Exercise 4

This fourth example explores the use of matching estimators, in particular, propensity score matching.

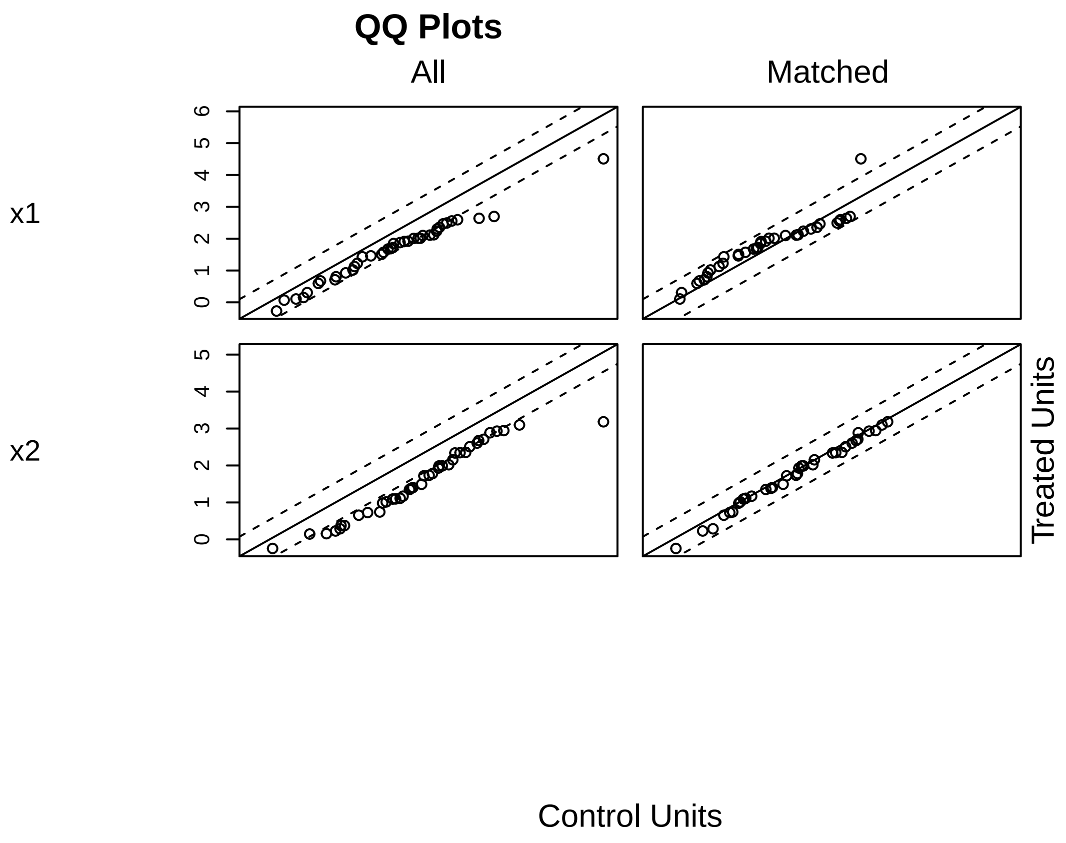
Here is a worked-out, and commented example of using MatchIt to perform PS Matching. Because the resulting outcome model is so simple, I am using a t-test. We could also use emmeans (in conjuction with a parametric model), but t-test works here.
I am also demonstrating a matching model checks.

The data is simulated, and if you like you can ignore the simulation code. It is provided here for the sole reasons that you can reproduce the example quickly without having to download data files.

