Code appendix

DataFrame

Sample data frame

```
library(readr)
library(pacman)
p_load(MatchIt, dplyr, survey, tableone, twang, ipw, ggplot2)
working_data <- read.csv('/cloud/project/data/third_grade_data_cleaned.csv')
working_data <- working_data %>%
  filter(Demographic.Variable == "SWD")
head(working_data)
```

```
##    X    DBN Percent_Attendance Demographic.Variable X..Poverty   borough
## 1  6 01M015               92.3                  SWD      0.847 Manhattan
## 2 13 01M019               91.5                  SWD      0.770 Manhattan
## 3 18 01M020               91.9                  SWD      0.736 Manhattan
## 4 30 01M034               88.9                  SWD      0.979 Manhattan
## 5 41 01M063               92.2                  SWD      0.818 Manhattan
## 6 43 01M064               87.7                  SWD      0.922 Manhattan
##   self_contained_option gifted_talented_option X..Male X..Black
## 1                     0                       1   0.479    0.274
## 2                     1                       0   0.556    0.191
## 3                     1                       0   0.509    0.103
## 4                     0                       0   0.550    0.318
## 5                     0                       0   0.507    0.182
## 6                     0                       0   0.588    0.208
##   Economic.Need.Index Percent_Chronically_Absent
## 1               0.890                      28.6
## 2               0.679                      31.6
## 3               0.800                      42.9
## 4               0.937                      45.0
## 5               0.762                      28.6
## 6               0.882                      54.5
```

Descriptive Stats

```
working_data %>%
  group_by(self_contained_option) %>%
  summarise(mean_attendance = mean(Percent_Attendance),
            mean_chronic_absent = mean(Percent_Chronically_Absent))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   self_contained_option mean_attendance mean_chronic_absent
##                   <int>           <dbl>               <dbl>
```

```
## 1                    0           93.0                24.2
## 2                    1           91.4                32.5
```

```
with(working_data, t.test(Percent_Attendance ~ self_contained_option))
```

```
##
##  Welch Two Sample t-test
##
## data:  Percent_Attendance by self_contained_option
## t = 6.8404, df = 394.4, p-value = 3.024e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.122526 2.028028
## sample estimates:
## mean in group 0 mean in group 1
##        93.00936        91.43408
```

```
with(working_data, t.test(Percent_Chronically_Absent ~ self_contained_option))
```

```
##
##  Welch Two Sample t-test
##
## data:  Percent_Chronically_Absent by self_contained_option
## t = -5.884, df = 351.63, p-value = 9.335e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -11.022223  -5.499767
## sample estimates:
## mean in group 0 mean in group 1
##        24.21429        32.47528
```

```
table(working_data$self_contained_option)
```

```
##
##   0   1
## 203 534
```

```
## the t test is statistically signifiant, but that
# is done without any matching

# find the diffrences on the covariates
school_covariates <- c('X..Poverty', 'X..Black', 'X..Male', 'Economic.Need.Index')
working_data %>%
  group_by(self_contained_option) %>%
  select(one_of(school_covariates)) %>%
  summarise_all(funs(mean(., na.rm=T)))
```

```
## Adding missing grouping variables: `self_contained_option`
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
```

```
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## # A tibble: 2 x 5
##   self_contained_option X..Poverty X..Black X..Male Economic.Need.Index
##                   <int>      <dbl>    <dbl>   <dbl>               <dbl>
## 1                     0      0.632    0.181   0.506               0.609
## 2                     1      0.804    0.281   0.514               0.765
```

```
# do a t.test to find if there are statistically different differences

lapply(school_covariates, function(v) {
  t.test(working_data[, v] ~ working_data[ ,'self_contained_option'])
})
```

```
## [[1]]
##
##  Welch Two Sample t-test
##
## data:  working_data[, v] by working_data[, "self_contained_option"]
## t = -7.8428, df = 261.44, p-value = 1.124e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2149956 -0.1287040
## sample estimates:
## mean in group 0 mean in group 1
##       0.6322906       0.8041404
##
##
## [[2]]
##
##  Welch Two Sample t-test
##
## data:  working_data[, v] by working_data[, "self_contained_option"]
## t = -5.072, df = 444.37, p-value = 5.791e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.13913838 -0.06142434
## sample estimates:
## mean in group 0 mean in group 1
##       0.1806700       0.2809513
##
##
## [[3]]
##
##  Welch Two Sample t-test
##
## data:  working_data[, v] by working_data[, "self_contained_option"]
## t = -3.8629, df = 359.18, p-value = 0.0001329
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.012454552 -0.004051417
```

```
## sample estimates:
## mean in group 0 mean in group 1
##       0.5056946       0.5139476
##
##
## [[4]]
##
##  Welch Two Sample t-test
##
## data:  working_data[, v] by working_data[, "self_contained_option"]
## t = -7.2326, df = 278.27, p-value = 4.602e-12
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1988636 -0.1137724
## sample estimates:
## mean in group 0 mean in group 1
##       0.6091576       0.7654757
```

