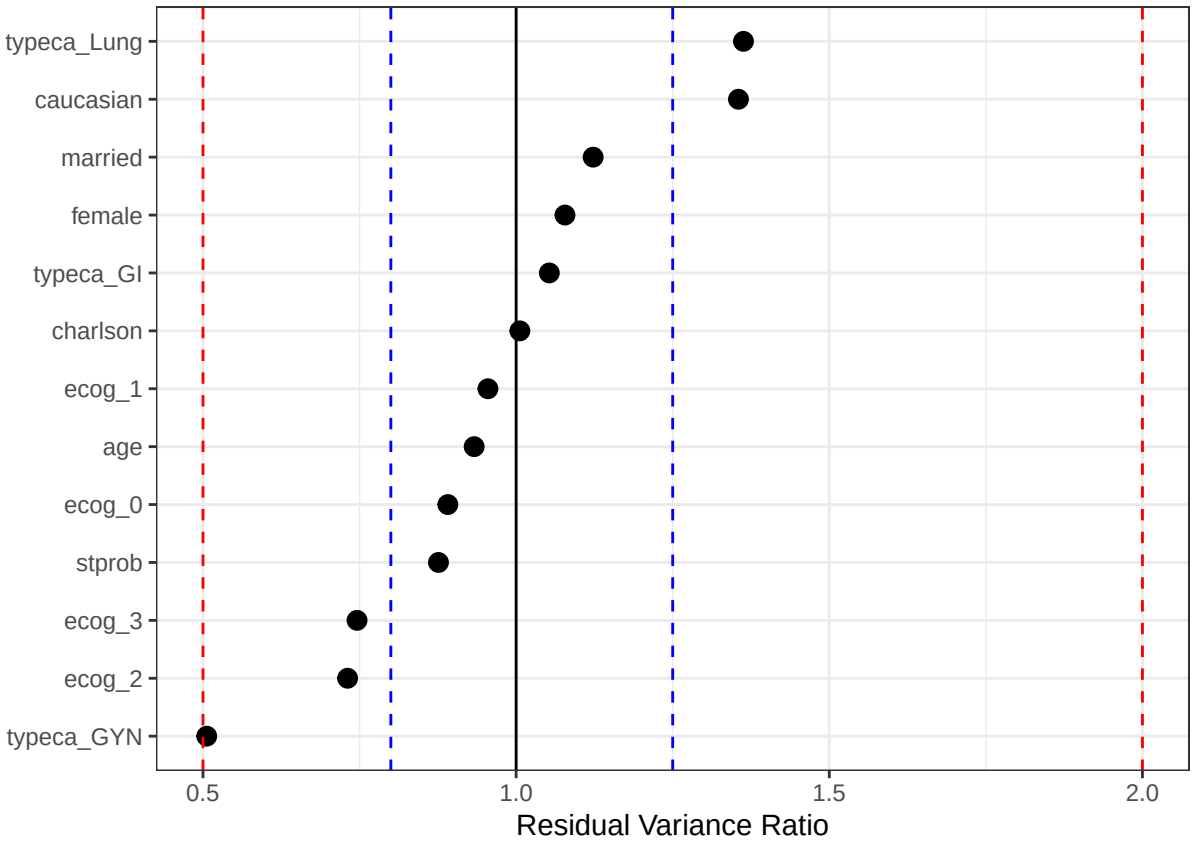
Now, then, applying the rule to our sample prior to propensity score adjustment, we get ...

```
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)

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 = "")
```


typeca_Lung
caucasian
married
female
typeca_GI
charlson
ecog_1
age
ecog_0
stprob
ecog_3
ecog_2
typeca_GYN
0.5
1.0
1.5
2.0
Residual Variance Ratio

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.

