Homework 4

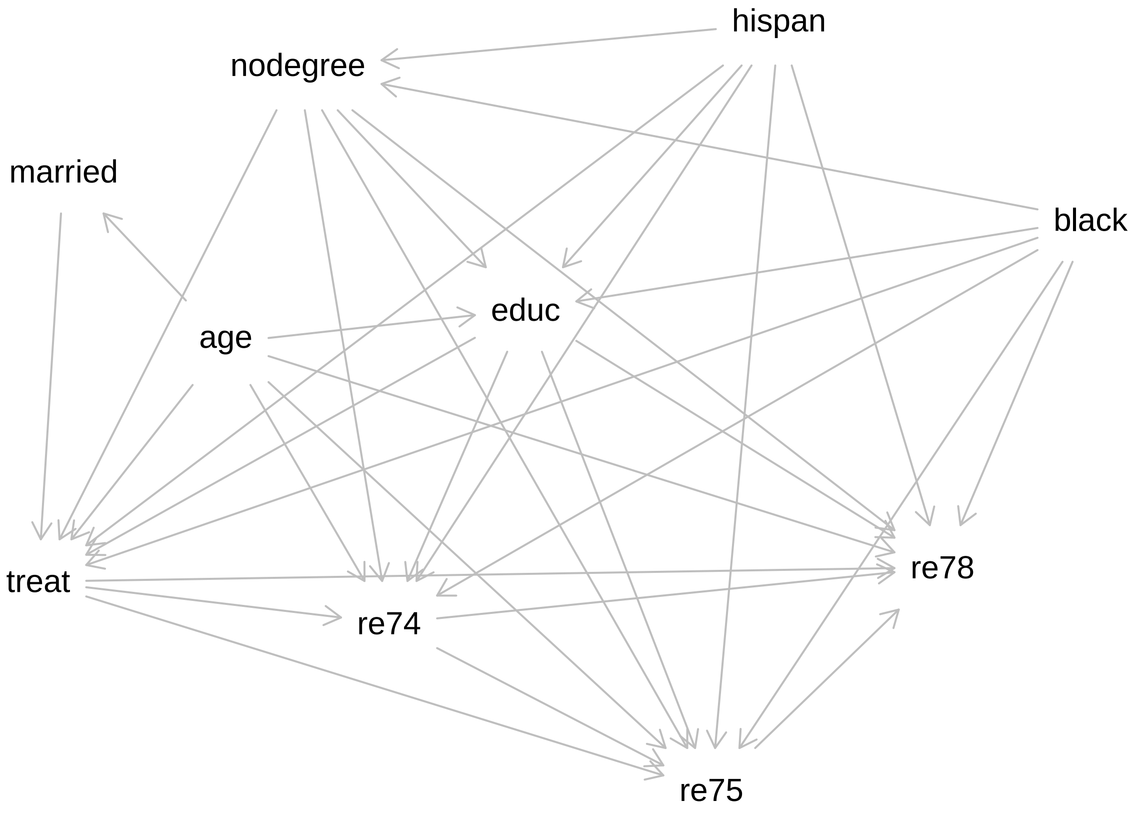
Yuki Joyama

```r
# libraries
library(tidyverse)
library(ggplot2)
library(dagitty)
library(tableone)
library(knitr)
library(personalized)

# setup plot theme
theme_set(
  theme_minimal() +
    theme(legend.position = "top")
  )
```

```r
# import data
df = read_csv("./data/hw4_data.csv") |>
  mutate(
    treat = as.factor(treat),
    black = as.factor(black),
    hispan = as.factor(hispan),
    married = as.factor(married),
    nodegree = as.factor(nodegree)
  )
```

1. DAG

```r
g = dagitty('dag {
age [pos="-1.842,-0.039"]
black [pos="1.458,-0.317"]
educ [pos="-0.698,-0.114"]
hispan [pos="0.269,-0.760"]
married [pos="-2.461,-0.426"]
nodegree [pos="-1.567,-0.659"]
re74 [pos="-1.219,0.594"]
re75 [pos="0.011,0.971"]
re78 [outcome,pos="0.893,0.468"]
treat [exposure,pos="-2.558,0.500"]
age -> educ
age -> married
age -> re74
age -> re75
age -> re78
age -> treat
black -> educ
black -> nodegree
black -> re74
black -> re75
black -> re78
black -> treat
educ -> re74
educ -> re75
educ -> re78
educ -> treat
hispan -> educ
hispan -> nodegree
hispan -> re74
hispan -> re75
hispan -> re78
hispan -> treat
married -> treat
nodegree -> educ
nodegree -> re74
nodegree -> re75
nodegree -> re78
nodegree -> treat
re74 -> re75
re74 -> re78
re75 -> re78
treat -> re74
treat -> re75
treat -> re78
}')
plot(g)
```