Matching

Nearest Neighbor Matching

```
# set seed
set.seed(1731)
## issue with the matching, so what if we switch treatment and control
working_data$self_contained_binary <- ifelse(working_data$self_contained_option == 1, 0, 1)
working_data$new_outcome_labelled <- ifelse(working_data$self_contained_binary == 1, "no SC", "SC")
library(MatchIt)
school_nearest <- matchit(formula = self_contained_binary ~ Economic.Need.Index +
                            X..Black + X..Male + X..Poverty, data = working_data,
        method = "nearest",
        family = "binomial",
        caliper = 0.25)
summary(school_nearest)
```

```
##
## Call:
## matchit(formula = self_contained_binary ~ Economic.Need.Index +
##     X..Black + X..Male + X..Poverty, data = working_data, method = "nearest",
##     family = "binomial", caliper = 0.25)
##
## Summary of balance for all data:
##                       Means Treated Means Control SD Control Mean Diff eQQ Med
## distance                     0.3746        0.2377     0.1239    0.1369  0.1085
## Economic.Need.Index          0.6092        0.7655     0.1955   -0.1563  0.1560
## X..Black                     0.1807        0.2810     0.2757   -0.1003  0.0800
## X..Male                      0.5057        0.5139     0.0256   -0.0083  0.0080
## X..Poverty                   0.6323        0.8041     0.1785   -0.1718  0.1430
##                       eQQ Mean eQQ Max
## distance                0.1369  0.3268
## Economic.Need.Index     0.1550  0.3450
## X..Black                0.1002  0.3150
## X..Male                 0.0086  0.0650
## X..Poverty              0.1705  0.4090
```

```
##
##
## Summary of balance for matched data:
##                       Means Treated Means Control SD Control Mean Diff eQQ Med
## distance                     0.3215        0.3122     0.1564    0.0092  0.0058
## Economic.Need.Index          0.6689        0.6757     0.2412   -0.0069  0.0100
## X..Black                     0.1983        0.2004     0.2456   -0.0021  0.0160
## X..Male                      0.5081        0.5090     0.0216   -0.0010  0.0030
## X..Poverty                   0.6998        0.7081     0.2311   -0.0082  0.0080
##                       eQQ Mean eQQ Max
## distance                0.0110  0.0399
## Economic.Need.Index     0.0162  0.1040
## X..Black                0.0179  0.0880
## X..Male                 0.0039  0.0420
## X..Poverty              0.0184  0.1240
##
## Percent Balance Improvement:
##                       Mean Diff. eQQ Med eQQ Mean eQQ Max
## distance                 93.2678 94.6852  91.9739 87.7799
## Economic.Need.Index      95.6123 93.5897  89.5455 69.8551
## X..Black                 97.9493 80.0000  82.1404 72.0635
## X..Male                  88.0201 62.5000  54.3730 35.3846
## X..Poverty               95.2166 94.4056  89.1973 69.6822
##
## Sample sizes:
##           Control Treated
## All           534     203
## Matched       177     177
## Unmatched     357      26
## Discarded       0       0
```

```
#create the matched set
nearest_matched <- match.data(school_nearest)
#350 schools were matched
# dim(nearest_matched)
## now look at the means of the covariates
#matching was successful because the poverty rates are around .72 now together
nearest_matched %>%
  group_by(self_contained_option) %>%
  select(X..Poverty) %>%
  summarise_all(funs(mean))
```

```
## Adding missing grouping variables: `self_contained_option`
```

```
## # A tibble: 2 x 2
##   self_contained_option X..Poverty
##                   <int>      <dbl>
## 1                     0      0.700
## 2                     1      0.708
```

```
## also can conduct a t test to assess the matches
with(nearest_matched, t.test(Percent_Attendance ~self_contained_option))
```

```
##
##  Welch Two Sample t-test
##
```

```
## data:  Percent_Attendance by self_contained_option
## t = 1.9783, df = 351.65, p-value = 0.04868
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.003391158 1.161580593
## sample estimates:
## mean in group 0 mean in group 1
##        92.68192        92.09944
```

```
### ## estimating treatment effects

model <- lm(Percent_Attendance ~ self_contained_binary, data = nearest_matched)
summary(model)
```

```
##
## Call:
## lm(formula = Percent_Attendance ~ self_contained_binary, data = nearest_matched)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.7994 -1.5994  0.4006  1.9799  5.7181
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              92.0994     0.2082 442.352   <2e-16 ***
## self_contained_binary     0.5825     0.2944   1.978   0.0487 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.77 on 352 degrees of freedom
## Multiple R-squared:  0.011,  Adjusted R-squared:  0.008186
## F-statistic: 3.913 on 1 and 352 DF,  p-value: 0.04868
```