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)
```

```
tidy(adj.hospice, exponentiate = TRUE, conf.int = TRUE, conf.level = 0.95)
```

```
# A tibble: 3 x 7
  term        estimate std.error statistic    p.value conf.low conf.high
  <chr>          <dbl>     <dbl>     <dbl>      <dbl>    <dbl>     <dbl>
1 (Intercept)    0.823     0.175    -1.11  0.265         0.583      1.16
2 treated        1.07      0.228     0.311 0.756         0.684      1.68
3 linps          2.25      0.176     4.61  0.00000404    1.61       3.21
```

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.

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)

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.

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)
```

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
```

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
```

```
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

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.

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:

```
## 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
```

```
## 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
```

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)

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.

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
```

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

```
## 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),])
```

As a sanity check, let's ensure that our matched sample has 150 treated and 150 control subjects.

```
canc3.matchedsample %>% count(treated_f)
```

```
# A tibble: 2 x 2
  treated_f        n
  <fct>        <int>
1 Control        150
2 Intervention   150
```

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.

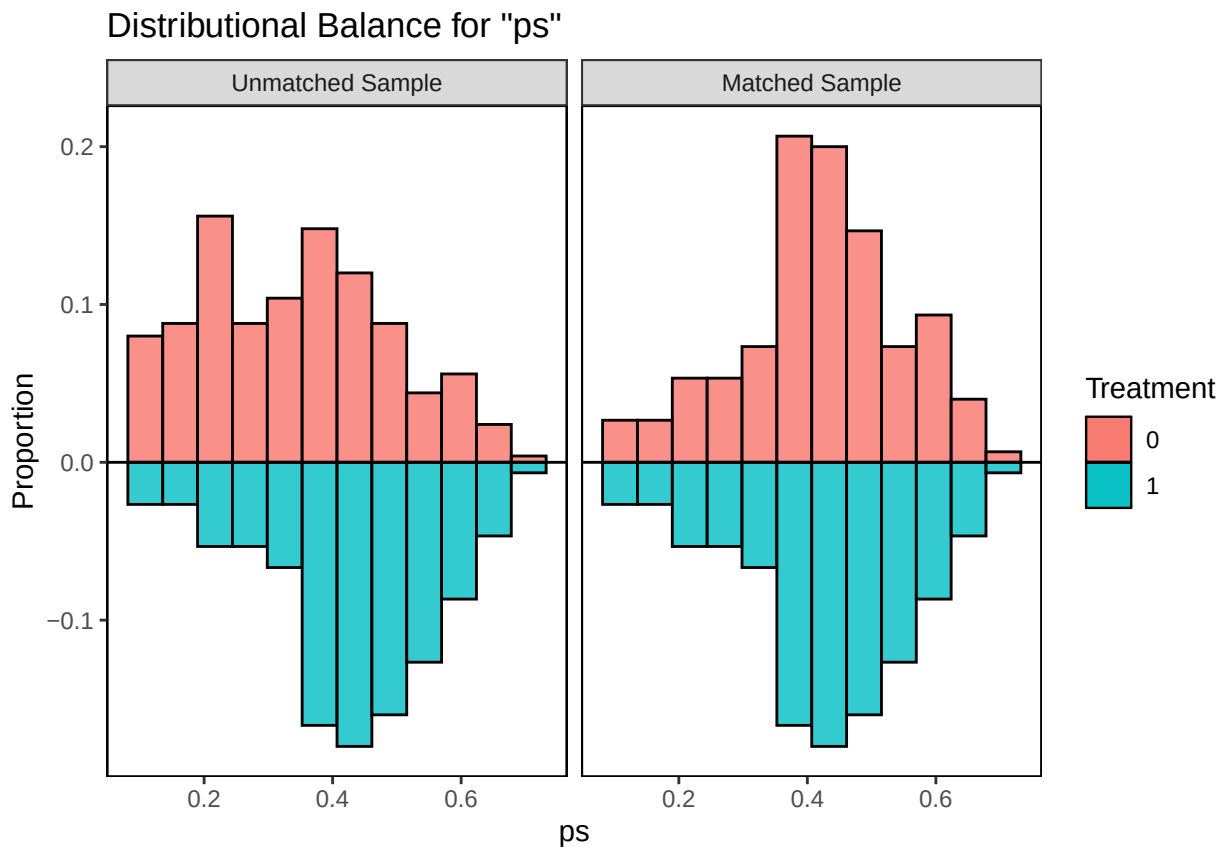
```
covs_1 <- canc3 %>%
    select(age, female, caucasian, married, typeca, stprob,
           charlson, ecog, ps, linps)

b <- bal.tab(match1,
             treat = canc3$treated,
             covs = covs_1,
             quick = FALSE, un = TRUE, disp.v.ratio = TRUE)

b
```

```
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
```