Note of the variables in the DAG:
`treat`: treatment assignment (job training); exposure
- Suppose that the job training was completed before 1974, `treat` is likely to infludence `re74`, `re75` and `re78`
`age`: age in years
- `age` may affect `married`, `treat`, `educ` and all the income status (`re74`, `re75`, `re78`)
`educ`: education in years
- `educ` may affect all the income status and `treat`
`black`, `hispan`: indicators for African American and hispanic
- Both of the ethnicity indicator may affect all the income status, `nodegree`, `treat`, and `educ`
`married`: indicator for married
- `married` could influence `treat`
`nodegree`: indicator for highschool degree
- `nodegree` may affect `treat`, `educ`, and all the income status
`re74`: income in 1974
- `re74` can influence `re75` and `re78`
`re75`: income in 1975
- `re75` can influence `re78`
`re78`: income in 1978; outcome

Given the DAG, covariates that need to be adjusted in investigating the effect of exposure on the outcome are `nodegree`, `hispan`, `black`, `educ` and `age`.

2. I will evaluate the covariate balance using standardized mean differences.

```r
cov = c("nodegree", "hispan", "black", "educ", "age")

# construct a table
tab = CreateTableOne(vars = cov, strata = "treat", data = df, test = FALSE)
print(tab, smd = TRUE)
```

```
##                      Stratified by treat
##                       0             1             SMD   
##   n                     429           185               
##   nodegree = 1 (%)      256 (59.7)    131 (70.8)   0.235
##   hispan = 1 (%)         61 (14.2)     11 ( 5.9)   0.277
##   black = 1 (%)          87 (20.3)    156 (84.3)   1.671
##   educ (mean (SD))    10.24 (2.86)  10.35 (2.01)   0.045
##   age (mean (SD))     28.03 (10.79) 25.82 (7.16)   0.242
```

We can see that most covariates have SMD > 0.1 except for `educ`, indicating the potential imbalance between the two treatment group.

3. Propensity score estimates are calculated by fitting a logistic regression.

```r
# fit PS model
ps.fit <- glm(treat ~ nodegree + hispan + black + educ + age, family = "binomial", data = df)

ps.fit |>
  broom::tidy() |>
  kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	-4.6689752	1.0063696	-4.6394238	0.0000035
nodegree1	0.7812141	0.3303690	2.3646714	0.0180461
hispan1	1.0760070	0.4168588	2.5812269	0.0098450
black1	3.3041742	0.2811462	11.7525129	0.0000000
educ	0.1523227	0.0643134	2.3684437	0.0178631
age	-0.0054560	0.0125728	-0.4339522	0.6643232

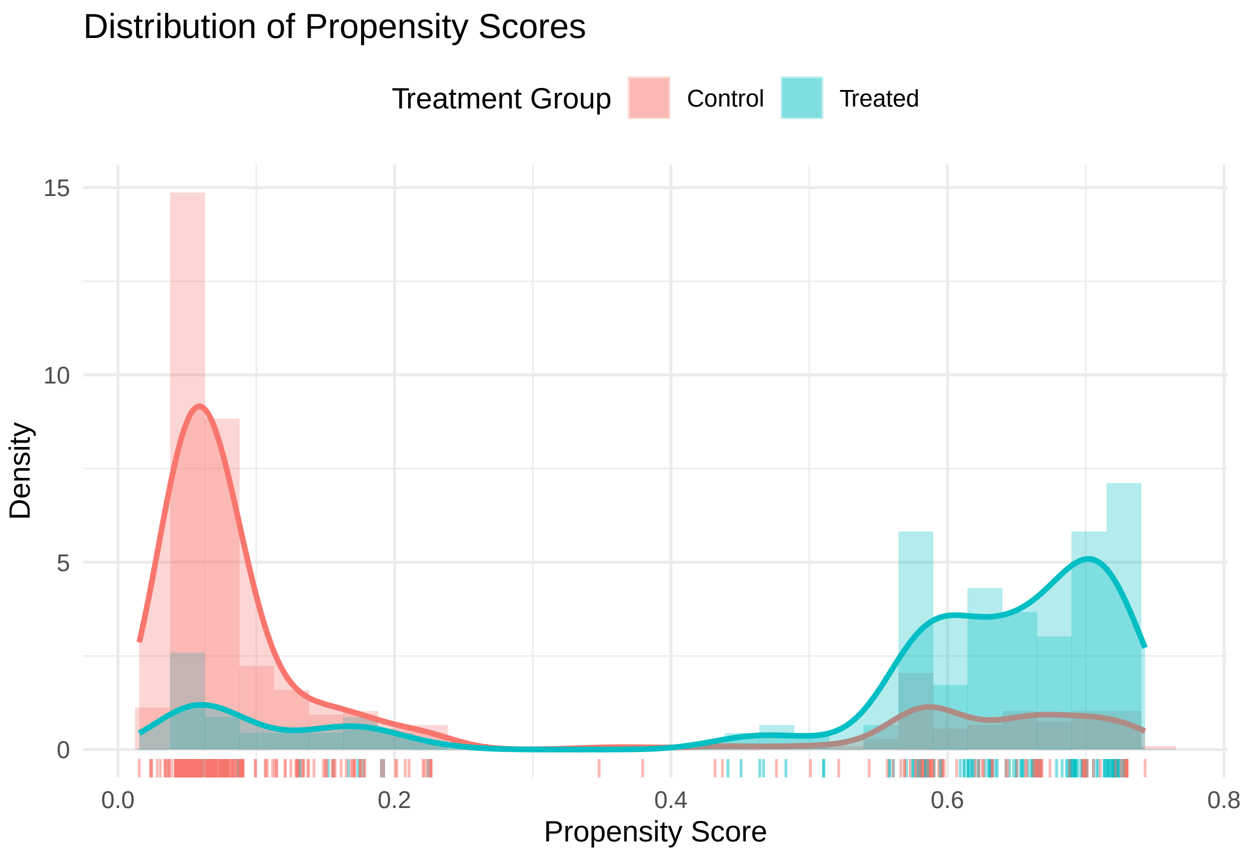
```r
# estimate PS
df.ps <- predict(ps.fit, type = 'response')
print(df.ps[1:50])
```