IPTW

IPTW

```
## IPTW
#add the propensity scores
working_data$ps <- school_nearest$distance
#estimate the effect of SC option using IPTW
#create IPTW weights - 0 is no SC, which is the treatment
working_data$iptw <- ifelse(working_data$new_outcome_labelled == 'no SC', 1/(working_data$ps),
                            1/(1-working_data$ps))
#stabilized weights
working_data$stable.iptw <- ifelse(working_data$new_outcome_labelled == 'no SC',
                                   (mean(working_data$ps[working_data$new_outcome_labelled == 'no SC'])/
                                   (mean(1-working_data$ps[working_data$new_outcome_labelled == 'SC'])/(
working_data_nomiss<- working_data %>%
  select(Percent_Attendance, self_contained_binary, X..Poverty, X..Male, X..Black, Economic.Need.Index,
         self_contained_binary)
#weighted data - create a weighted version of the data
working_data_weighted <- svydesign(ids = ~1, data = working_data_nomiss, weights = working_data_nomiss$
#check the balance
```

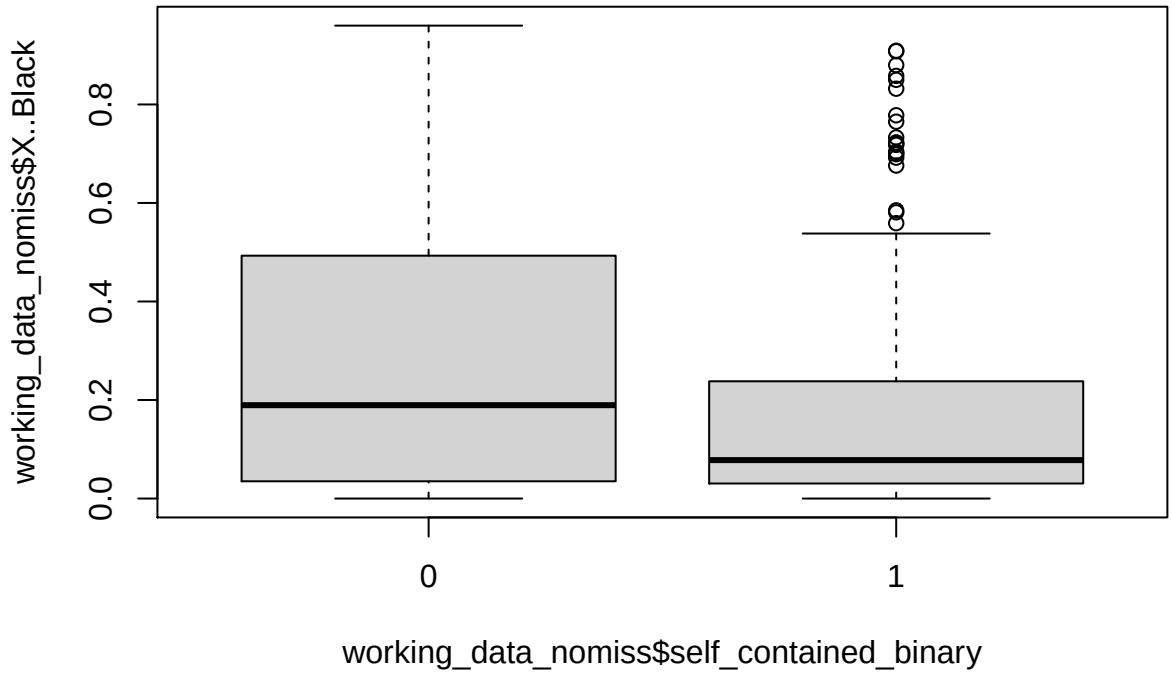
```r
SC_iptw_table <- svyCreateTableOne(vars = school_covariates, strata = "self_contained_binary", data = w
                                   test = F)
SC_iptw_table
```

```
##                                 Stratified by self_contained_binary
##                                  0             1
##   n                              728.26        757.55
##   X..Poverty (mean (SD))           0.77 (0.21)   0.77 (0.23)
##   X..Black (mean (SD))             0.26 (0.27)   0.24 (0.26)
##   X..Male (mean (SD))              0.51 (0.03)   0.51 (0.03)
##   Economic.Need.Index (mean (SD))  0.73 (0.22)   0.73 (0.24)
```