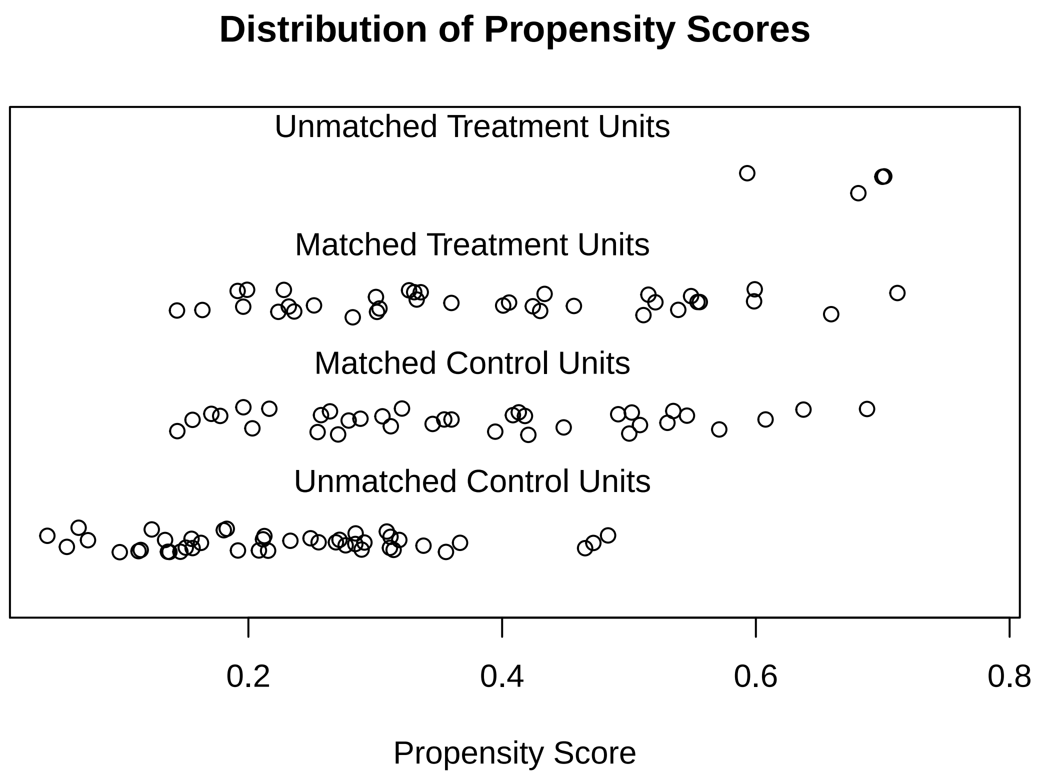
```
####simulation code##########################
set.seed(123)
n <- 120
u <- rnorm(n,0,1)
x1 <- .9*u + rnorm(n,2,.6)
x2 <- .9*u + rnorm(n,2,.6)
p <- (1/(1+exp(-.5+.1*x1+.5*x2)))
t <- rbinom(n,1,p)
y <- .5*t - .9*x1 - .9*x2 + .3*I(x1^2) - .3*I(x2^2) + .4*t*x1 - .3*t*x2 + .3*t*I(x1^2) + rnorm(n,0,.2)
t <- factor(t)
dfex4 <- data.frame(x1,x2,t,y)
dfex4$t <- factor(dfex4$t)

############################################

####analysis code###########################
library(MatchIt)
#here I am recoding the factor as numeric, because MatchIt seems to sometimes have issues with factors
dfex4$tn <- as.numeric(dfex4$t) -1

#this is the definition of the ps model
#first, we define which covariates predict the treatment using a simple R formula
#with method, we define the type of matching, e.g., nearest neighbor
#we define other parameters, like caliper, and whether units to discard
#with distance we define a logistic regression model to estimate the PS
m1 <- matchit(tn~x1+x2, method="nearest", distance="logit", caliper = .2, discard="none", data=dfex4)

#Here we have summaries and plots
#the summary provides sample sizes, and standardized mean difference before and after matching
summary(m1,standardize = TRUE,interactions = TRUE)
```

```
##
## Call:
## matchit(formula = tn ~ x1 + x2, data = dfex4, method = "nearest",
##     distance = "logit", discard = "none", caliper = 0.2)
##
## Summary of balance for all data:
##                     Means Treated Means Control SD Control Std. Mean Diff.
## distance                   0.4119        0.2940     0.1469          0.6964
## x1                         1.6312        2.1817     1.0763         -0.5924
## x2                         1.5859        2.3056     0.9630         -0.7573
## distancexdistance          0.1977        0.1078     0.1019          0.6079
## distancexx1              0.5694        0.5309     0.2430          0.1139
## distancexx2              0.4990        0.5435     0.1319         -0.2229
## x1xx1                    3.5026        5.9035     5.3585         -0.6825
## x1xx2                    3.0495        5.6377     4.4208         -0.9436
## x2xx2                    3.3957        6.2316     4.7829         -0.9219
##                   eCDF Med eCDF Mean eCDF Max
## distance            0.2125    0.1994   0.3250
## x1                  0.1312    0.1618   0.3125
## x2                  0.2000    0.1931   0.3000
## distancexdistance   0.2125    0.1994   0.3250
## distancexx1         0.0375    0.0436   0.1250
## distancexx2         0.0625    0.0669   0.1500
## x1xx1               0.1312    0.1608   0.3125
## x1xx2               0.2000    0.1866   0.2875
## x2xx2               0.2000    0.1930   0.3000
##
##
## Summary of balance for matched data:
##                   Means Treated Means Control SD Control Std. Mean Diff.