```
##          1          2          3          4          5          6          7 
## 0.70890885 0.17352013 0.57429368 0.72003809 0.61181465 0.66089500 0.58360334 
##          8          9         10         11         12         13         14 
## 0.71450603 0.72158352 0.04647944 0.66455356 0.62265153 0.63106456 0.68833095 
##         15         16         17         18         19         20         21 
## 0.59626339 0.69761702 0.61492967 0.69299354 0.56090387 0.57962038 0.72441625 
##         22         23         24         25         26         27         28 
## 0.05951577 0.06156125 0.72332569 0.69876670 0.71783314 0.72223248 0.19219049 
##         29         30         31         32         33         34         35 
## 0.69991390 0.72332569 0.69991390 0.44122686 0.72223248 0.58757541 0.58094921 
##         36         37         38         39         40         41         42
## 0.63147843 0.51040509 0.58360334 0.59484193 0.69183153 0.58625263 0.06782935
##         43         44         45         46         47         48         49
## 0.63233393 0.15559436 0.72767188 0.72223248 0.63233393 0.66698171 0.51040509
##         50
## 0.58360334
```

Listed values are the propensity scores for each observation in the dataset (only showing the first 50 observations out of 614).

4.

```r
# append ps estimates to dataset
df <- df |>
  mutate(
    ps.est = df.ps
  )

prop.func <- function(x, trt)
{
  # propensity score model
  propens.model <- glm(as.factor(treat) ~ as.factor(nodegree) + as.factor(hispan) + as.factor(black) + 
  pi.x <- predict(propens.model, type = "response")
  pi.x
}

# histogram and density plot to check the overlap
ggplot(df, aes(x = ps.est, fill = as.factor(treat))) +
  geom_histogram(aes(y = ..density..), bins = 30, alpha = 0.3, position = "identity") +
  geom_density(aes(color = as.factor(treat)), alpha = 0.3, size = 1) +  # add density plot
  scale_fill_manual(values = c("#F8766D", "#00BFC4"), name = "Treatment Group", labels = c("Control", "
                    guide = guide_legend(override.aes = list(color = NA))) +
  scale_color_manual(values = c("#F8766D", "#00BFC4"), guide = "none") +
  geom_rug(aes(color = as.factor(treat)), sides = "b", alpha = 0.5) +  # add rug plot
  labs(title = "Distribution of Propensity Scores",
       x = "Propensity Score",
       y = "Density")
```



The histogram shows that the propensity score estimates does not overlap in ranges approximately [0, 0.0375] and [0.7375, 1]. I will trim the samples whose PS estimate is in these ranges.

```
# trimming
# eliminate controls for whom the P(A=1|C) is less that the min(P(A=1|C)) found in the treated group
min(df$ps.est[df$treat == 1])
```

```
## [1] 0.04528526
```

```
head(df$ps.est[which(df$treat == 0)] <= min(df$ps.est[df$treat == 1]))
```

```
##   186   187   188   189   190   191
## FALSE FALSE FALSE FALSE FALSE FALSE
```

```
length(df$ps.est[which(df$treat == 0)]<= min(df$ps.est[df$treat == 1]))
```

```
## [1] 429
```

```
# eliminate treated for whom the P(A=1|C) is greater that the max(P(A=1|C)) found in the control group
max(df$ps.est[df$treat == 0])
```

```
## [1] 0.7428825
```

```r
head(df$ps.est[which(df$treat == 1)]>= max(df$ps.est[df$treat == 0]))
```

```
##     1     2     3     4     5     6 
## FALSE FALSE FALSE FALSE FALSE FALSE
```

```r
length(df$ps.est[which(df$treat == 1)]>= max(df$ps.est[df$treat == 0]))
```

```
## [1] 185
```

```r
df_trim = df[df$ps.est >= min(df$ps.est[df$treat == 1]) & df$ps.est <= max(df$ps.est[df$treat == 0]),]

dim(df_trim)
```

```
## [1] 571  12
```

After the trimming, sample size went down from 614 to 571, which may lead to reduced statistical power. In terms of the generalizability, trimming will disable the external validity because we are only looking at individuals with non-extreme propensity scores.

5. Here I will reevaluate the covariate balance using the trimmed sample.