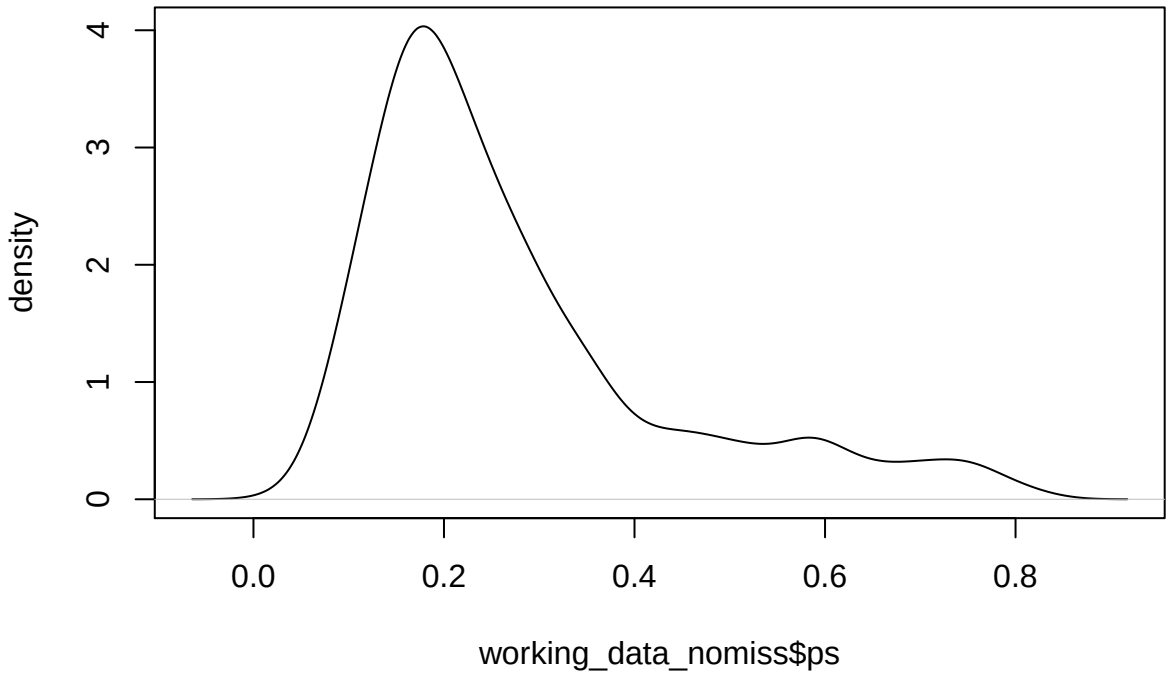
```r
print(SC_iptw_table, smd=T)
```

```
##                                 Stratified by self_contained_binary
##                                  0             1             SMD
##   n                              728.26        757.55
##   X..Poverty (mean (SD))           0.77 (0.21)   0.77 (0.23)  0.001
##   X..Black (mean (SD))             0.26 (0.27)   0.24 (0.26)  0.041
##   X..Male (mean (SD))              0.51 (0.03)   0.51 (0.03)  0.047
##   Economic.Need.Index (mean (SD))  0.73 (0.22)   0.73 (0.24)  0.006
```

```r
boxplot(working_data_nomiss$X..Black ~ working_data_nomiss$self_contained_binary)
```




```r
### check the covariate distribution for all the weights
for(i in 1:nrow(working_data_nomiss)){
  working_data_nomiss$X..Black[i] <- ifelse(working_data_nomiss$self_contained_binary == 1,
                                    (working_data_nomiss$X..Black[i]*(mean(working_data_nomiss$ps[worki
                                    (mean(1-(working_data_nomiss$ps[working_data_nomiss$self_contained_
}
# plot the weights
ipwplot(working_data_nomiss$ps, logscale = F,
        main = "propensity scores")
```




```
#lastly, test if distributions of covaraites are similar/diffeernt before or after weighting
ks.test(working_data_nomiss$X..Black[working_data_nomiss$self_contained_binary == 1],
        working_data_nomiss$X..Black[working_data_nomiss$self_contained_binary == 0])
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  working_data_nomiss$X..Black[working_data_nomiss$self_contained_binary == 1] and working_data_
## D = 0.2197, p-value = 1.363e-06
## alternative hypothesis: two-sided
```

```
#estimate the ate
mod_out_iptw <- lm(Percent_Attendance ~ self_contained_binary, weights = working_data_nomiss$iptw,
                   data = working_data_nomiss)
summary(mod_out_iptw)
```

```
##
## Call:
## lm(formula = Percent_Attendance ~ self_contained_binary, data = working_data_nomiss,
##     weights = working_data_nomiss$iptw)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -17.350  -2.229   0.411   2.932  13.617
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            91.6400     0.1521 602.579  < 2e-16 ***
## self_contained_binary   0.5950     0.2130   2.794  0.00534 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.104 on 735 degrees of freedom
## Multiple R-squared:  0.01051,    Adjusted R-squared:  0.009162
## F-statistic: 7.805 on 1 and 735 DF,  p-value: 0.005345
```

