Matching for Adjustment and Causal Inference
Class 2: Distance Matrices,Propensity Scores, Calipers, Exact Matching, Combining Distance Matrices

Jake Bowers

August 01, 2023

1. Overview and Review
2. How to assess the randomization process in an experiment (to teach us how to assess research designs in observational studies).
3. Assessing comparisons in observational studies
4. Matching on Many Covariates: Using the Mahalnobis Distance to Scale Euclidean Distance
5. Matching on Many Covariates: Using Propensity Scores
6. Matching Tricks of the Trade: Calipers, Exact Matching
7. The separation problem in Logistic Regression

Last Time I

1. Yet more evidence that adjustment for background covariates using the linear model ("controlling for") is difficult: difficult to explain, difficult to justify and assess, etc.. Too many specifications to choose from, too difficult to assess the influence of functional form assumptions (let alone extrapolation and interpolation) with many covariates. Although we will use the linear model for estimation we will avoid it for adjustment.
2. Stratification is an old and simple idea: hold constant by holding constant directly — breaking continuous variables into pieces, or just estimating effects within groups. This is easy to explain. The adjustment is transparent and occurs **without reference to outcomes**.

Last Time II

3. Block-randomized experiments are well known and methods for estimating overall ATE from block-randomized studies are also well established: so stratification based approaches need not leave us with many imprecise treatment effects, for example. So, we can use the general techniques of combining block-specific or stratum-specific effects by weighting from that literature. This leaves us with two kinds of weights (a) block-size weights and (b) precision weights (which add the ratio of treated to control to its measure of information contributed to the overall estimate from a given block).
4. We can assess the success of a stratification by comparing it directly to a randomized experiment — leading to a hypothesis test or a balance test (based on randomization as the standard of comparison). We can use `balanceTest` from `RItools` for this or, in the `coin` package `independenceTest` does the same thing, or we can do it directly if we have small numbers of observations.
5. We can assess the success of a stratification just by inspecting the blocks from the perspective of substantive and disciplinary knowledge.

Today: Propensity distances, exact matching, calipers, combining distance matrices

1. Optimal full matching (optimal following Paul R Rosenbaum (2010), Chap 8 discussion and cites therein) creates stratifications that minimize differences between treated and control units — this side-steps questions about cut-points or about numbers of groups. The number of sets is optimal in so far as it minimizes overall within set differences.
2. To create a stratified research design (something like a block-randomized experiment), we first need a distance matrix — something that records the similarities/differences between each treated and each control unit. Last time we used (1) distances on a single variable and (2) we used a Mahalanobis distance to represent multivariate distance in a space of more than one covariate.
3. One way to combine covariates is the Euclidean distance, another (scaled version) is the Mahalanobis distance, another way to combine covariates is the propensity score (which gives different weight to different covariates).
4. When we have a categorical or binary covariate that is important sometimes we want to exactly stratify on it — leading to exact matching.
5. Sometimes we want to restrict the possible matches — and to allow the matching algorithm to exclude certain units from the research design entirely. This is the role of calipers.
6. We can combine distance matrices in order to make a strong argument about our research design.

But first let's talk about how to assess a **randomized** research design so that we can apply these ideas to an **stratified observational study**.

1. Overview and Review
2. How to assess the randomization process in an experiment (to teach us how to assess research designs in observational studies).
3. Assessing comparisons in observational studies
4. Matching on Many Covariates: Using the Mahalnobis Distance to Scale Euclidean Distance
5. Matching on Many Covariates: Using Propensity Scores
6. Matching Tricks of the Trade: Calipers, Exact Matching
7. The separation problem in Logistic Regression

The Neyman-Rubin Model for (simple) experiments

This is what randomization ensures:

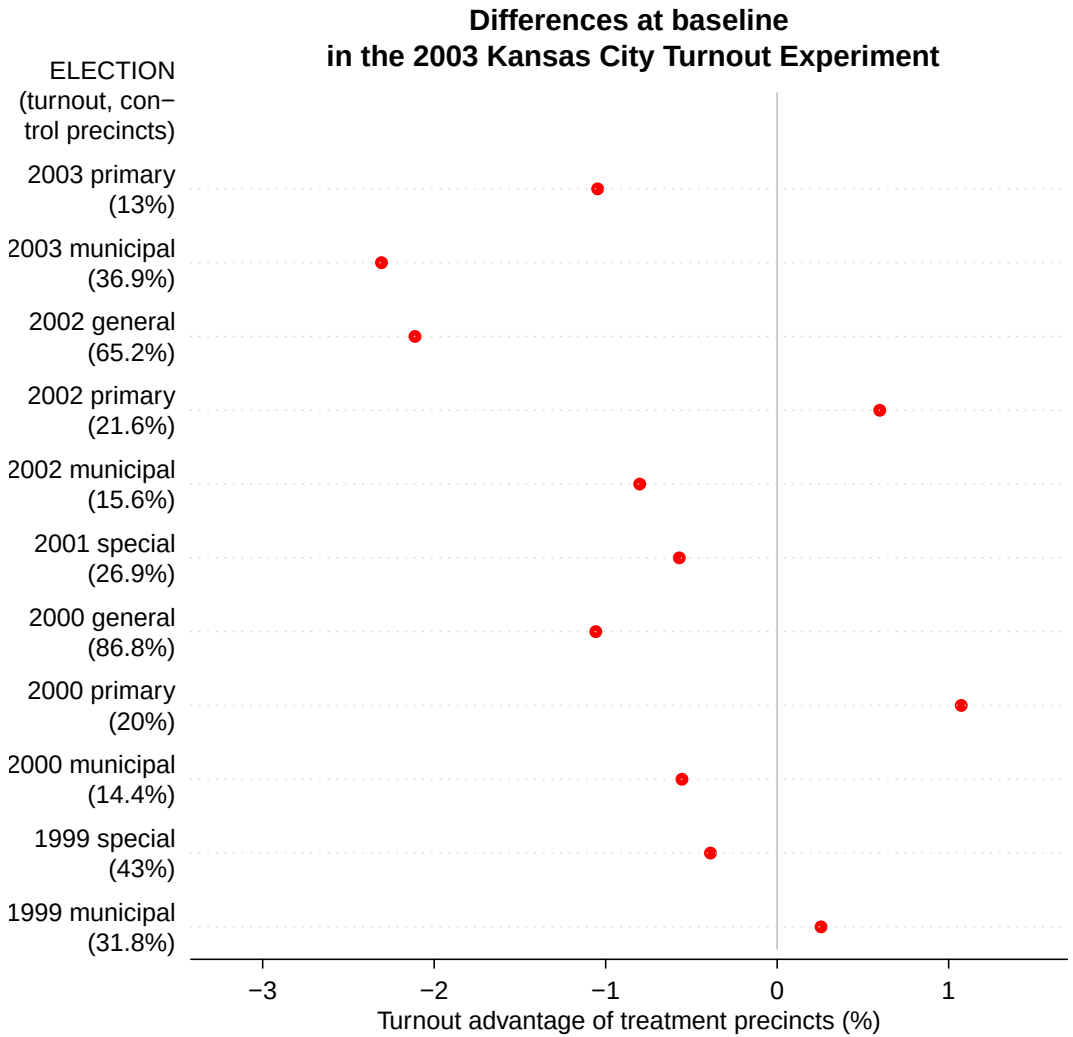
$$(y_t, y_c, X) \perp Z$$

i.e., each of the distributions of X, y_c and y_t is balanced between treatment and control groups (in expectation; given variability from randomization).

- In controlled experiments, random assignment justifies this argument.
- In natural experiments, justified otherwise, this is an article of faith.
- In an experiment, the $x$es aren't necessary for inference (although they can be used, carefully, to increase precision in both the design and analysis phases of a project).
- **However, the part with the $x$es has testable consequences** if you worried about the success of the randomization — say, the path between the random numbers on your computer and the application in the field.

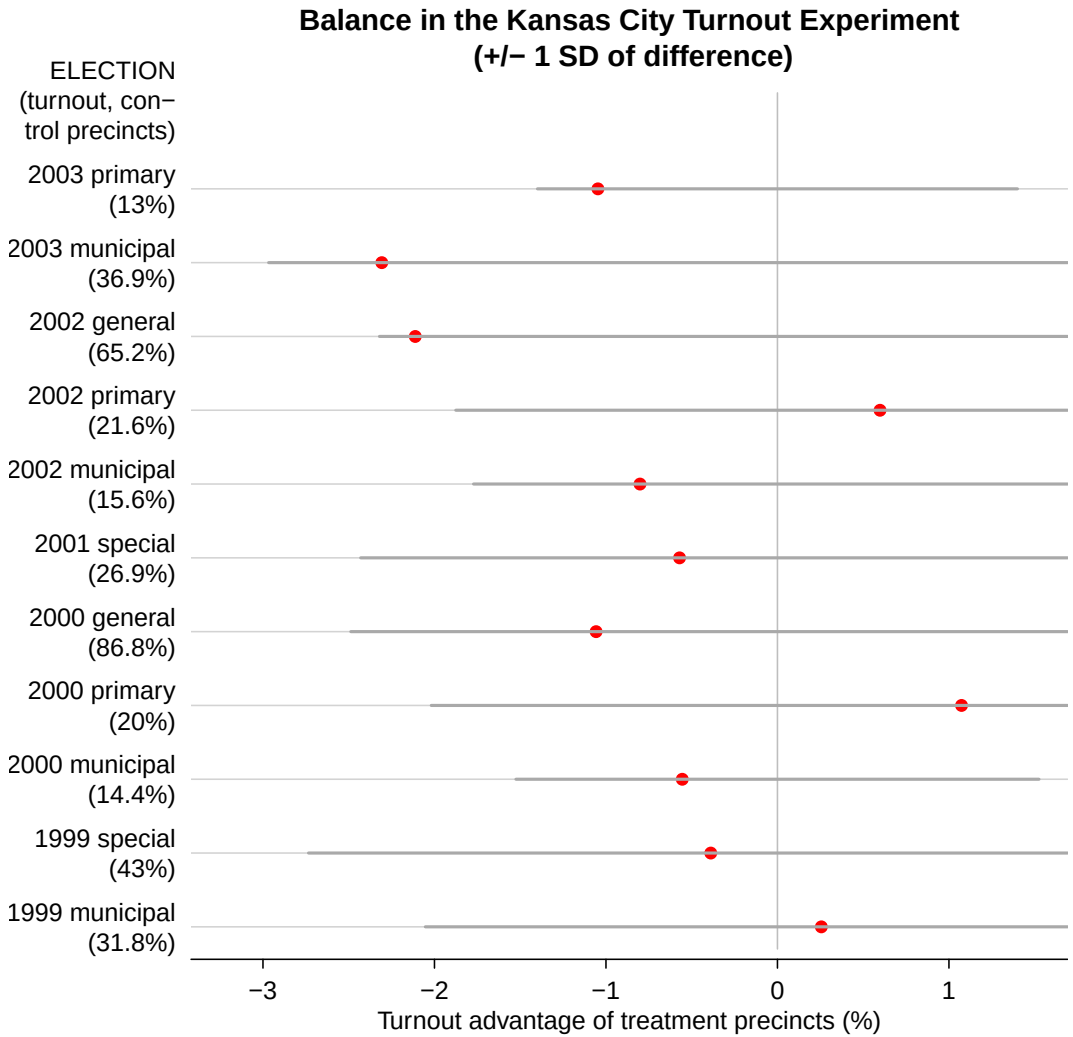
Covariate balance in experiments: What does it look like?