```r
# construct a table
tab_trim = CreateTableOne(vars = cov, strata = "treat", data = df_trim, test = FALSE)
print(tab_trim, smd = TRUE)
```

```
##                    Stratified by treat
##                     0             1             SMD   
##   n                   386           185               
##   nodegree = 1 (%)   232 (60.1)    131 (70.8)    0.227
##   hispan = 1 (%)      60 (15.5)     11 ( 5.9)    0.314
##   black = 1 (%)       87 (22.5)    156 (84.3)    1.578
##   educ (mean (SD)) 10.47 (2.63)  10.35 (2.01)    0.054
##   age (mean (SD))  26.21 (9.28)  25.82 (7.16)    0.047
```

Compared to untrimmed data, the balance of `nodegree`, `black` and `age` appear to be improved while `educ` and `hispan` appear to be worsened.

6. Subclassification
   First I will refit the model using the trimmed data and recheck the overlap.

```r
# fit PS model
ps.fit_trim <- glm(treat ~ nodegree + hispan + black + educ + age, family = "binomial", data = df_trim)

ps.fit_trim |>
  broom::tidy() |>
  kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	-4.4833461	1.0137243	-4.4226485	0.0000097
nodegree1	0.7219629	0.3302166	2.1863314	0.0287914
hispan1	0.9709541	0.4176188	2.3249772	0.0200732
black1	3.1925557	0.2824490	11.3031244	0.0000000
educ	0.1370930	0.0652493	2.1010666	0.0356351
age	-0.0008086	0.0129672	-0.0623585	0.9502774

```
# estimate PS
df.ps_trim <- predict(ps.fit_trim, type = 'response')
print(df.ps_trim[1:50])
```

```
##          1          2          3          4          5          6          7
## 0.71285226 0.17159272 0.58177585 0.71450459 0.62274896 0.65638382 0.58315243
##          8          9         10         11         12         13         14
## 0.71367914 0.70783841 0.05391550 0.65693075 0.61643370 0.62559424 0.68573839
##         15         16         17         18         19         20         21
## 0.59316802 0.68713077 0.61528590 0.68643500 0.57980710 0.58256263 0.71516392
##         22         23         24         25         26         27         28
## 0.06931906 0.07187265 0.71499917 0.68730458 0.71417458 0.71483437 0.19133929
##         29         30         31         32         33         34         35
## 0.68747834 0.71499917 0.68747834 0.48518715 0.71483437 0.58374200 0.58275926
##         36         37         38         39         40         41         42
## 0.64440855 0.52408694 0.58315243 0.62027619 0.68626093 0.58354550 0.07290428
##         43         44         45         46         47         48         49
## 0.62578362 0.15339217 0.71565781 0.71483437 0.62578362 0.65729513 0.52408694
##         50
## 0.58315243
```

I will create three strata. The breaks are shown below.