Subclassification

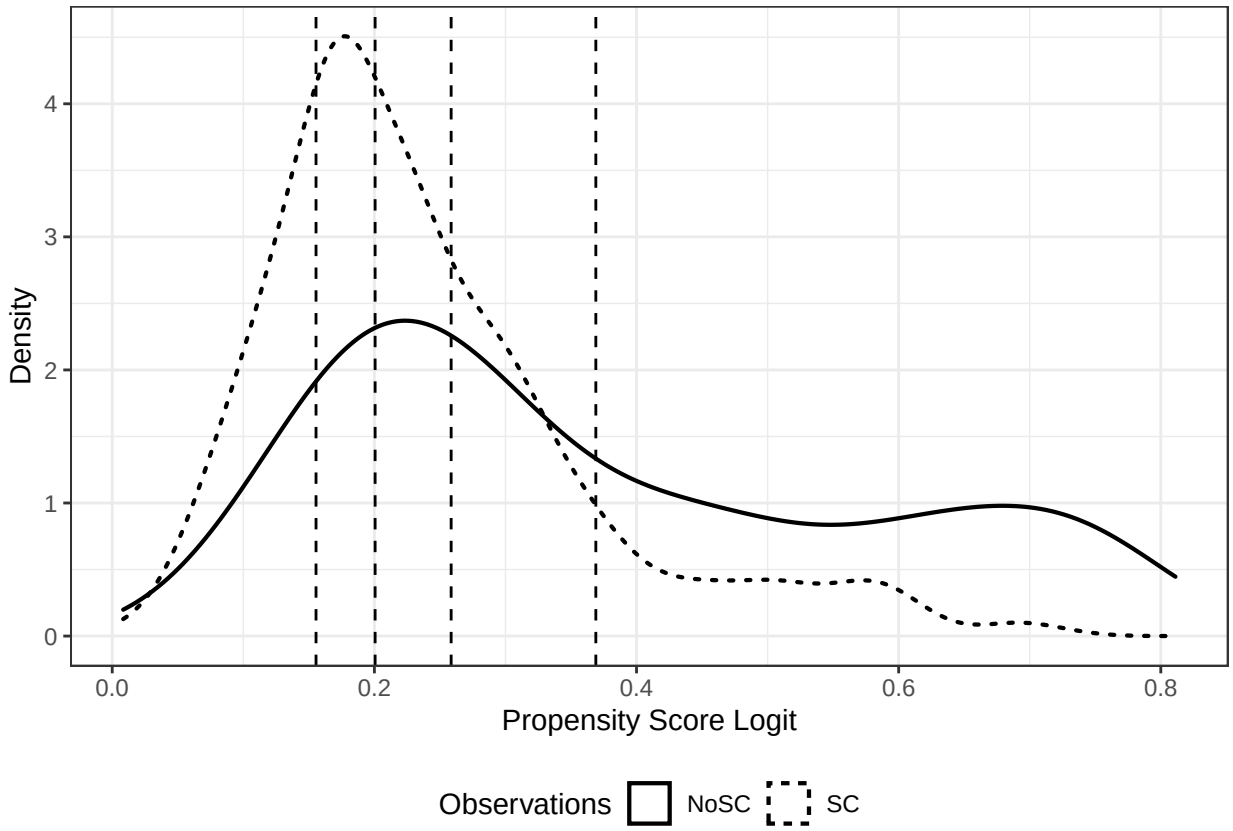
Subclassification

```
mod2 <- matchit(formula = self_contained_binary ~ Economic.Need.Index +
                          X..Black + X..Male + X..Poverty, data = working_data_nomiss,
                        method = "subclass", subclass = 5)
wd_nomiss2 <- data.frame(cbind(working_data_nomiss, match.data(mod2)[,c("distance", "subclass")]))
head(wd_nomiss2)
```

```
##   Percent_Attendance self_contained_binary X..Poverty X..Male  X..Black
## 1               92.3                     1      0.847   0.479 0.3262787
## 2               91.5                     0      0.770   0.556 0.4802217
## 3               91.9                     0      0.736   0.509 0.1151658
## 4               88.9                     1      0.979   0.550 1.2084091
## 5               92.2                     1      0.818   0.507 0.2941218
## 6               87.7                     1      0.922   0.588 0.9755674
##   Economic.Need.Index         ps      iptw stable.iptw   distance subclass
## 1               0.890 0.31457824  3.178859   1.1907983 0.31720769        3
## 2               0.679 0.14899045  1.175075   0.8957056 0.15597982        1
## 3               0.800 0.33502764  1.503822   1.1462943 0.33304117        3
## 4               0.937 0.09857800 10.144252   3.8000288 0.09440881        1
## 5               0.762 0.23179874  4.314087   1.6160538 0.23491121        2
## 6               0.882 0.07986802 12.520656   4.6902280 0.07910688        1
```

```
## so, all students in subclass 3 have similar propensity scores, etc.

dat <- wd_nomiss2[,c("distance", "self_contained_binary", "subclass")]
dat$Observations <- rep("NoSC", length(wd_nomiss2$self_contained_binary))
dat$Observations[dat$self_contained_binary == 0] <- "SC"
dat$ymax <- 1
quant <- quantile(wd_nomiss2$distance, probs = seq(0,1,1/5))
q <- data.frame(id = names(quant), values = unname(quant), stringsAsFactors = FALSE)
pp <- ggplot(data = dat, aes(x = distance, group = Observations))
pp + geom_density(aes(x = distance, linetype = Observations), size = 0.75, data = dat)+
  xlab("Propensity Score Logit") +