- Arceneaux, 2005
- Kansas City, November 2003
- Completely randomized design: 14 precincts $\rightarrow$ Tx; 14 $\rightarrow$ Control.
- Substantively large baseline differences (red dots)
- Differences not large compared to other possible assignments from same design; compared to other possible experiments with the same design.
- $\mathbf{Pr}(\chi^2 > x) = .91$ (Hansen and Bowers, 2008). (grey lines)



Covariate balance in experiments: What does it look like?

- Arceneaux, 2005
- Kansas City, November 2003
- Completely randomized design: 14 precincts $\rightarrow$ Tx; 14 $\rightarrow$ Control.
- Substantively large baseline differences (red dots)
- Differences not large compared to other possible assignments from same design; compared to other possible experiments with the same design.
- $\mathbf{Pr}(\chi^2 > x) = .91$ (Hansen and Bowers, 2008). (grey lines)



How did we do this?

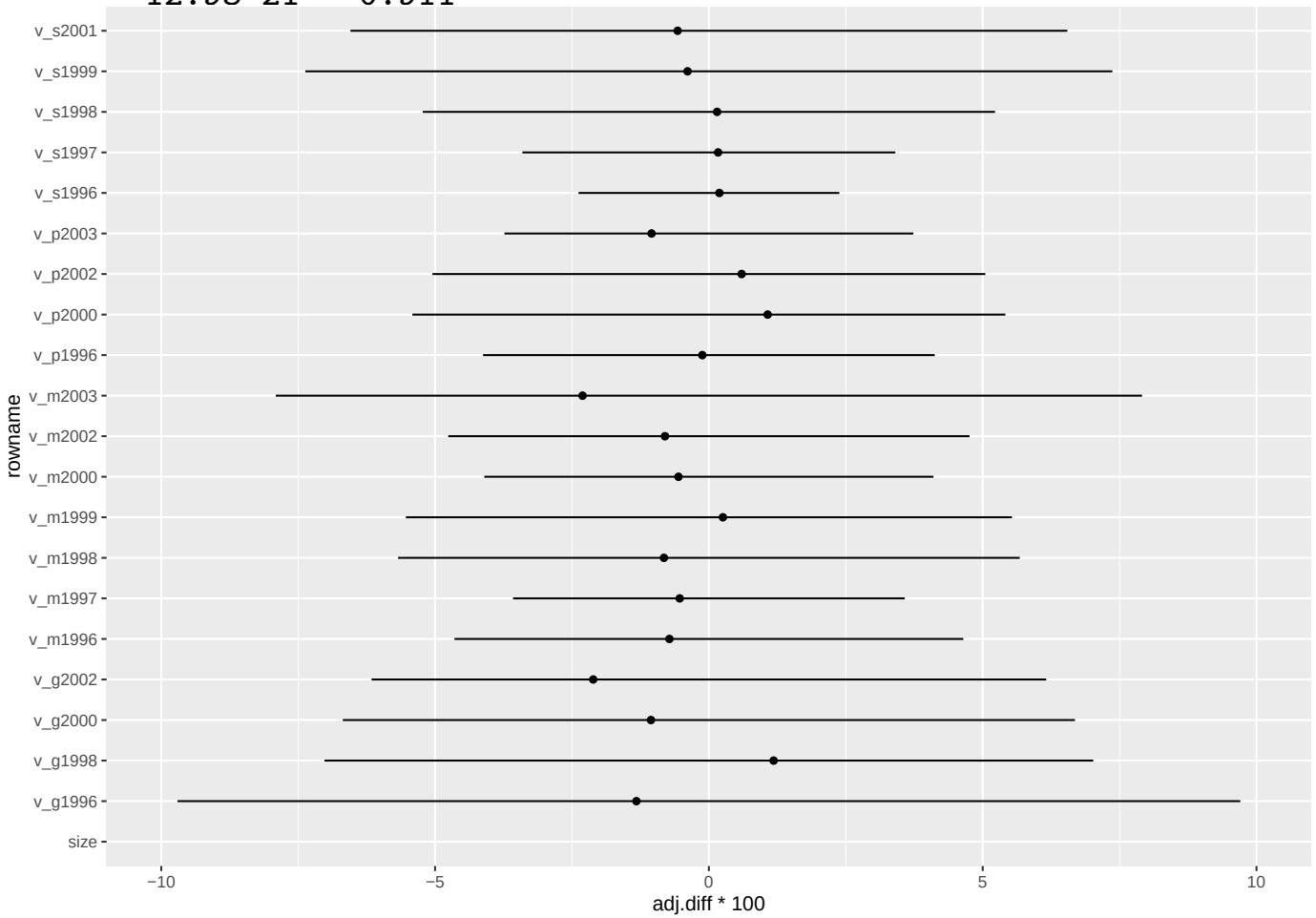
```
acorn <- read.csv(here("data", "acorn03.csv"), row.names = 1)
xb1 <- balanceTest(
  z ~ v_p2003 + v_m2003 + v_g2002 + v_p2002 + v_m2002 + v_s2001 +
    v_g2000 + v_p2000 + v_m2000 + v_s1999 + v_m1999 + v_g1998 +
    v_m1998 + v_s1998 + v_m1997 + v_s1997 + v_g1996 + v_p1996 +
    v_m1996 + v_s1996 + size,
  p.adjust.method = "none",
  data = acorn
)

xb1$results
```

```
, , strata = --

         stat
vars        Control Treatment std.diff  adj.diff pooled.sd       z      p
  v_p2003   0.12996   0.11949 -0.28041 -0.010469   0.03734 -0.74815 0.4544
  v_m2003   0.36868   0.34561 -0.29171 -0.023072   0.07909 -0.77763 0.4368
  v_g2002   0.65195   0.63083 -0.34283 -0.021119   0.06160 -0.91003 0.3628
  v_p2002   0.21589   0.22187  0.11848  0.005980   0.05048  0.31883 0.7499
  v_m2002   0.15567   0.14766 -0.16841 -0.008016   0.04760 -0.45233 0.6510
  v_s2001   0.26898   0.26327 -0.08722 -0.005709   0.06546 -0.23492 0.8143
  v_g2000   0.86812   0.85755 -0.15821 -0.010576   0.06685 -0.42512 0.6707
  v_p2000   0.20033   0.21106  0.19827  0.010735   0.05414  0.53177 0.5949
  v_m2000   0.14404   0.13849 -0.13536 -0.005552   0.04101 -0.36405 0.7158
  v_s1999   0.42957   0.42569 -0.05275 -0.003887   0.07369 -0.14216 0.8870
  v_m1999   0.31756   0.32012  0.04624  0.002559   0.05534  0.12463 0.9008
  v_g1998   0.43718   0.44900  0.16844  0.011824   0.07019  0.45243 0.6510
  v_m1998   0.18136   0.17315 -0.14463 -0.008210   0.05677 -0.38886 0.6974
  v_s1998   0.24104   0.24254  0.02863  0.001496   0.05226  0.07719 0.9385
```

How did we do this?

```
xb1$overall
```

```
   chisquare df p.value
--     12.93 21   0.911
```



DeMystifying balanceTest

```
d_stat <- function(zz, mm, ss) {
  ## this is the d_statistic (harmonic mean weighted diff of means statistic)
  ## from Hansen and Bowers 2008 almost directly from balanceTest.Engine
  h.fn <- function(n, m) {
    (m * (n - m)) / n
  }
  myssn <- apply(mm, 2, function(x) {
    sum((zz - unsplit(tapply(zz, ss, mean), ss)) * x)
  })
  hs <- tapply(zz, ss, function(z) {
    h.fn(m = sum(z), n = length(z))
  })
  mywtsum <- sum(hs)
  myadjdiff <- myssn / mywtsum
  return(myadjdiff)
}
```

DeMystifying balanceTest

Recall our discussion of estimation "holding constant" within strata?

```
## This is another version that might be more clear in regards what is going on.
d_stat_v2 <- function(zz, mm, ss) {
  ## mm is a data.frame
  dat <- cbind(mm, z = zz, s = ss)
  datb <- dat %>%
    group_by(s) %>%
    summarize(
      across(.cols = all_of(names(mm)), function(x) {
        mean(x[z == 1]) - mean(x[z == 0])
      }),
      nb = n(),
      pib = mean(z),
      nbwt = nb / nrow(dat),
      hbwt0 = pib * (1 - pib) * nbwt
    )
  datb$hbwt <- datb$hbwt0 / sum(datb$hbwt0)
  # datb[,15:27]
  adjmns <- datb %>% summarize(across(.cols = all_of(names(mm)), function(x) {
    sum(x * hbwt)
  }))
  adjmnsmat <- as.matrix(adjmns)
  return(adjmnsmat)
}
```

DeMystifying balanceTest

```
acorncovs <- c("v_p2003", "v_m2003", "v_g2002", "v_p2002", "v_m2002", "v_s2001", "v_g2000", "v_p2
```

```
dstats1 <- d_stat(zz = acorn$z, mm = acorn[, acorncovs], ss = rep(1, nrow(acorn)))
dstats2 <- d_stat_v2(zz = acorn$z, mm = acorn[, acorncovs], ss = rep(1, nrow(acorn)))
```

```
dstats1[1:5]
```

```
  v_p2003   v_m2003   v_g2002   v_p2002   v_m2002
-0.010469 -0.023072 -0.021119  0.005980 -0.008016
```

```
dstats2[1:5]
```

```
[1] -0.010469 -0.023072 -0.021119  0.005980 -0.008016
```

```
xb1$results[, "adj.diff", ]
```

```
  v_p2003   v_m2003   v_g2002   v_p2002   v_m2002   v_s2001   v_g2000   v_p2000   v_m2000   v_s19
-0.010469 -0.023072 -0.021119  0.005980 -0.008016 -0.005709 -0.010576  0.010735 -0.005552 -
0.003887  0.002559  0.011824 -0.008210
  v_s1998   v_m1997   v_s1997   v_g1996   v_p1996   v_m1996   v_s1996      size
 0.001496 -0.005326  0.001684 -0.013232 -0.001199 -0.007205  0.001915 11.000000
```

```
stopifnot(all.equal(dstats1, dstats2[1, ]))
```

The reference distribution of the d^2 stat

For all vectors $z \in \Omega$ get `adj.diffs`. This is the distribution of the d statistic for one-by-one balance assessment. Next question is about the distribution of the d^2 statistic: does it follow a χ^2 distribution in this case?