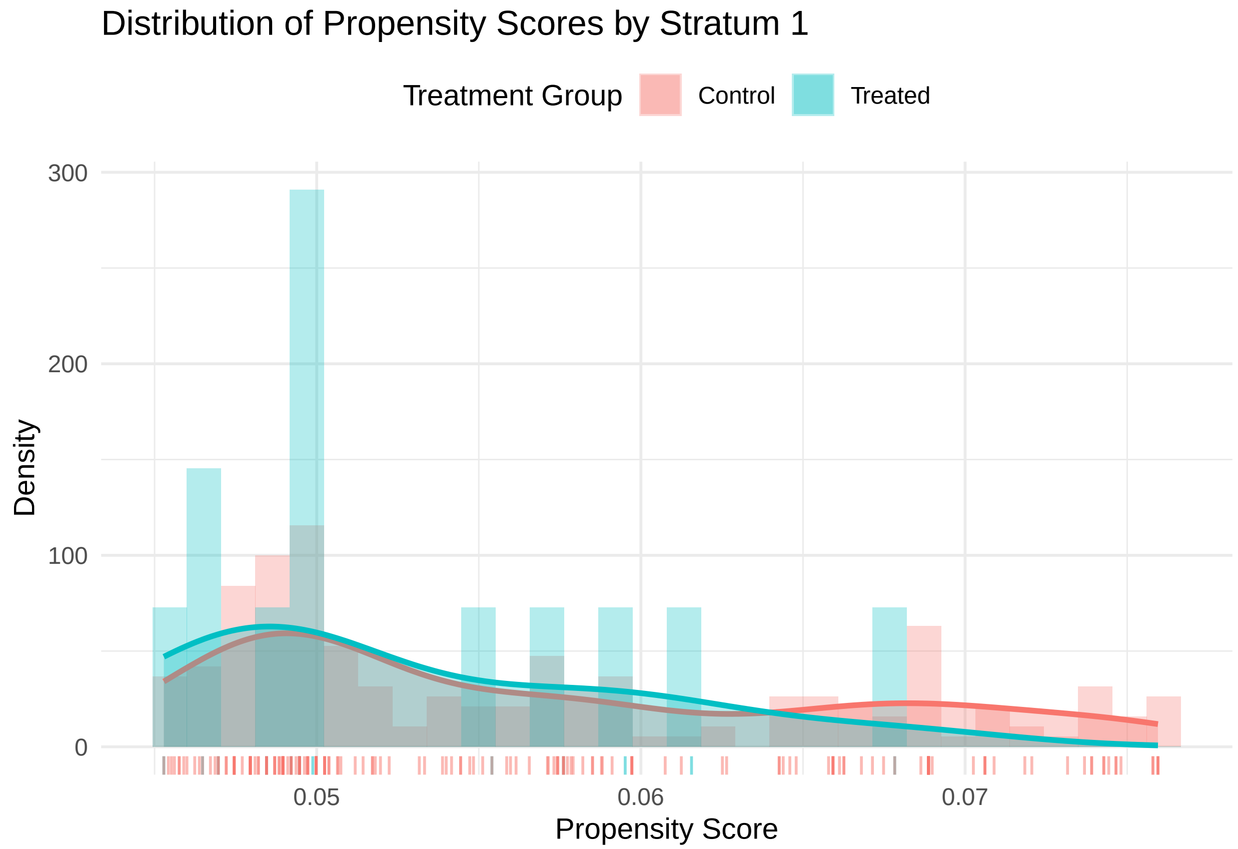
```
df_trim = df_trim |>
  mutate(ps.strata = gtools::quantcut(ps.est, 3))

# breaks
quantile(df_trim$ps.est, probs = seq(0, 1, length.out = 4))
```

```
##         0%  33.33333%  66.66667%       100%
## 0.04528526 0.07594860 0.58492860 0.74288249
```

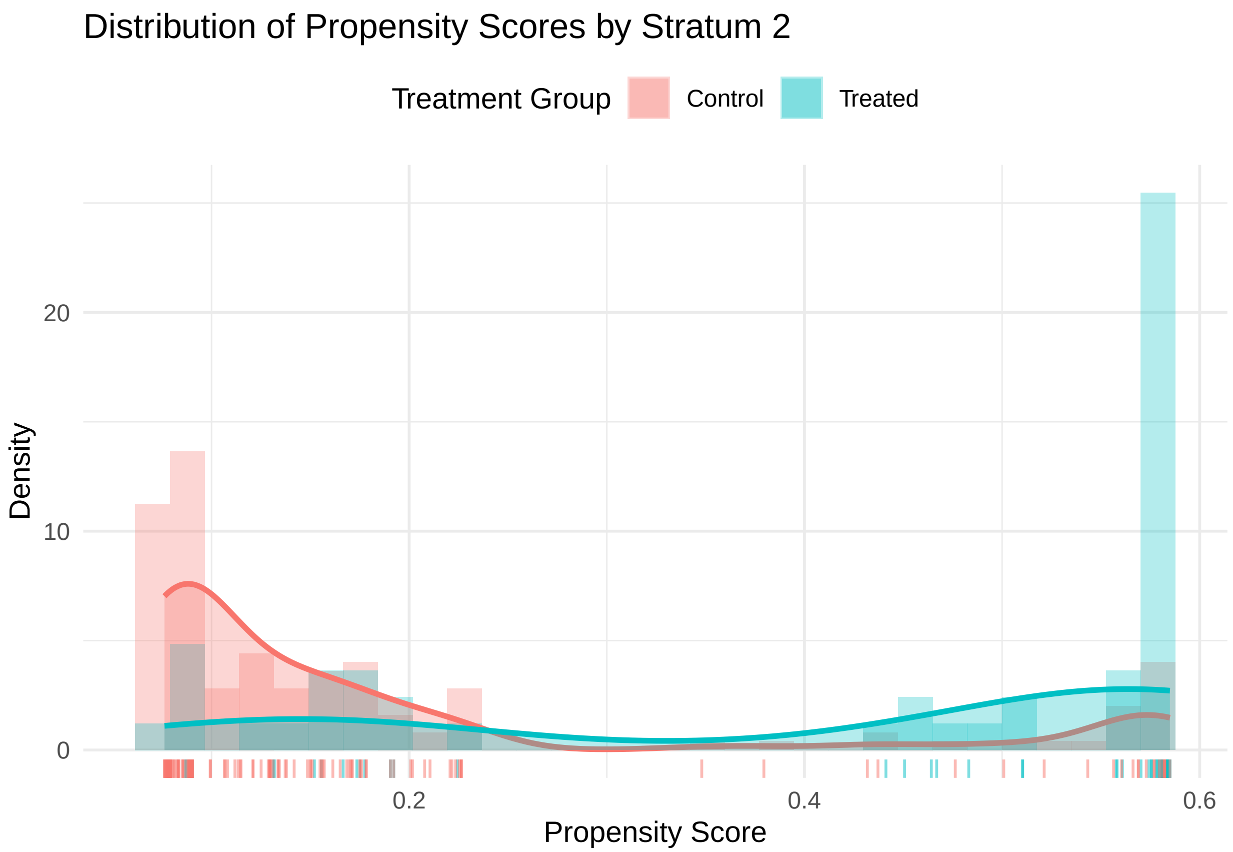
```
breaks = quantile(df_trim$ps.est, probs = seq(0, 1, length.out = 4))

# save dataset for each strata
df_trim1 = df_trim |>
  filter(ps.strata == "[0.0453,0.0759]")

df_trim2 = df_trim |>
  filter(ps.strata == "(0.0759,0.585]")

df_trim3 = df_trim |>
  filter(ps.strata == "(0.585,0.743]")
```