6.2.1 Distributional Balance of the propensity scores

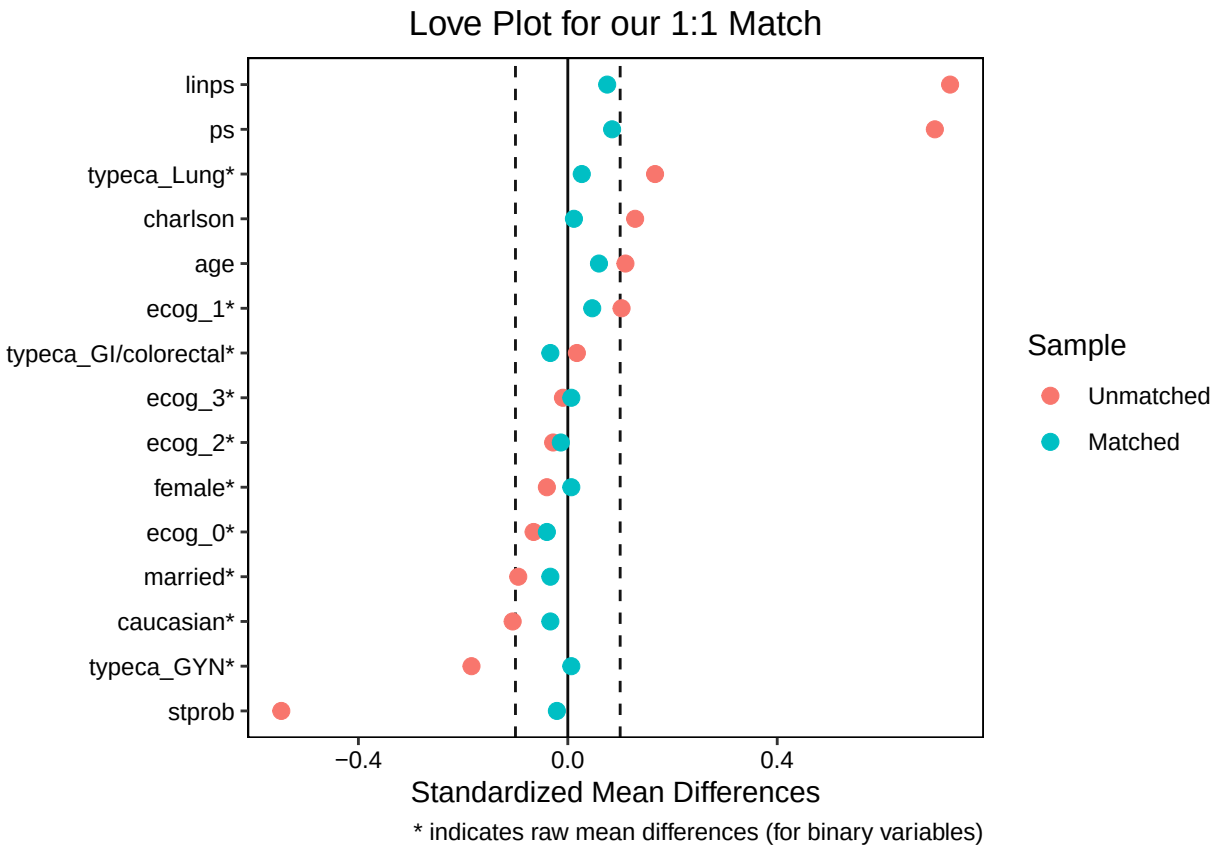
```r
bal.plot(obj = match1,
         treat = canc3$treated,
         covs = covs_1,
         var.name = "ps",
         which = "both",
         sample.names =
             c("Unmatched Sample", "Matched Sample"),
         type = "histogram", mirror = TRUE)
```



6.2.2 Love Plot of Standardized Differences

Note the use of stars to show the results for the indicator variables.