Get the randomization-based p-values:

```
xb1ds <- xb1$results[, "adj.diff", ]
xb1ps <- xb1$results[, "p", ]
obs.d <- d_stat(acorn$z, acorn[, acorncovs], rep(1, nrow(acorn)))
dps <- matrix(NA, nrow = length(obs.d), ncol = 1)
for (i in 1:length(obs.d)) {
  dps[i, ] <- 2 * min(mean(d_dist[i, ] >= obs.d[i]), mean(d_dist[i, ] <= obs.d[i]))
}
## You can compare this to the results from balanceTest
round(cbind(randinfps = dps[, 1], xbps = xb1ps, obsdstats = obs.d, xbdstats = xb1ds), 3)
```

```
         randinfps  xbps obsdstats xbdstats
v_p2003      0.466 0.454    -0.010   -0.010
v_m2003      0.448 0.437    -0.023   -0.023
v_g2002      0.380 0.363    -0.021   -0.021
v_p2002      0.758 0.750     0.006    0.006
v_m2002      0.661 0.651    -0.008   -0.008
v_s2001      0.823 0.814    -0.006   -0.006
v_g2000      0.699 0.671    -0.011   -0.011
v_p2000      0.618 0.595     0.011    0.011
v_m2000      0.711 0.716    -0.006   -0.006
v_s1999      0.892 0.887    -0.004   -0.004
v_m1999      0.918 0.901     0.003    0.003
v_g1998      0.679 0.651     0.012    0.012
v_m1998      0.695 0.697    -0.008   -0.008
```

Reference distribution of the d^2 stat I

The d^2 statistic is a linear function of the probably correlated d-statistics: a linear combination of correlated variables is $d^T \Sigma_d^{-1} d$ where d is a vector of the stratum adjusted differences in means and Σ_d is the variance-covariance matrix of the distribution of the d statistics under the sharp null of no differences. With only one variable, this is basically a standardized d statistic (after taking sqrt, and forced to be positive).

```
d2_stat <- function(dstats, ddist = NULL, theinvcov = NULL) {
  ## d is the vector of d statistics
  ## ddist is the matrix of the null reference distributions of the d statistics
  if (is.null(theinvcov) & !is.null(ddist)) {
    as.numeric(t(dstats) %*% solve(cov(t(ddist))) %*% dstats)
  } else {
    as.numeric(t(dstats) %*% theinvcov %*% dstats)
  }
}
```

Reference distribution of the d^2 stat

The distribution of the d^2 statistic arises from the distribution of the d statistics — for each draw from the set of treatment assignments we can collapse the d-statistics into one d^2. And so we can calculate the p-value for the d^2.

```
## Here we have the inverse of the covariance/variance matrix of the d statistics
invCovDDist <- solve(cov(t(d_dist)))
obs.d2 <- d2_stat(obs.d, d_dist, invCovDDist)

d2_dist <- apply(d_dist, 2, function(thed) {
  d2_stat(thed, theinvcov = invCovDDist)
})
## The chi-squared reference distribution only uses a one-sided p-value going in the positive dire
d2p <- mean(d2_dist >= obs.d2)
cbind(obs.d2, d2p)
```

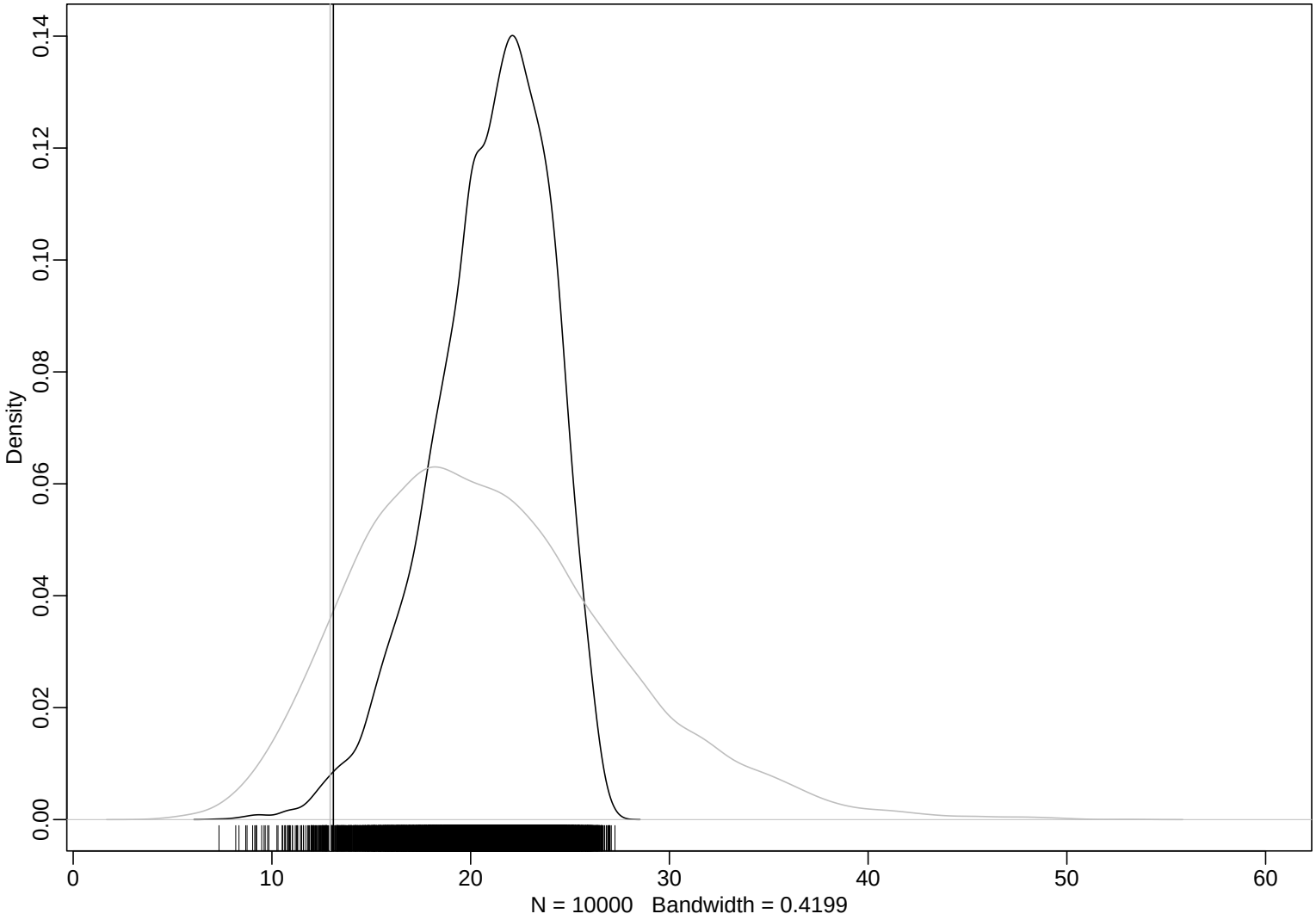
```
     obs.d2    d2p
[1,]  13.09 0.9882
```

```
xb1$overall
```

```
   chisquare df p.value
--     12.93 21   0.911
```

Why differences between balanceTest and d2?

I suspect that $N = 28$ is too small. `balanceTest` uses an asymptotic approximation to the randomization distribution. But the substantive differences are small.



Summary

- Randomization balances covariate distributions between treated and control groups.
- We can use randomization inference to check the randomization procedure (mostly useful if there is a long chain of communication between the random number generator and the field).
- **Randomization does not imply exact equivalence. Large differences in covariates easily arise in small experiments.**

1. Overview and Review
2. How to assess the randomization process in an experiment (to teach us how to assess research designs in observational studies).
3. Assessing comparisons in observational studies
4. Matching on Many Covariates: Using the Mahalnobis Distance to Scale Euclidean Distance
5. Matching on Many Covariates: Using Propensity Scores
6. Matching Tricks of the Trade: Calipers, Exact Matching
7. The separation problem in Logistic Regression

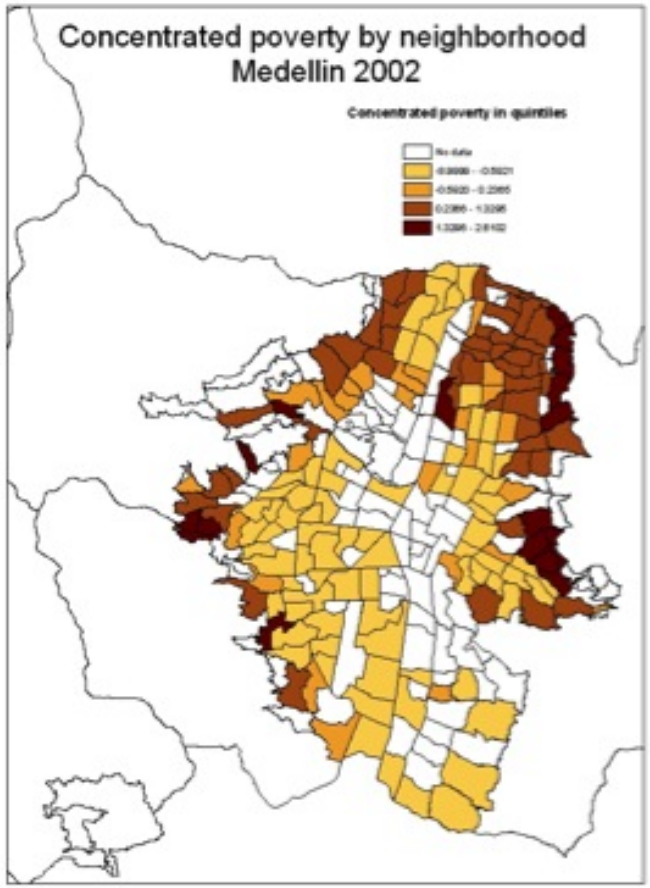
Assessing comparisons in observational studies

Introducing the Medellin Data

Cerdá et al. collected data on about roughly 45 neighborhoods in Medellin, Colombia. About 22 had access to the new Metrocable line and 23 did not.