Below is the histogram and density plot of the propensity scores in each stratum.

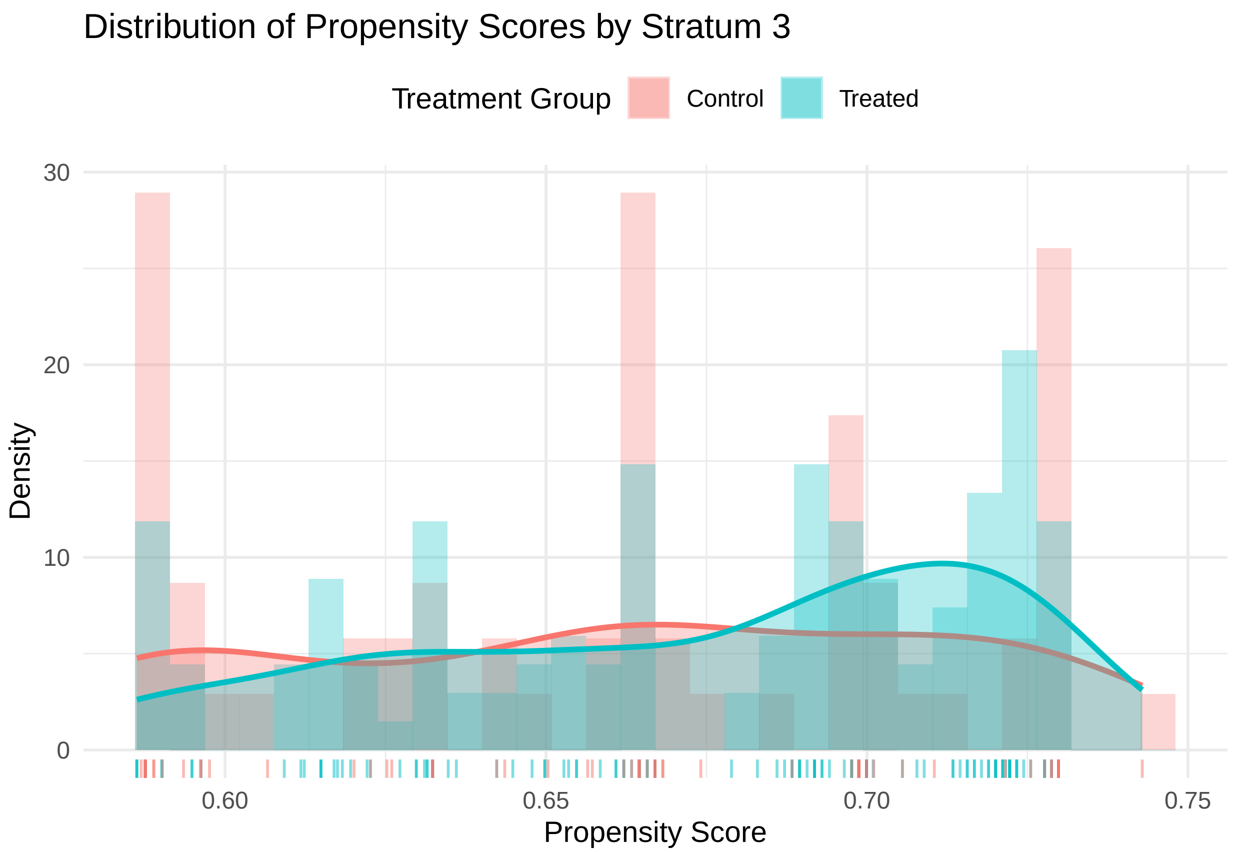
```
# plot propensity scores with subclass boundaries by treatment group
# [0.0453,0.0759]
ggplot(df_trim1, aes(x = ps.est, fill = as.factor(treat))) +
  geom_histogram(aes(y = ..density..), bins = 30, alpha = 0.3, position = "identity") +
  geom_density(aes(color = as.factor(treat)), alpha = 0.3, size = 1) +  # add density plot
  scale_fill_manual(values = c("#F8766D", "#00BFC4"), name = "Treatment Group", labels = c("Control", "
                    guide = guide_legend(override.aes = list(color = NA))) +
  scale_color_manual(values = c("#F8766D", "#00BFC4"), guide = "none") +
  geom_rug(aes(color = as.factor(treat)), sides = "b", alpha = 0.5) +  # add rug plot
  labs(title = "Distribution of Propensity Scores by Stratum 1",
       x = "Propensity Score",
       y = "Density")
```



```
# (0.0759,0.585]
ggplot(df_trim2, aes(x = ps.est, fill = as.factor(treat))) +
  geom_histogram(aes(y = ..density..), bins = 30, alpha = 0.3, position = "identity") +
  geom_density(aes(color = as.factor(treat)), alpha = 0.3, size = 1) +  # add density plot
  scale_fill_manual(values = c("#F8766D", "#00BFC4"), name = "Treatment Group", labels = c("Control", "
                    guide = guide_legend(override.aes = list(color = NA))) +
  scale_color_manual(values = c("#F8766D", "#00BFC4"), guide = "none") +
  geom_rug(aes(color = as.factor(treat)), sides = "b", alpha = 0.5) +  # add rug plot
  labs(title = "Distribution of Propensity Scores by Stratum 2",
       x = "Propensity Score",
       y = "Density")
```



```
# (0.585,0.743]
ggplot(df_trim3, aes(x = ps.est, fill = as.factor(treat))) +
  geom_histogram(aes(y = ..density..), bins = 30, alpha = 0.3, position = "identity") +
  geom_density(aes(color = as.factor(treat)), alpha = 0.3, size = 1) +  # add density plot
  scale_fill_manual(values = c("#F8766D", "#00BFC4"), name = "Treatment Group", labels = c("Control", "
                    guide = guide_legend(override.aes = list(color = NA))) +
  scale_color_manual(values = c("#F8766D", "#00BFC4"), guide = "none") +
  geom_rug(aes(color = as.factor(treat)), sides = "b", alpha = 0.5) +  # add rug plot
  labs(title = "Distribution of Propensity Scores by Stratum 3",
       x = "Propensity Score",
       y = "Density")
```