## distance                 0.3834        0.3749     0.1477          0.0500
## x1                       1.7582        1.6340     0.9207          0.1338
## x2                       1.7329        1.8194     0.8587         -0.0910
## distancexdistance        0.1697        0.1618     0.1183          0.0535
## distancexx1              0.6005        0.5359     0.2800          0.1912
## distancexx2              0.5353        0.5617     0.1409         -0.1325
## x1xx1                    3.7772        3.4940     3.2674          0.0805
## x1xx2                    3.3694        3.2880     2.7611          0.0297
## x2xx2                    3.7640        4.0269     3.1658         -0.0855
##                   eCDF Med eCDF Mean eCDF Max
## distance            0.0278    0.0316   0.1111
## x1                  0.0833    0.0818   0.1944
## x2                  0.0278    0.0293   0.0833
## distancexdistance   0.0278    0.0316   0.1111
## distancexx1         0.0556    0.0556   0.1389
## distancexx2         0.0833    0.0880   0.1667
## x1xx1               0.0833    0.0818   0.1944
## x1xx2               0.0556    0.0532   0.1389
## x2xx2               0.0278    0.0293   0.0833
##
## Percent Balance Improvement:
##                   Std. Mean Diff. eCDF Med eCDF Mean eCDF Max
## distance                  92.8173  86.9281   84.1325  65.8120
## x1                        77.4212  36.5079   49.4407  37.7778
## x2                        87.9852  86.1111   84.8176  72.2222
## distancexdistance         91.1991  86.9281   84.1325  65.8120
## distancexx1              -67.8070 -48.1481  -27.2872 -11.1111
## distancexx2               40.5430 -33.3333  -31.5334 -11.1111
## x1xx1                     88.2056  36.5079   49.1460  37.7778
## x1xx2                     96.8562  72.2222   71.4622  51.6908
## x2xx2                     90.7274  86.1111   84.8094  72.2222
##
## Sample sizes:
##           Control Treated
## All            80      40
## Matched        36       36
## Unmatched      44        4
## Discarded       0        0
```

```
#these are diagnostic plots, that check whether the distributions of the covariates are equal
plot(m1,type="QQ")
```