Introducing the Medellin Data

Cerdá et al. collected data on about roughly 45 neighborhoods in Medellin, Colombia. About 22 had access to the new Metrocable line and 23 did not.



Introducing the Medellin Data: Variables Collected

```
## The Intervention
nhTrt         Intervention neighborhood (0=no Metrocable station, 1=Metrocable station)

## Some Covariates (there are others, see the paper itself)
nh03          Neighborhood id
nhGroup       Treatment (T) or Control (C)
nhTrt         Treatment (1) or Control (0)
nhHom         Mean homicide rate per 100,000 population in 2003
nhDistCenter Distance to city center (km)
nhLogHom      Log Homicide (i.e. log(nhHom))

## Outcomes (BE03,CE03,PV03,QP03,TP03 are baseline versions)
BE      Neighborhood amenities Score 2008
CE      Collective Efficacy Score 2008
PV      Perceived Violence Score 2008
QP      Trust in local agencies Score 2008
TP      Reliance on police Score 2008
hom     Homicide rate per 100,000 population Score 2008-2003 (in log odds)

HomCount2003 Number of homicides in 2003
Pop2003      Population in 2003
HomCount2008 Number of homicides in 2008
Pop2008      Population in 2008
```

Get rates from counts:

```
meddat <- mutate(meddat,
  HomRate03 = (HomCount2003 / Pop2003) * 1000,
  HomRate08 = (HomCount2008 / Pop2008) * 1000
)
```

What is the effect of the Metrocable on Homicides? I

One approach: Estimate the average treatment effect of Metrocable on Homicides after the stations were built.

```
## code here
themeans <- group_by(meddat, nhTrt) %>% summarise(ybar = mean(HomRate08))
diff(themeans$ybar)
```

```
[1] -0.2899
```

```
lmOne <- lm(HomRate08 ~ nhTrt, meddat)
coef(lmOne)["nhTrt"]
```

```
  nhTrt
-0.2899
```

```
library(estimatr)
difference_in_means(HomRate08 ~ nhTrt, meddat)
```

```
Design:  Standard
      Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper    DF
nhTrt  -0.2899     0.1508  -1.922  0.06137  -0.5942  0.01445 41.87
```

Another approach, test the null of no effects:

```
balanceTest(nhTrt ~ HomRate08, data = meddat)
```

What is the effect of the Metrocable on Homicides? II

```
          strata():        --
          stat      Treatment Control adj.diff std.diff    z
vars
HomRate08               0.40    0.69     -0.29    -0.6    -1.9  .
---Overall Test---
   chisquare df p.value
--       3.5  1   0.063
---
Signif. codes:  0 '***' 0.001 '** ' 0.01 '*  ' 0.05 '.  ' 0.1 '   ' 1
```

```
meddat$nhTrtF <- factor(meddat$nhTrt)
test2 <- oneway_test(HomRate08 ~ nhTrtF, data = meddat, distribution = asymptotic())
test3 <- oneway_test(HomRate08 ~ nhTrtF, data = meddat, distribution = approximate(nresample = 10
test4 <- wilcox_test(HomRate08 ~ nhTrtF, data = meddat, distribution = approximate(nresample = 10
pvalue(test2)
```

```
[1] 0.06317
```

```
pvalue(test3)
```

```
[1] 0.068
99 percent confidence interval:
 0.04909 0.09114
```

```
pvalue(test4)
```

What is the effect of the Metrocable on Homicides? III

```
[1] 0.022
99 percent confidence interval:
 0.01185 0.03694
```

Do we have any concerns about confounding?

Sometimes people ask about "bias from observed confounding" or "bias from selection on observables".

How would we interpret the following results where `nhAboveHS` is proportion with more than a high school education in the neighborhood in 2003 or so and `nhTrt` is 0=no station built, 1=station built? (Recall how we justified the use of `balanceTest` in terms of randomization above.)

```
xbMed1 <- balanceTest(nhTrt ~ nhAboveHS, data = meddat)
xbMed1$overall
```

```
   chisquare df p.value
--     4.598  1 0.03201
```

```
xbMed1$results
```

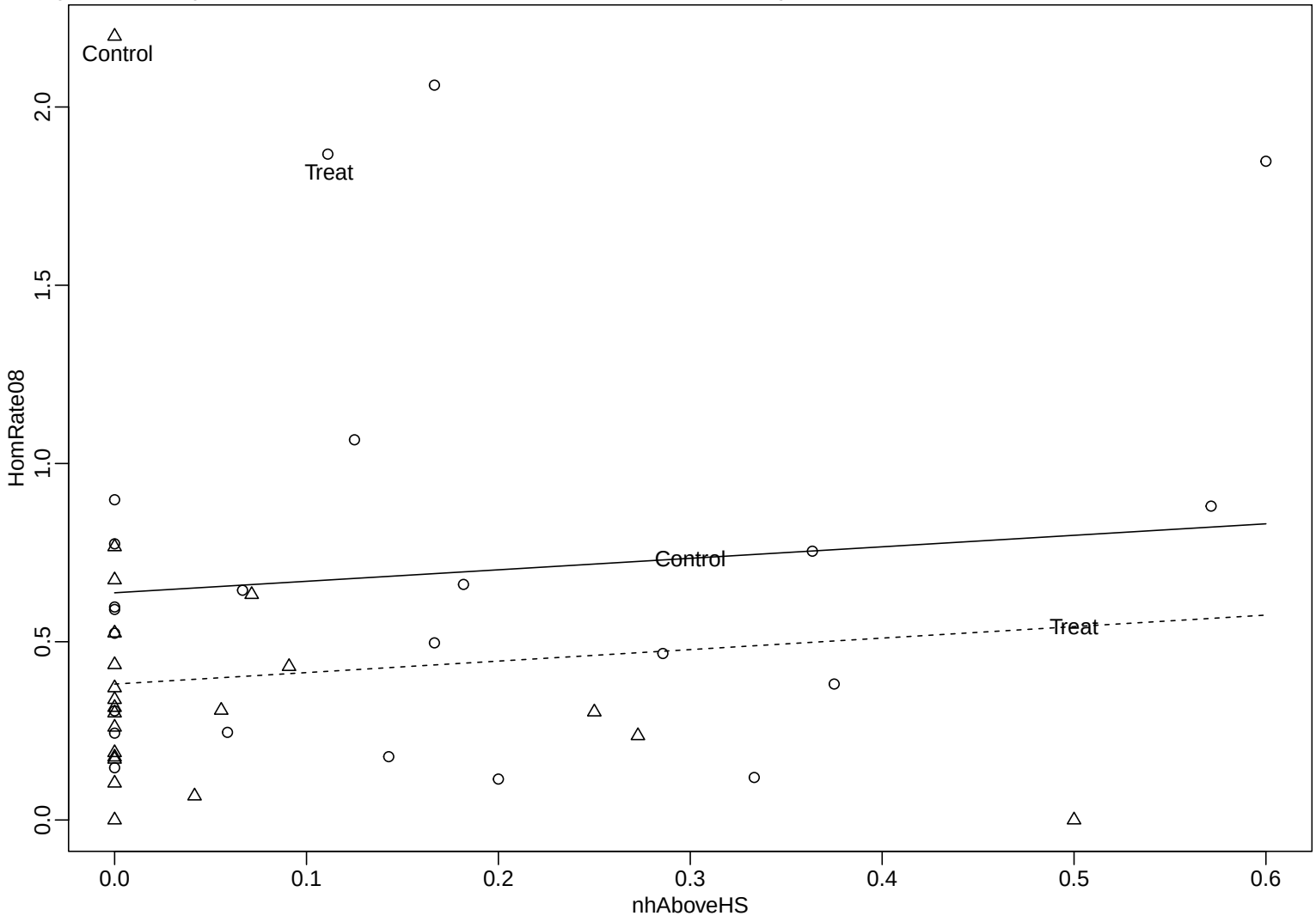
```
, , strata = --

           stat
vars        Control Treatment std.diff adj.diff pooled.sd      z       p
  nhAboveHS   0.163   0.05829  -0.6708  -0.1047    0.1561 -2.144 0.03201

attr(,"NMpatterns")
[1] "(_any Xs recorded_)"
attr(,"originals")
[1] 1
attr(,"term.labels")
[1] "nhAboveHS"
attr(,"include.NA.flags")
[1] TRUE
```

How would you adjust for Proportion Above HS Degree?

Part of the Metrocable effect is not about Metrocable per se, but rather about the education of people in the neighborhood. How should we remove `nhAboveHS` from our estimate or test? What strategies can you think of?

One approach to this problem: model-based adjustment

Let's try to just adjust for this covariate in a very common manner:



Exactly what does this kind of adjustment do?

Notice that I can get the same coefficient (the effect of Metrocable on Homicides adjusted for HS-Education in the neighborhood) either directly (as earlier) or via **residualization**:

```
coef(lm1)["nhTrt"]
```