Now, let's inspect the covariate balance in each stratum.

```
# stratum 1
tab_trim1 = CreateTableOne(vars = cov, strata = "treat", data = df_trim1, test = FALSE)
print(tab_trim1, smd = TRUE)
```

```
##                      Stratified by treat
##                       0             1             SMD
##   n                     180            13
##   nodegree = 1 (%)      80 (44.4)       4 (30.8)   0.285
##   hispan = 1 (%)         4 ( 2.2)       0 ( 0.0)   0.213
##   black = 1 (%)          0 ( 0.0)       0 ( 0.0)  <0.001
##   educ (mean (SD)) 10.64 (2.68)  11.08 (2.10)   0.180
##   age (mean (SD))  27.16 (8.63)  27.85 (7.51)   0.085
```

```
# stratum 2
tab_trim2 = CreateTableOne(vars = cov, strata = "treat", data = df_trim2, test = FALSE)
print(tab_trim2, smd = TRUE)
```

```
##                      Stratified by treat
##                       0             1             SMD
##   n                     142            47
##   nodegree = 1 (%)     105 (73.9)      22 (46.8)   0.577
##   hispan = 1 (%)        56 (39.4)      11 (23.4)   0.351
##   black = 1 (%)         23 (16.2)      31 (66.0)   1.172
##   educ (mean (SD)) 10.24 (2.77)  10.09 (2.76)   0.056
##   age (mean (SD))  25.98 (9.63)  26.38 (6.96)   0.048
```

```
# stratum 3
tab_trim3 = CreateTableOne(vars = cov, strata = "treat", data = df_trim3, test = FALSE)
print(tab_trim3, smd = TRUE)
```

```
##                     Stratified by treat
##                      0              1              SMD
##   n                   64             125
##   nodegree = 1 (%)    47 ( 73.4)    105 ( 84.0)   0.260
##   hispan = 1 (%)       0 (  0.0)      0 (  0.0)  <0.001
##   black = 1 (%)       64 (100.0)    125 (100.0)  <0.001
##   educ (mean (SD)) 10.50 (2.09)    10.37 (1.63)    0.070
##   age (mean (SD))  24.06 (10.01)  25.39 (7.20)    0.152
```

Stratum 1 and 3 show acceptable balance for most covariates with `nodegree` being the most imbalanced one. Stratum 2 still shows significant imbalance, especially for the covariate `black` and `nodegree`. This may be due to the limited propensity score distribution within the range of (0.0759, 0.585].