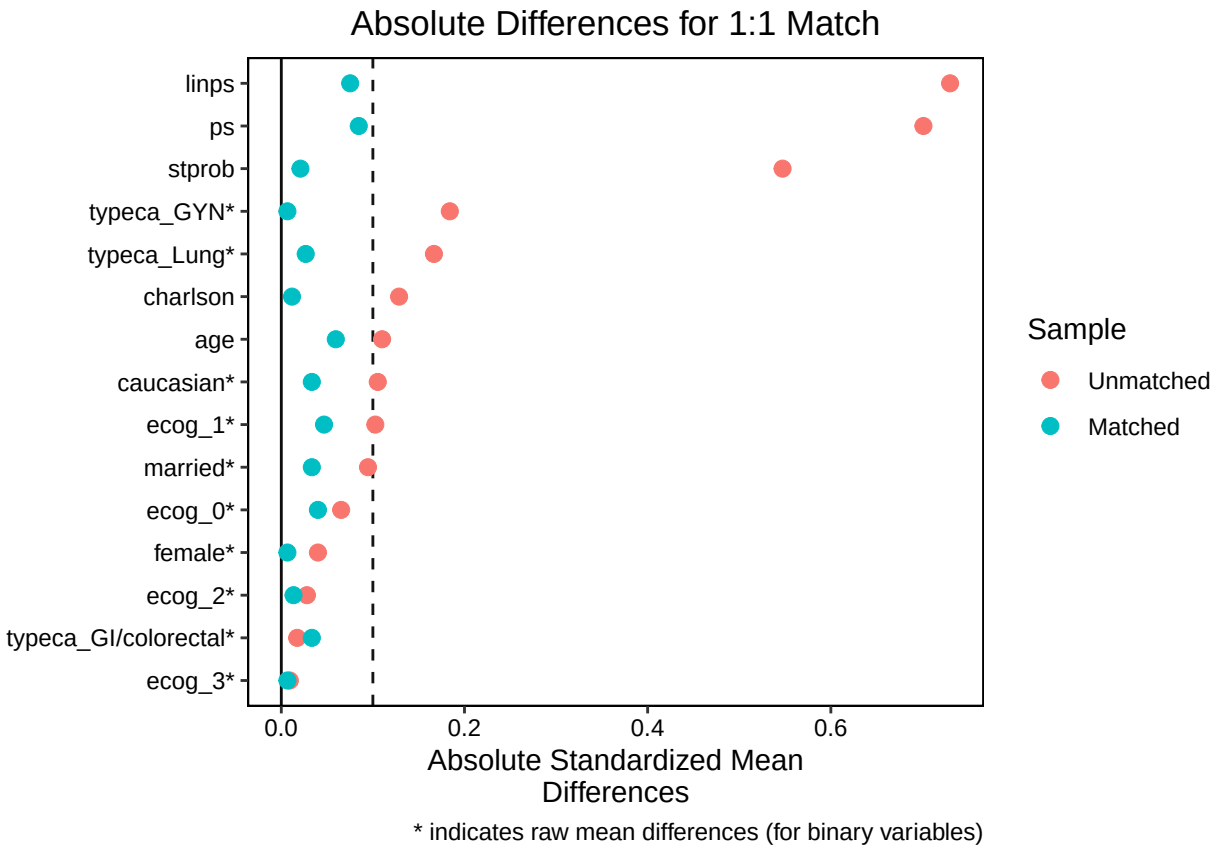
```
love.plot(b,
          threshold = .1, size = 3,
          var.order = "unadjusted",
          stats = "mean.diffs",
          stars = "raw",
          sample.names = c("Unmatched", "Matched"),
          title = "Love Plot for our 1:1 Match") +
    labs(caption = "* indicates raw mean differences (for binary variables)")
```



6.2.3 Plot of Variance Ratios

Note the use of stars to show the results for the indicator variables.

```
love.plot(b, threshold = .1, size = 3,
          var.order = "unadjusted",
          stats = "mean.diffs",
          stars = "raw",
          abs = TRUE,
          sample.names = c("Unmatched", "Matched"),
          title = "Absolute Differences for 1:1 Match") +
    labs(caption = "* indicates raw mean differences (for binary variables)")
```



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.