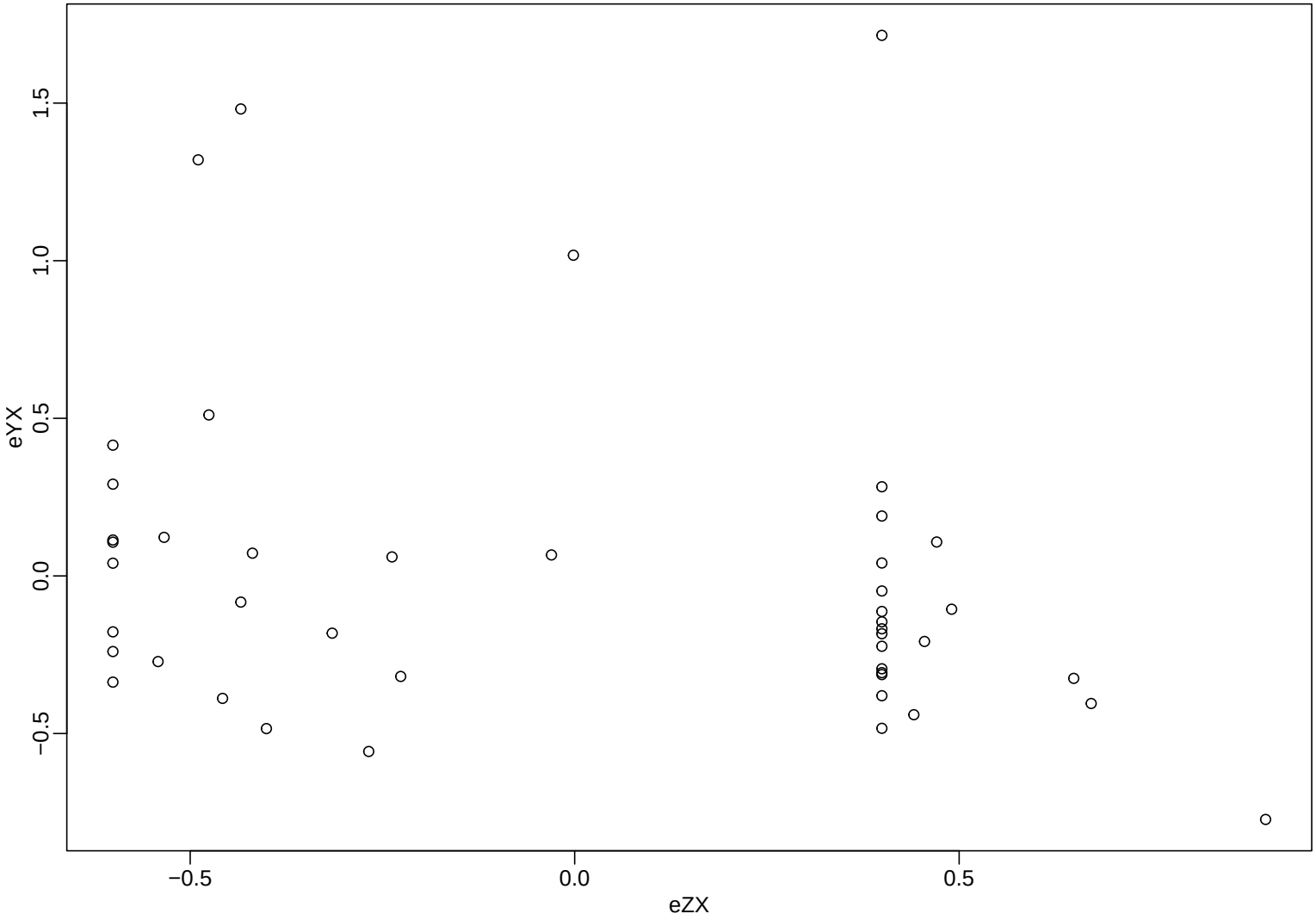
```
  nhTrt
-0.2561
```

```
eYX <- residuals(lm(HomRate08 ~ nhAboveHS, data = meddat))
eZX <- residuals(lm(nhTrt ~ nhAboveHS, data = meddat))
lm1a <- lm(eYX ~ eZX)
coef(lm1a)[2]
```

```
    eZX
-0.2561
```

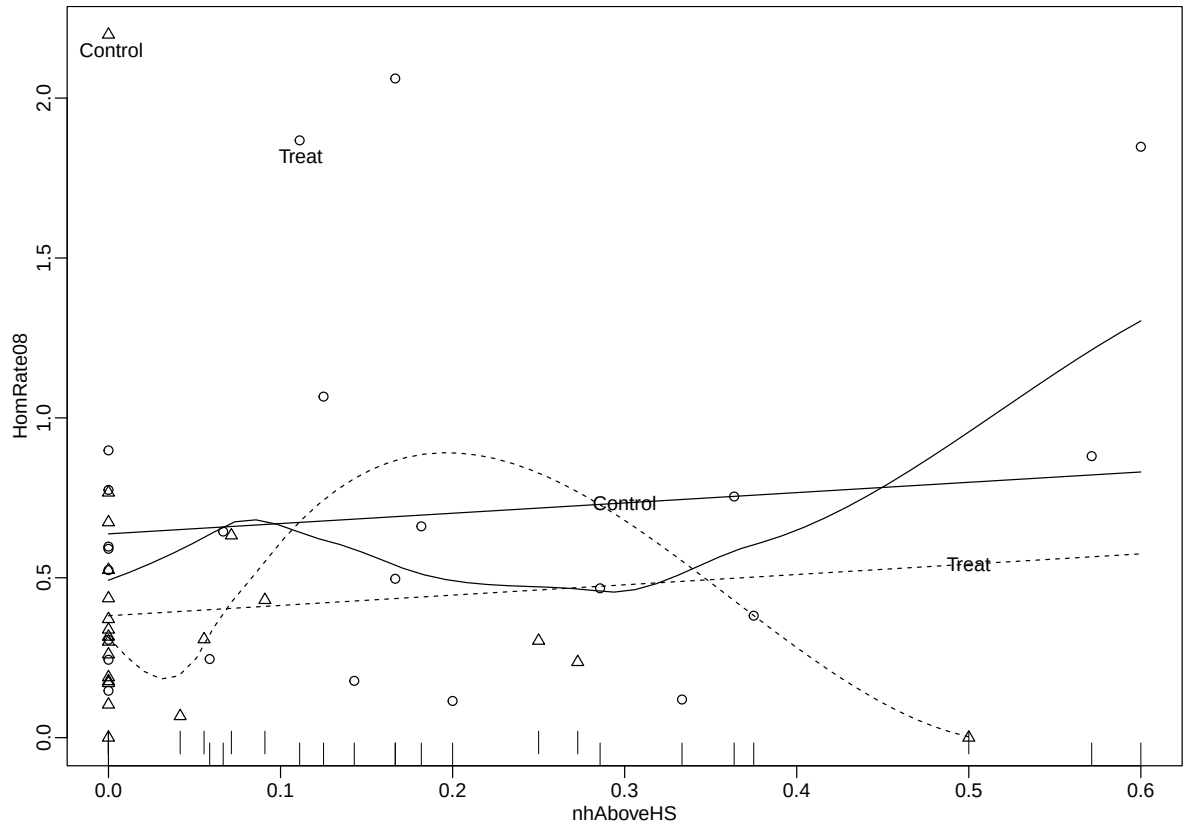
Exactly what does this kind of adjustment do?

So, how would you explain what it means to "control for HS-Education" here?



Did we adjust enough?

Maybe adding some more information to the plot can help us decide whether, and to what extend, we effectively "controlled for" the proportion of the neighborhood with more than High School education. Specifically, we might be interested in assessing extrapolation/interpolation problems arising from our linear assumptions.



How should we interpret this adjustment? How should we judge the improvement that we made? What concerns might we have?

How would you adjust for Proportion Above HS Degree?

So, part of the Metrocable effect might not reflect the causal effect of Metrocable per se, but rather the education of people in the neighborhood. How should we remove `nhAboveHS` from our estimate or test? What strategies can you think of?
Features of a good adjustment process:

- Blind to outcome analysis (to preserve false positive rate and deter critics). Able to be pre-registered. Perhaps even reviewed by stakeholders.
- Easy to interpret ("controlling for" versus "holding constant")
- Easy to diagnoses (Easy to answere the question "Did we adjust enough?")

Stratification V 1.0

```
lm1a <- lm(HomRate08 ~ nhTrt, data = meddat, subset = nhAboveHS >= .1)
lm1b <- lm(HomRate08 ~ nhTrt, data = meddat, subset = nhAboveHS < .1)
res_strat <- c(hiEd_Effect = coef(lm1a)["nhTrt"], loEd_Effect = coef(lm1b)["nhTrt"])
res_strat
```

```
hiEd_Effect.nhTrt loEd_Effect.nhTrt
         -0.65828          -0.06237
```

```
n_strat <- table(meddat$nhAboveHS >= .1)
n_strat
```

```

FALSE  TRUE
   29    16
```

```
stopifnot(sum(n_strat) == nrow(meddat)) ## A test of code
sum(res_strat * rev(n_strat) / 45) ## What is happening here?
```

```
[1] -0.2743
```

```
## Putting this together
outcome_analysis_strat <- meddat %>%
  group_by(nhAboveHS >= .1) %>%
  summarize(
    nb = n(), nT = sum(nhTrt), nC = nb - nT, pr_trt = mean(nhTrt), bar_y_t = mean(HomRate08[nhTrt
    bar_y_c = mean(HomRate08[nhTrt == 0]), ate_b = bar_y_t - bar_y_c
  )

outcome_analysis_strat <- outcome_analysis_strat %>% mutate(
  nbwt = nb / sum(nb),
  prec_wt0 = nbwt * pr_trt * (1 - pr_trt)
)
```

Stratified adjustment V 2.0

One-step stratified estimation.

```
## Weight by block size
ate1c <- difference_in_means(HomRate08 ~ nhTrt, blocks = I(nhAboveHS >= .1), data = meddat)
ate1c
```

```
Design:  Blocked
      Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
nhTrt  -0.2743     0.1146  -2.393   0.0214  -0.5057 -0.04276 41
```

```
## Weight by both block size and  proportion in treatment vs control ("harmonic weight")
lm1c <- lm_robust(HomRate08 ~ nhTrt, fixed_effects = ~ I(nhAboveHS >= .1), data = meddat)
coef(lm1c)["nhTrt"]
```

```
 nhTrt
-0.224
```

```
lm1d <- lm(HomRate08 ~ nhTrt + I(nhAboveHS >= .1), data = meddat)
coef(lm1d)["nhTrt"]
```

```
 nhTrt
-0.224
```

Stratified adjustment V 2.0 I

One-step stratified testing

```
xbate1 <- balanceTest(nhTrt ~ HomRate08 + strata(I(nhAboveHS >= .1)), data = meddat)
xbate1$results[, c("adj.diff", "p"), ]
```

```
          strata
stat       I(nhAboveHS >= 0.1)       --
  adj.diff             -0.1436 -0.28986
  p                     0.1724  0.06317
```

```
xbate1$overall
```

```
                    chisquare df p.value
I(nhAboveHS >= 0.1)     1.862  1 0.17238
--                      3.452  1 0.06317
```

```
## Effect of the treatment on the treated weights
outcome_analysis_strat$nTwt <- with(outcome_analysis_strat, nT / sum(nT))
with(outcome_analysis_strat, sum(ate_b * nTwt))
```

```
[1] -0.1436
```