7. I will calculate the MACE of participation in a job training on wages using the following codes.

```
# calculate point estimate and confidence intervals
df_trim |>
  group_by(ps.strata) |> # group by PS strata only
  summarise(
    mean.Y.0 = mean(re78[treat == 0]),
    mean.Y.1 = mean(re78[treat == 1]),
    var.Y.0 = var(re78[treat == 0]),
    var.Y.1 = var(re78[treat == 1]),
    n.0 = sum(treat == 0),
    n.1 = sum(treat == 1)
  ) |>
  mutate(
    ACE.strata = mean.Y.1 - mean.Y.0, # stratum-specific ACEs
    prop = (n.0 + n.1)/nrow(df_trim), # proportion of each stratum
    MACE = round(sum(ACE.strata * prop), 2), # point estimate of MACE
    # variance of a stratified estimator for a weighted mean difference across strata
    varace = sum((prop^2)*(var.Y.0/n.0 + var.Y.1/n.1)),
    ci.l = MACE - qnorm(1 - 0.05/2) * sqrt(varace), # CI lower limit
    ci.u = MACE + qnorm(1 - 0.05/2) * sqrt(varace), # CI upper limit
    z_score = MACE / sqrt(varace),
    "P Value" = round(2 * (1 - pnorm(abs(z_score))), 2), # calculate p-value
    "95%CI" = paste0("(", round(ci.l, 0), ", ", round(ci.u, 0), ")")
  ) |>
  select(c(MACE, "95%CI", "P Value")) |>
  ungroup() |>
  distinct() |>
  kable()
```

MACE	95%CI	P Value
33.21	(-1371, 1437)	0.96

The point estimate of MACE appears to be very small and both 95% CI and p-value >0.05 implies that the job training program did not have a statistically significant effect on participants' income in 1978 in the trimmed sample.

8. I will use construct linear regression model to directly adjust for the confounders.

```r
# linear model
df |>
  lm(re78 ~ treat + nodegree + hispan + black + educ + age, data = _) |>
  broom::tidy() |>
  kable()
```

term	estimate	std.error	statistic	p.value
(Intercept)	-230.67883	2538.70487	-0.0908648	0.9276300
treat1	829.25814	810.47146	1.0231799	0.3066304
nodegree1	-81.86415	888.69177	-0.0921176	0.9266350
hispan1	469.40252	984.85328	0.4766218	0.6338030
black1	-1984.02901	791.90107	-2.5054001	0.0124921
educ	494.65987	165.28327	2.9928006	0.0028766
age	90.48256	31.15853	2.9039418	0.0038190

This is still conditinal effects within the strata of observed covariates. I will calculate the marginal effect by standardization as follows:

```r
# predict outcome values
lm.fit = lm(re78 ~ treat + nodegree + hispan + black + educ + age, data = df)
df$pred <- predict(lm.fit, newdata = df)

# MACE estimate
# difference between mean predicted values for rows with A=1 and mean predicted values for rows with A
df |>
  group_by(treat) |>
  summarise(
    mean.Y = mean(pred)
  ) |>
  pivot_wider(
    names_from = treat,
    names_glue = "mean.Y.{treat}", values_from = mean.Y
  ) |>
  mutate(
    MACE = mean.Y.1-mean.Y.0, # calculate MACE
    n.0 = sum(df$treat == 0),
    n.1 = sum(df$treat == 1),
    var.Y.0 = var(df$pred[df_trim$treat == 0]),
    var.Y.1 = var(df$pred[df_trim$treat == 1]),
    se = sqrt(var.Y.0/n.0 + var.Y.1/n.1), # standard error
    ci.l = MACE - qnorm(1 - 0.05/2) * sqrt(se), # CI lower limit
    ci.u = MACE + qnorm(1 - 0.05/2) * sqrt(se), # CI upper limit
    z_score = MACE / sqrt(se),
    "P Value" = 2 * (1 - pnorm(abs(z_score))), # calculate p-value
    "95%CI" = paste0("(", round(ci.l, 0), ", ", round(ci.u, 0), ")")
  ) |>
  select(c(MACE, "95%CI", "P Value")) |>
  ungroup() |>
  distinct() |>
  kable()
```

MACE	95%CI	P Value
-635.0262	(-659, -611)	0

From the results, we observe that `treat = 1` is associated with $635.03 less income than `treat = 0`. The 95% CI and p-value <0.05 indicate statistical significance, suggesting a negative impact of the job training program on income in 1978. This finding contrasts with the subclassification approach. It's possible that residual imbalance remains in the subclassification model due to a limited number of strata, which could bias the results. Alternatively, the linear regression model may suffer from violations of modeling assumptions. Further investigation is necessary to determine the optimal model to minimize bias in both approaches.

9.

Subclassification Approach
Advantages: Because the modeling decisions come before looking at outcome data, it work against p-hacking. Also, the outcome model makes fewer modeling assumption than regression based approach conditioning on many covariates. Visualizing a potential positivity violation is helpful for reducing the outcome bias.

Disadvantages: In order to balancing the covariates between two exposure groups, we often sacrifice the sample size, so the power can go down.

Regression Based Approach
Advantages: When the number of confounders is small, this approach requires less calculation and could be more efficient.

Disadvantages: As mentioned above, p-hacking could be problematic. When there are too many confounders compared to the number of events, by the rule of thumb, this approach is not recommended.