  ylab("Density") +
  geom_vline(xintercept = quant[(2:5)], linetype = "dashed") +
  theme_bw() +
  theme(legend.position = "bottom")
```




```
##estimate the ATE

mod_out_sub <- lm(Percent_Attendance ~ self_contained_binary +factor(subclass) + factor(subclass) *self_
                  data = wd_nomiss2)
summary(mod_out_sub)
```

```
##
## Call:
## lm(formula = Percent_Attendance ~ self_contained_binary + factor(subclass) +
##     factor(subclass) * self_contained_binary - 1, data = wd_nomiss2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3658  -1.7000   0.2889   1.8634   8.4342
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## self_contained_binary                     0.8708     0.4577   1.902   0.0575
## factor(subclass)1                        90.4658     0.1787 506.224   <2e-16
## factor(subclass)2                        91.3568     0.2414 378.518   <2e-16
## factor(subclass)3                        92.4167     0.2463 375.173   <2e-16
## factor(subclass)4                        93.3111     0.3672 254.109   <2e-16
## factor(subclass)5                        93.0286     1.0199  91.213   <2e-16
## self_contained_binary:factor(subclass)2  -0.2101     0.6707  -0.313   0.7542
## self_contained_binary:factor(subclass)3  -0.5143     0.6692  -0.769   0.4424
## self_contained_binary:factor(subclass)4  -0.2644     0.7255  -0.364   0.7156
## self_contained_binary:factor(subclass)5   1.1006     1.1947   0.921   0.3572
##
```

```
## self_contained_binary                    .
## factor(subclass)1                        ***
## factor(subclass)2                        ***
## factor(subclass)3                        ***
## factor(subclass)4                        ***
## factor(subclass)5                        ***
## self_contained_binary:factor(subclass)2
## self_contained_binary:factor(subclass)3
## self_contained_binary:factor(subclass)4
## self_contained_binary:factor(subclass)5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.698 on 727 degrees of freedom
## Multiple R-squared:  0.9991, Adjusted R-squared:  0.9991
## F-statistic: 8.544e+04 on 10 and 727 DF,  p-value: < 2.2e-16
```