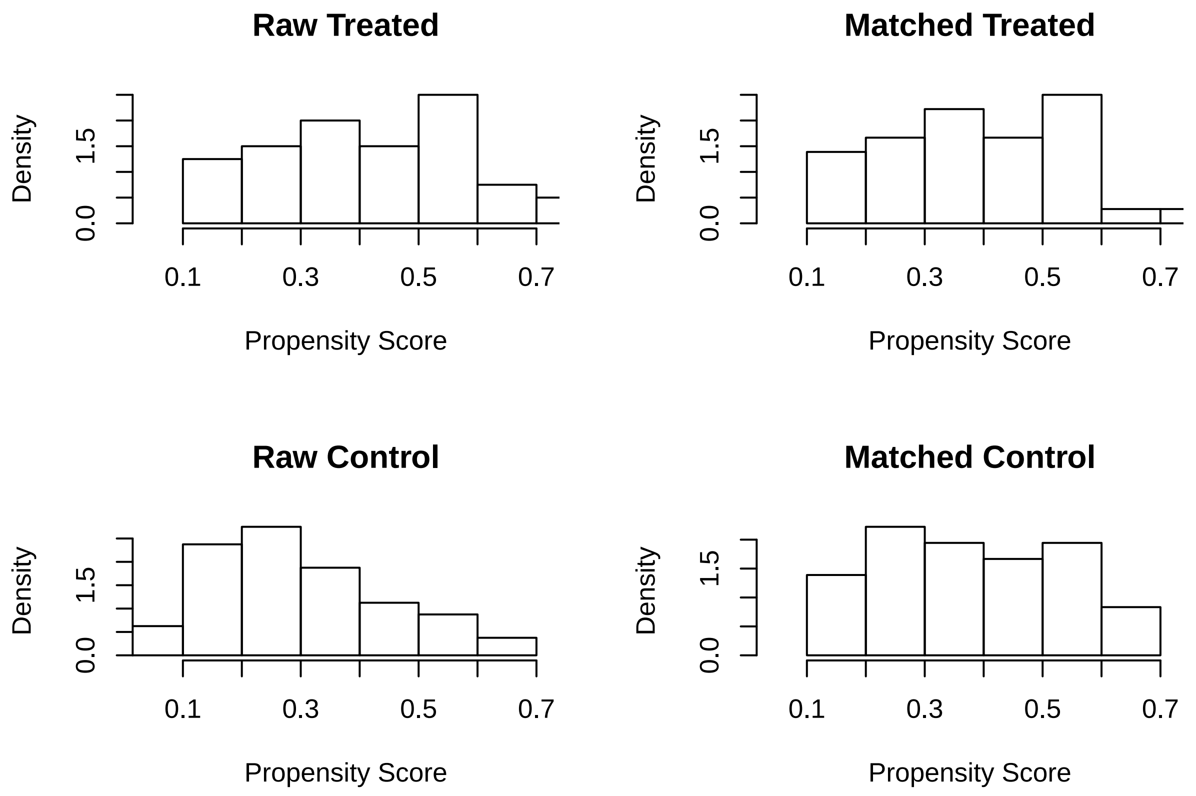


```
plot(m1,type="jitter")
```

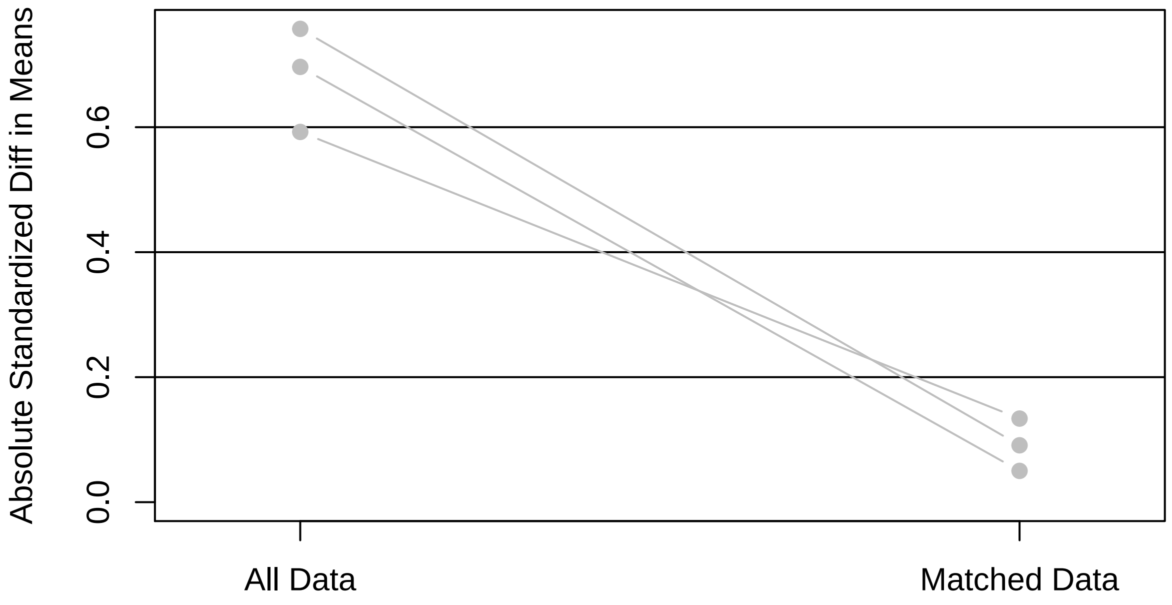


```
## [1] "To identify the units, use first mouse button; to stop, use second."
## integer(0)
```

```
plot(m1,type="hist")
```



```r
plot(summary(m1,standardize = TRUE))
```



```
## [1] "To identify the variables, use first mouse button; to stop, use second."
```

```
## integer(0)
```

```
#the data extraction returns by default a long datafile
md1 <-match.data(m1)

#if we want a matched datafile this ratehr complicated self-written function will do it
get_treated_df = function(m, out_matched_cases_only = F) {
  is.na(m$match.matrix)[m$match.matrix=='-1'] = T
  df = m$model$data[rownames(m$match.matrix),]
  df$matched_id = rownames(m$match.matrix)
  df$matched_cases_in_control = apply(m$match.matrix,1, function(row) sum(!is.na(row)))
  if(out_matched_cases_only) {
    return(df[apply(m$match.matrix, 1, function(row) any(!is.na(row))),])
  } else {
    return(df)
  }
}
get_controlled_df = function(m, out_matched_cases_only=F) {
  is.na(m$match.matrix)[m$match.matrix=='-1'] = T
  apply(m$match.matrix, 2, function(rowname) {
    df = m$mode$data[rowname,]
    df$matched_id = rownames(m$match.matrix)
    df$matched_cases_in_control = apply(m$match.matrix,1, function(row) sum(!is.na(row)))

    if(out_matched_cases_only) {
```

```r
      return(df[apply(df, 1, function(row) all(!is.na(row))),])
    } else {
      return(df)
    }
  })
}

data_matched = cbind(get_treated_df(m1), get_controlled_df(m1)[[1]])

#Finally, we do a simple t-test to estiamte our treatment effect