Stratified adjustment V 2.0 II

```
## Approximating the as-if-randomized null distribution with a Normal
## approximation
hstest2 <- independence_test(HomRate08 ~ nhTrt | factor(nhAboveHS >= .1), data = meddat)

## Now using the "as-if-randomized" distribution directly
set.seed(12345)
hstest2_perm <- independence_test(HomRate08 ~ nhTrt | factor(nhAboveHS >= .1), data = meddat, dis

pvalue(hstest2)
```

```
[1] 0.1724
```

```
pvalue(hstest2_perm)
```

```
[1] 0.18
99 percent confidence interval:
 0.1702 0.1901
```

```
## Now trying different test statistics
hstest4 <- independence_test(HomRate08 ~ nhTrt | factor(nhAboveHS >= .1), data = meddat, ytrafo =
pvalue(hstest4)
```

```
[1] 0.03643
```

```
hstest5 <- wilcox_test(HomRate08 ~ factor(nhTrt) | factor(nhAboveHS >= .1), data = meddat)
pvalue(hstest5)
```

```
[1] 0.03643
```

Balance assessment after stratification

Did we adjust enough? What would enough mean? Use the testing approach but now focus only on the covariate(s) that you are trying to adjust.

```
xbHS1 <- balanceTest(nhTrt ~ nhAboveHS + strata(I(nhAboveHS >= .1)), data = meddat)
xbHS1$overall
```

```
                    chisquare df p.value
I(nhAboveHS >= 0.1)    0.3912  1 0.53166
--                     4.5979  1 0.03201
```

```
xbHS1$results[1, c("Treatment", "Control", "adj.diff", "std.diff", "z", "p"), ] ## the covariate
```

```
           strata
stat        I(nhAboveHS >= 0.1)       --
  Treatment            0.058286  0.05829
  Control              0.048844  0.16299
  adj.diff             0.009442 -0.10470
  std.diff             0.060489 -0.67076
  z                    0.625467 -2.14426
  p                    0.531665  0.03201
```

Disadvantages and Advantages of Simple Stratification

- (+) Easy to explain what "controlling for" or "adjustment" means.
- (-) Hard to justify any particular cut-point or number of cut-points / groups
- (-) We could probably adjust more — comparing neighborhoods similar in education rather than just within big strata

Can we improve stratified adjustment?

Rather than two strata, why not three?

```
lm1cut3 <- lm(HomRate08 ~ nhTrt + cut(nhAboveHS, 3), data = meddat)
coef(lm1cut3)["nhTrt"]
```

```
  nhTrt
-0.3161
```

But why those cuts? And why not 4? Why not...?

One idea: collect observations into strata such that the sum of the differences in means of `nhAboveHS` within strata is smallest? This is the idea behind `optmatch` and other matching approaches.

The optmatch workflow: The distance matrix

Introduction to optmatch workflow. To minimize differences requires a matrix of those differences (in general terms, a matrix of distances between the treated and control units)

```
tmp <- meddat$nhAboveHS
names(tmp) <- rownames(meddat)
absdist <- match_on(tmp, z = meddat$nhTrt, data = meddat)
absdist[1:3, 1:3]
```

```
         control
treatment    401     402    403
      101 0.1429 0.00000 0.1667
      102 0.1429 0.00000 0.1667
      103 0.0873 0.05556 0.1111
```

```
abs(meddat$nhAboveHS[meddat$nhTrt == 1][1] - meddat$nhAboveHS[meddat$nhTrt == 0][1])
```

```
[1] 0.1429
```

Created a Stratified Research Design

Here we create two stratified designs that minimize differences in proportion above HS education between neighborhoods with and without the new Metrocable stations:

```
fm1 <- fullmatch(absdist, data = meddat)
summary(fm1, min.controls = 0, max.controls = Inf)
```

```
Structure of matched sets:
8:1 2:1 1:1 1:2 1:4 1:5
  1   1   8   2   1   1
Effective Sample Size:  17
(equivalent number of matched pairs).
```

```
table(meddat$nhTrt, fm1)
```

```
   fm1
    1.1 1.14 1.15 1.16 1.17 1.18 1.19 1.2 1.20 1.21 1.22 1.3 1.6 1.7
  0   1    2    2    1    1    1    1   1    1    1    5   1   4   1
  1   1    1    1    1    1    1    1   8    1    1    1   2   1   1
```

```
pm1 <- pairmatch(absdist, data = meddat)
summary(pm1, min.controls = 0, max.controls = Inf)
```

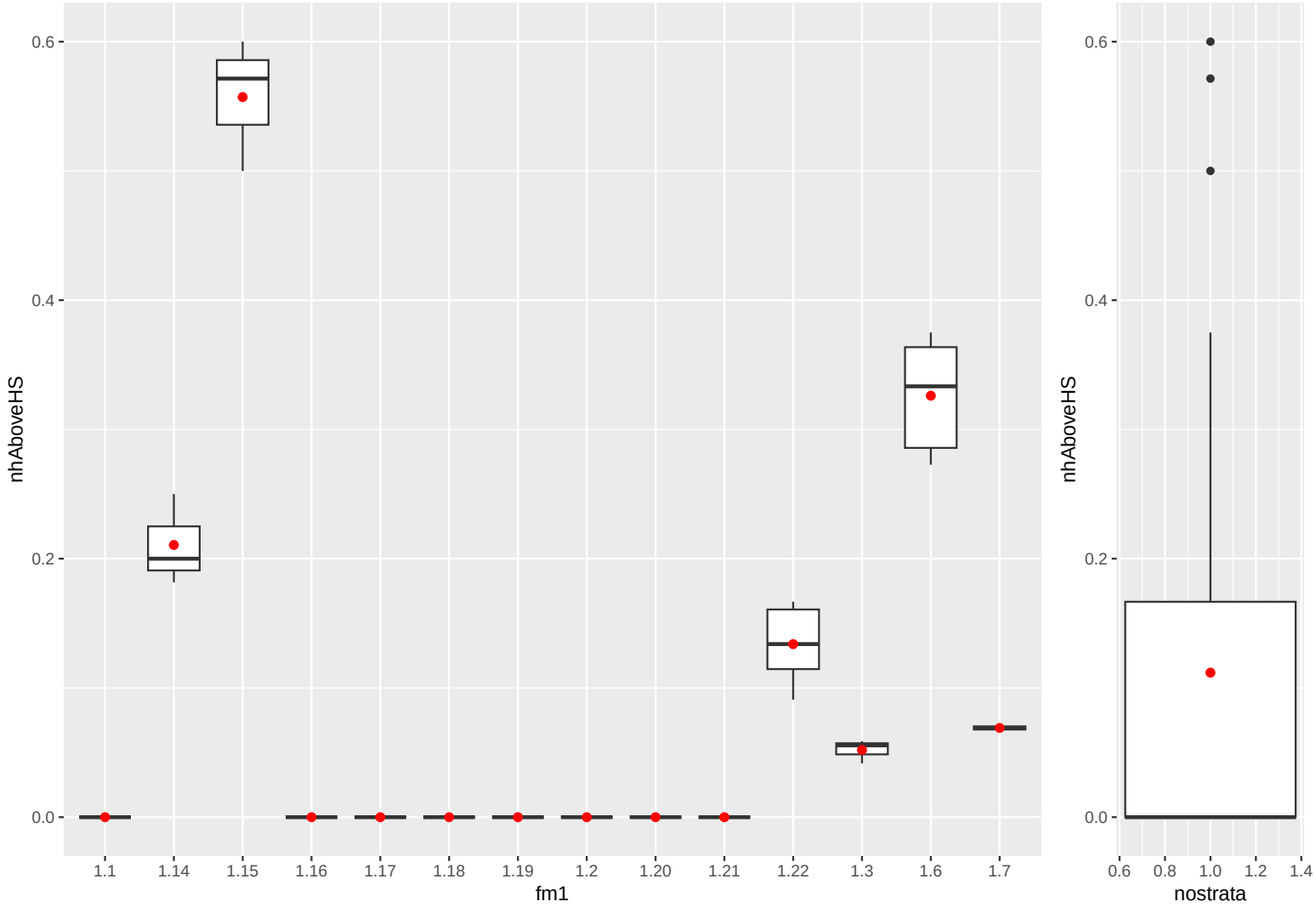
```
Structure of matched sets:
1:1 0:1
 22   1
Effective Sample Size:  22
(equivalent number of matched pairs).
```

```
table(meddat$nhTrt, pm1, exclude = c())
```

```
   pm1
    1.1 1.10 1.11 1.12 1.13 1.14 1.15 1.16 1.17 1.18 1.19 1.2 1.20 1.21 1.22 1.3 1.4 1.5 1.6 1.7
```

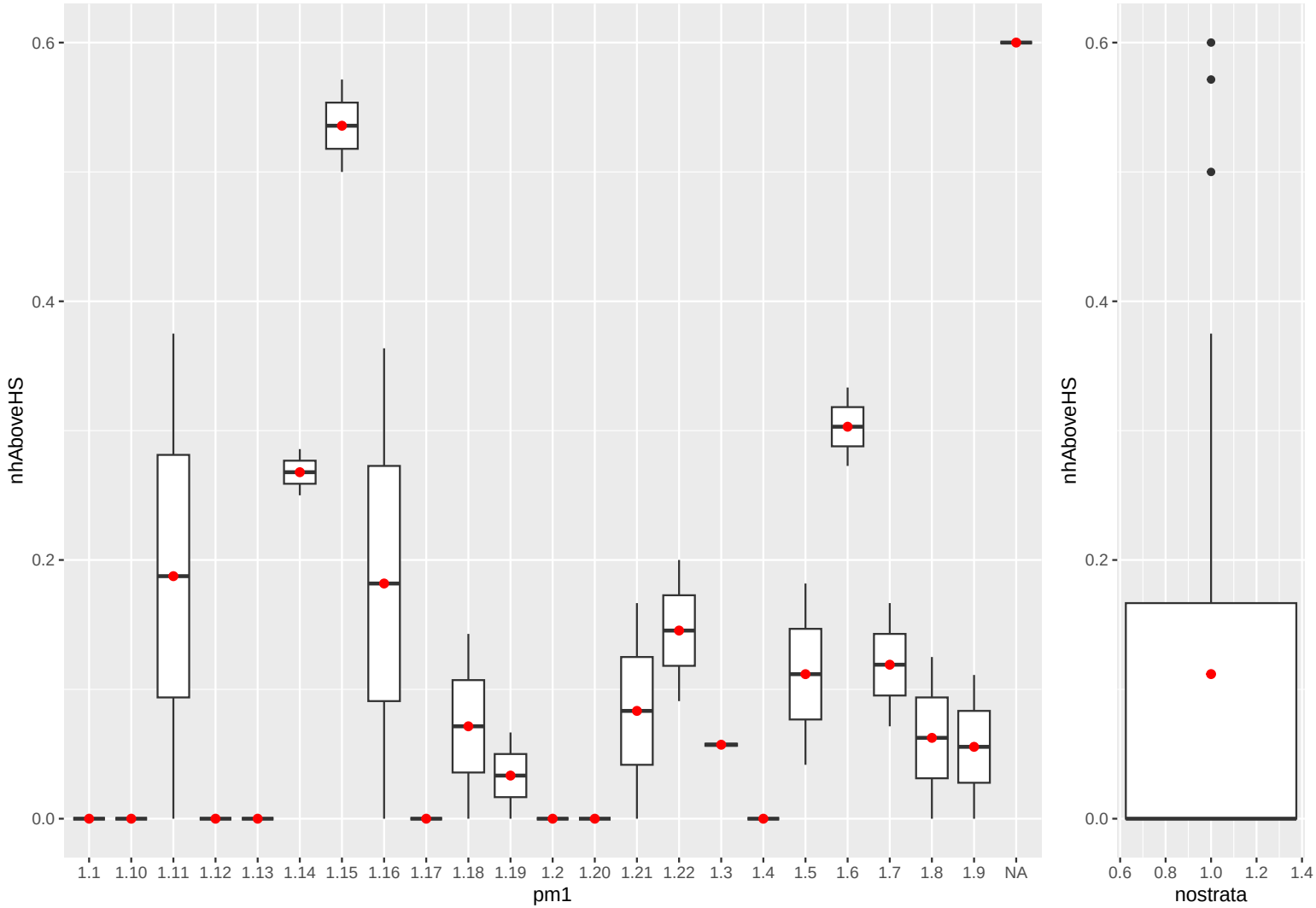
Evaluate the design: Within set differences