6.3.1 Checking Rubin's Rules 1 and 2

```
covs_for_rubin <- canc3 %>%
    select(linps)

rubin_m1 <- bal.tab(M = match1,
                treat = canc3$treated,
                covs = covs_for_rubin,
                un = TRUE, disp.v.ratio = TRUE)[1]

rubin_report_m1 <- tibble(
    status = c("Rule1", "Rule2"),
    Unmatched = c(rubin_m1$Balance$Diff.Un,
```

```
                  rubin_m1$Balance$V.Ratio.Un),
    Matched = c(rubin_m1$Balance$Diff.Adj,
                rubin_m1$Balance$V.Ratio.Adj))

rubin_report_m1 %>% knitr::kable(digits = 2)
```

status	Unmatched	Matched
Rule1	0.73	0.08
Rule2	0.69	1.07

Note that this approach uses the standard deviation of the linear propensity score within the treated group only to calculate Rubin's Rule 1.

6.3.2 Evaluate the balance using Rubin's Rule 3 after Matching

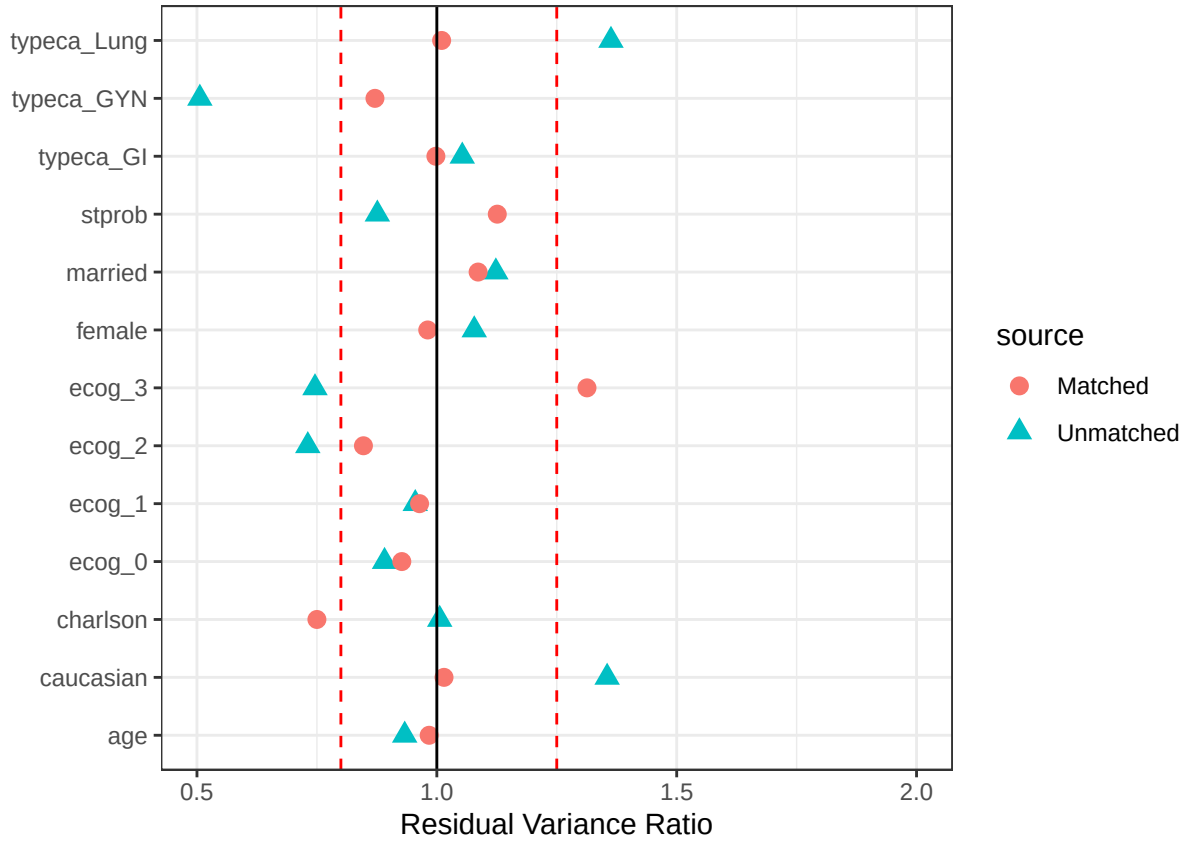
This wasn't something I was expecting you to do...

```
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 = "")
```