t.test(data_matched[,4],data_matched[,11],paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  data_matched[, 4] and data_matched[, 11]
## t = 6.322, df = 35, p-value = 2.914e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.300882 2.531548
## sample estimates:
## mean of the differences
##                1.916215
```

We see summary statistics before and after matching. Of special interest is the column on standardized mean differences. The plots (QQ,histogram,lineplot) all try to answer the question of whether balance was achieved.

By default, MatchIt returns a long dataframe, but we have a function that returns a wide dataframe (for dependent samples analyses). The last step is a simple t-test (we could have used emmeans, but it's not really necessary).

Exercise:

1.) Download the file dfex3a again from github (https://raw.githubusercontent.com/felixthoemmes/IPN_workshop/master/dfex3a.csv). You can download this file directly into R (no need to navigate to github in a browser, using the following code snippet:

```r
library(readr)
dfex3a <- read_csv("https://raw.githubusercontent.com/felixthoemmes/IPN_workshop/master/dfex3a.csv")
```

The file contains a treatment t, an outcome y, and covariates x1-x4. We assume that these variables are those that fulfill the back-door criterion. Obtain an *unadjusted* estimate for the effect of t on y.

2.) First, run a simple model (all linear terms) and carefully check balance. If you are unhappy with the balance, respecify the model, and check balance until it is achieved. Produce all plots that we discussed.

3.) Extract the wide dataset, and use a dependent samples t-test to estiamte the effect.