Differences within sets versus raw differences from the fullmatch.



Evaluate the design: Within set differences

Differences within sets versus raw differences from the pairmatch.



Evaluate the design: Inspect within set differences

```
# A tibble: 14 x 5
   fm1   mneddiffs mnAboveHS minAboveHS maxAboveHS
   <fct>     <dbl>     <dbl>      <dbl>      <dbl>
 1 1.1     0          0          0          0
 2 1.14    0.0591     0.211      0.182      0.25
 3 1.15   -0.0857     0.557      0.5        0.6
 4 1.16    0          0          0          0
 5 1.17    0          0          0          0
 6 1.18    0          0          0          0
 7 1.19    0          0          0          0
 8 1.2     0          0          0          0
 9 1.20    0          0          0          0
10 1.21    0          0          0          0
11 1.22   -0.0516     0.134      0.0909     0.167
12 1.3    -0.0102     0.0520     0.0417     0.0588
13 1.6    -0.0667     0.326      0.273      0.375
14 1.7     0.00476    0.0690     0.0667     0.0714
```

Evaluate the design: Inspect within set differences

```
# A tibble: 23 x 5
   pm1   mneddiffs mnAboveHS minAboveHS maxAboveHS
   <fct>     <dbl>     <dbl>      <dbl>      <dbl>
 1 1.1       0         0           0         0
 2 1.10      0         0           0         0
 3 1.11     -0.375     0.188       0         0.375
 4 1.12      0         0           0         0
 5 1.13      0         0           0         0
 6 1.14     -0.0357    0.268       0.25      0.286
 7 1.15     -0.0714    0.536       0.5       0.571
 8 1.16     -0.364     0.182       0         0.364
 9 1.17      0         0           0         0
10 1.18     -0.143     0.0714      0         0.143
# i 13 more rows
```

Evaluate the design: Compare to a randomized experiment.

The within-set differences look different from those that would be expected from a randomized experiment.

```
xbfm1 <- balanceTest(nhTrt ~ nhAboveHS + strata(fm1), data = meddat)
xbfm1$results
```

```
, , strata = fm1

           stat
vars        Control Treatment std.diff  adj.diff pooled.sd      z       p
  nhAboveHS 0.06558   0.05829 -0.04675 -0.007297    0.1561 -1.658 0.09728

, , strata = --

           stat
vars        Control Treatment std.diff adj.diff pooled.sd      z       p
  nhAboveHS   0.163   0.05829  -0.6708  -0.1047    0.1561 -2.144 0.03201

attr(,"NMpatterns")
[1] "(_any Xs recorded_)"
attr(,"originals")
[1] 1
attr(,"term.labels")
[1] "nhAboveHS"
attr(,"include.NA.flags")
[1] TRUE
```

```
xbfm1$overall
```

```
    chisquare df p.value
fm1     2.750  1 0.09728
```

What is balanceTest doing?

It compares the strata-weighted average of within-strata differences to that which would be expected if we were to repeat an experiment with the same stratified design and same covariate values (and `balanceTest` uses a large sample Normal approximation to this distribution.)

```
setmeanDiffs <- meddat %>%
  group_by(fm1) %>%
  summarise(
    diffAboveHS = mean(nhAboveHS[nhTrt == 1]) - mean(nhAboveHS[nhTrt == 0]),
    nb = n(),
    nTb = sum(nhTrt),
    nCb = sum(1 - nhTrt),
    hwt = (2 * (nCb * nTb) / (nTb + nCb))
  )
setmeanDiffs
```

```
# A tibble: 14 x 6
   fm1   diffAboveHS    nb   nTb   nCb   hwt
   <fct>       <dbl> <int> <int> <dbl> <dbl>
 1 1.1       0           2     1     1  1
 2 1.14      0.0591      3     1     2  1.33
 3 1.15     -0.0857      3     1     2  1.33
 4 1.16      0           2     1     1  1
 5 1.17      0           2     1     1  1
 6 1.18      0           2     1     1  1
 7 1.19      0           2     1     1  1
 8 1.2       0           9     8     1  1.78
 9 1.20      0           2     1     1  1
10 1.21      0           2     1     1  1
```

What is balanceTest doing with multiple sets/blocks?

The test statistic is a weighted average of the set-specific differences (same approach as we would use to test the null in a block-randomized experiment)

```
## The descriptive mean difference using block-size weights
with(setmeanDiffs, sum(diffAboveHS * nTb / sum(nTb)))
```

```
[1] -0.007297
```

```
## The mean diff used as the observed value in the testing
with(setmeanDiffs, sum(diffAboveHS * hwt / sum(hwt)))
```

```
[1] -0.0139
```

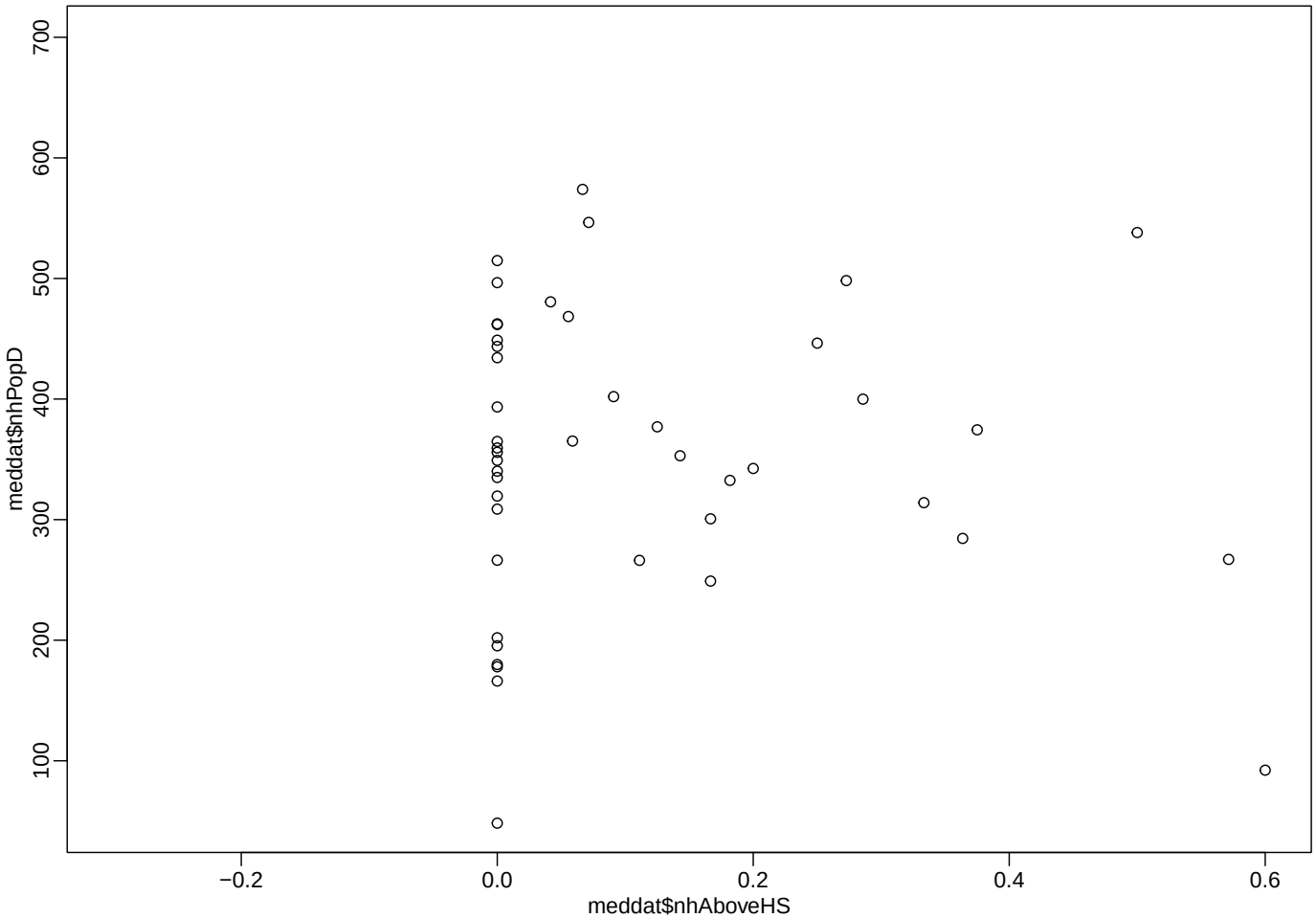
```
## Compare to balanceTest output
xbfm1$results[, , "fm1"]
```

```
  Control Treatment  std.diff  adj.diff pooled.sd         z         p
 0.065583  0.058286 -0.046746 -0.007297  0.156095 -1.658189  0.097279
```

1. Overview and Review
2. How to assess the randomization process in an experiment (to teach us how to assess research designs in observational studies).
3. Assessing comparisons in observational studies
4. Matching on Many Covariates: Using the Mahalnobis Distance to Scale Euclidean Distance
5. Matching on Many Covariates: Using Propensity Scores
6. Matching Tricks of the Trade: Calipers, Exact Matching
7. The separation problem in Logistic Regression