6.3.3 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	73	0.69	(0.51, 1.36)
After 1:1 Matching	8	1.07	(0.75, 1.31)

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

summary(model_hospice_matched)
```

```
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
```

```
tidy(model_hospice_matched, exponentiate = TRUE,
     conf.int = TRUE, conf.level = 0.95)
```

```
# 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     1.03     0.260     0.130   0.896    0.621      1.72
```

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

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?

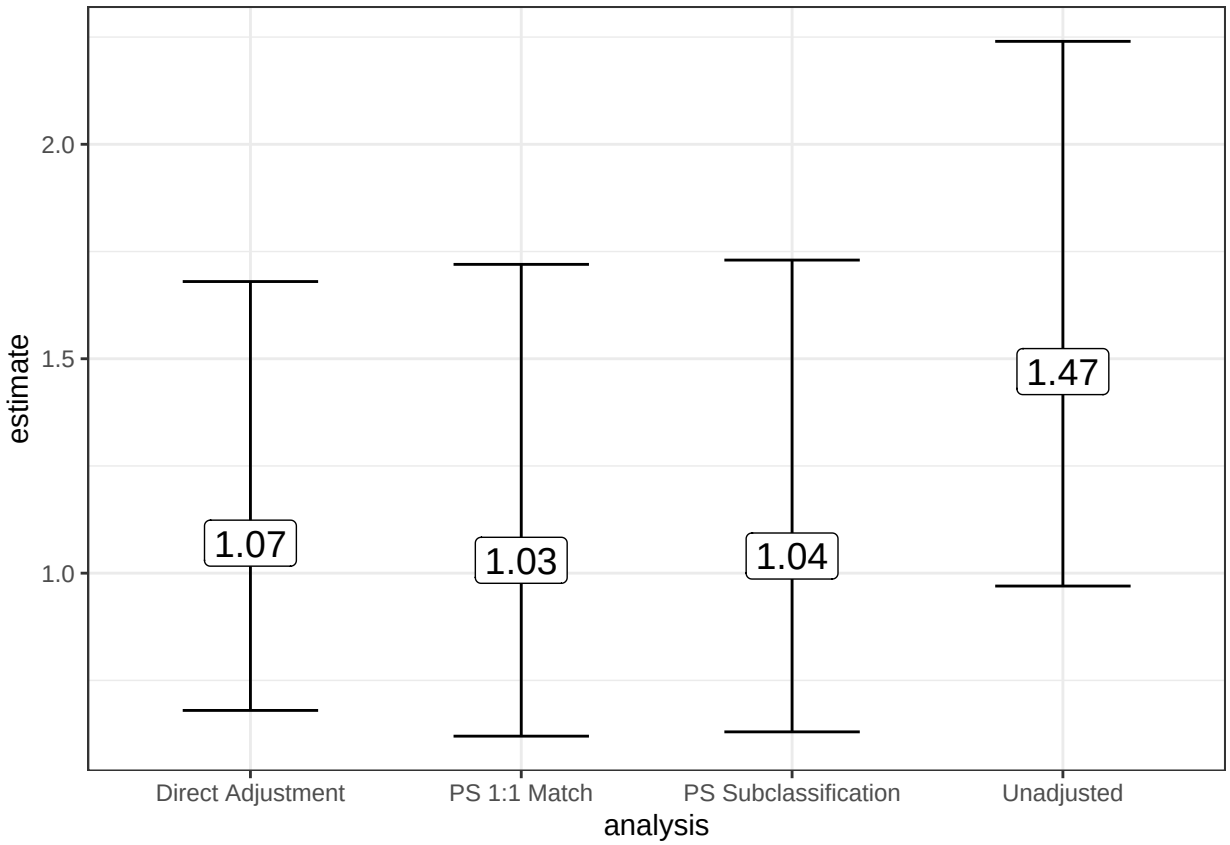
Estimating the **intervention effect** on the `hospice` outcome, we have yet to find a statistically significant result at the 5% significance level.

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)

7.1 Building a Data Frame of the Results

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

```
res_hospice <- tibble(
    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()
```


2.0
1.5
1.0
estimate
1.07
1.03
1.04
1.47
Direct Adjustment
PS 1:1 Match
PS Subclassification
Unadjusted
analysis