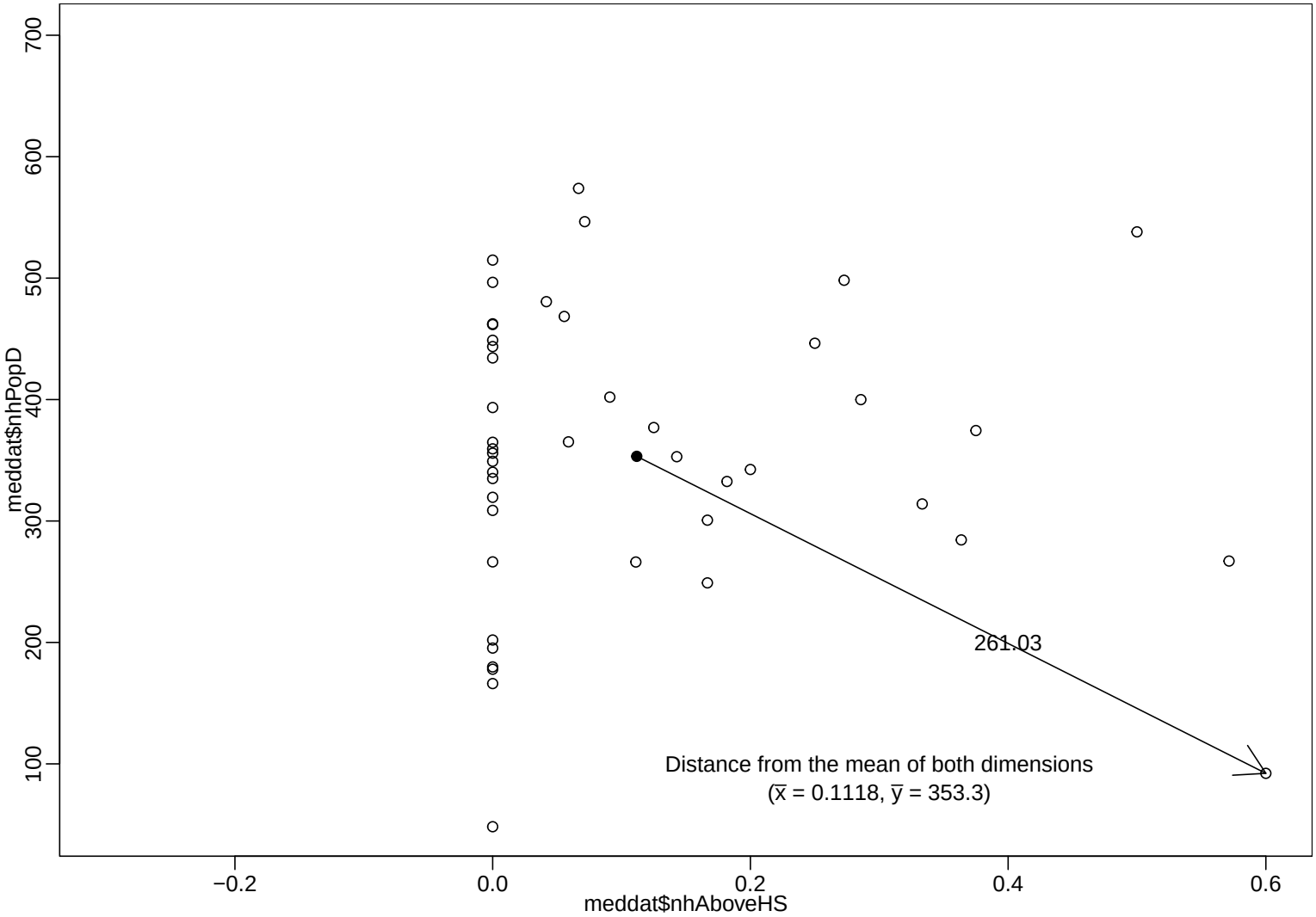
Dimension reduction using the Mahalanobis Distance

The general idea: dimension reduction. When we convert many columns into one column we reduce the dimensions of the dataset (to one column). We can use the idea of **multivariate distance** to produce distance matrices to minimize **multivariate distances**.



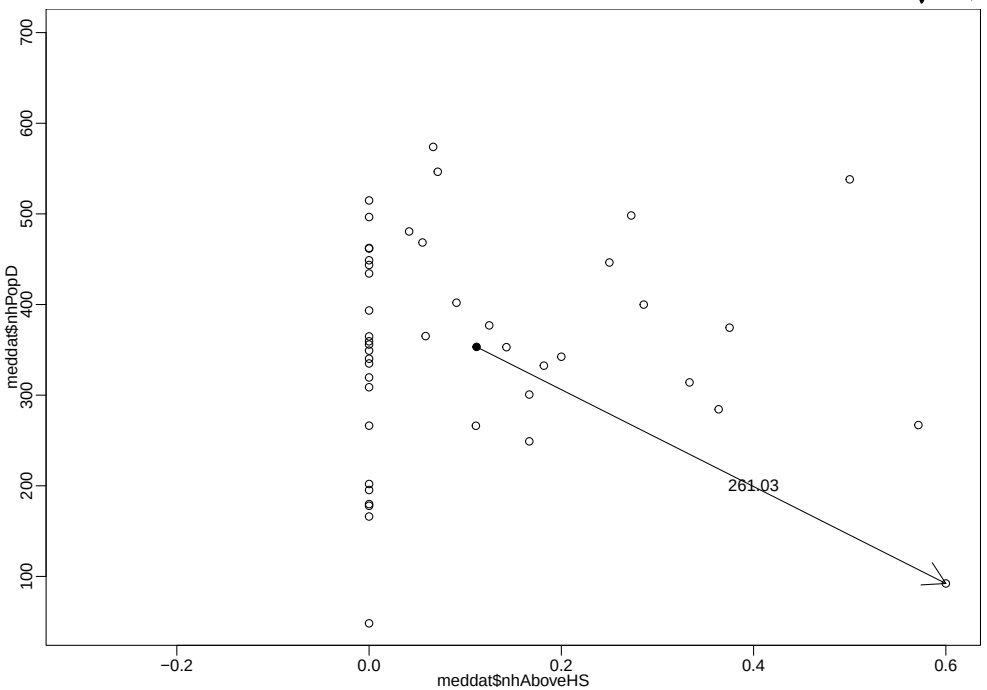
Dimension reduction using the Mahalanobis Distance

First, let's look at Euclidean distance: $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$



Dimension reduction using the Mahalanobis Distance

First, let's look at Euclidean distance: $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$



Distance between point in middle of the plot and unit "407".

```
tmp <- rbind(colMeans(X), X["407", ])
tmp
```

```
    nhAboveHS nhPopD
1      0.1118 353.25
407    0.6000  92.22
```

```
sqrt((tmp[1, 1] - tmp[2, 1])^2 + (tmp[1, 2] - tmp[2, 2])^2)
```

```
[1] 261
```

Problem: overweights variables with bigger scales (Population Density dominates

Dimension reduction using the Mahalanobis Distance

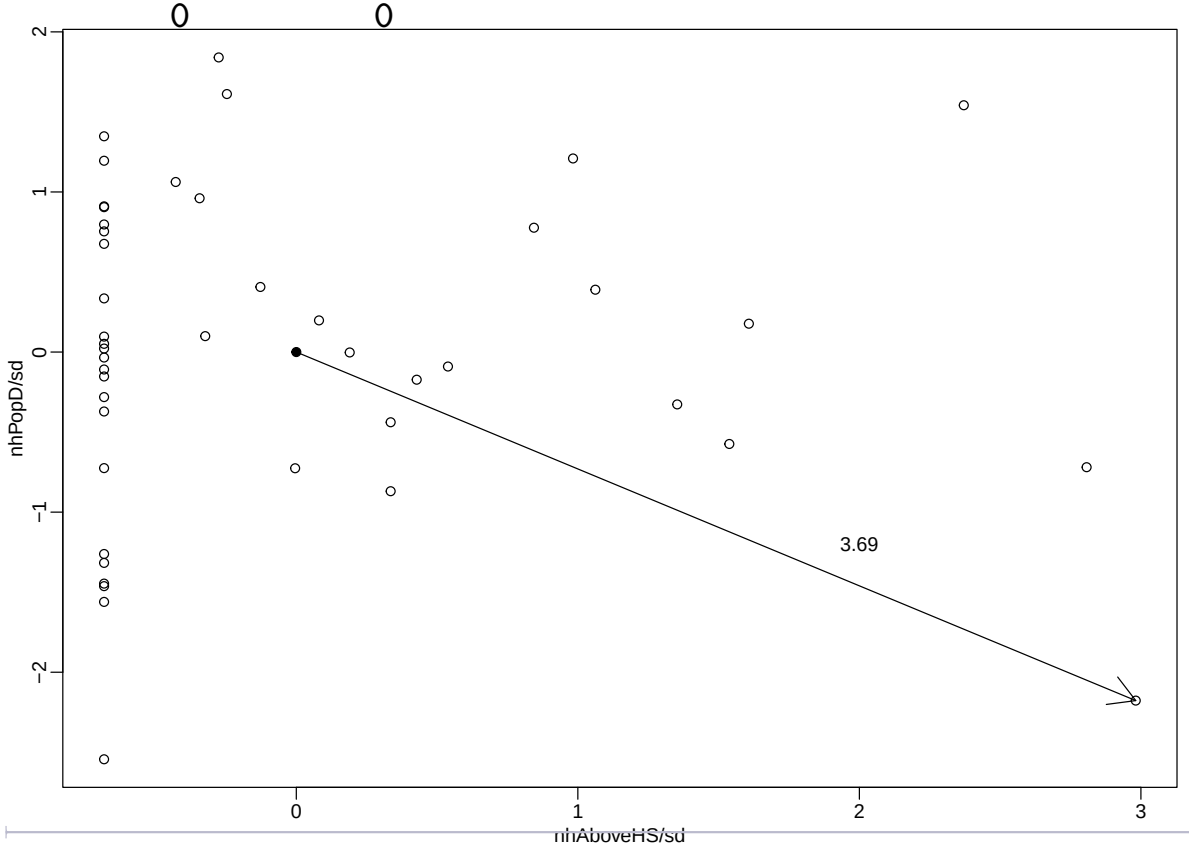
Now the standardized Euclidean distance so neither variable is overly dominant.

```
Xsd <- scale(X)
apply(Xsd, 2, sd) ## should be 1
```

```
nhAboveHS    nhPopD
        1         1
```

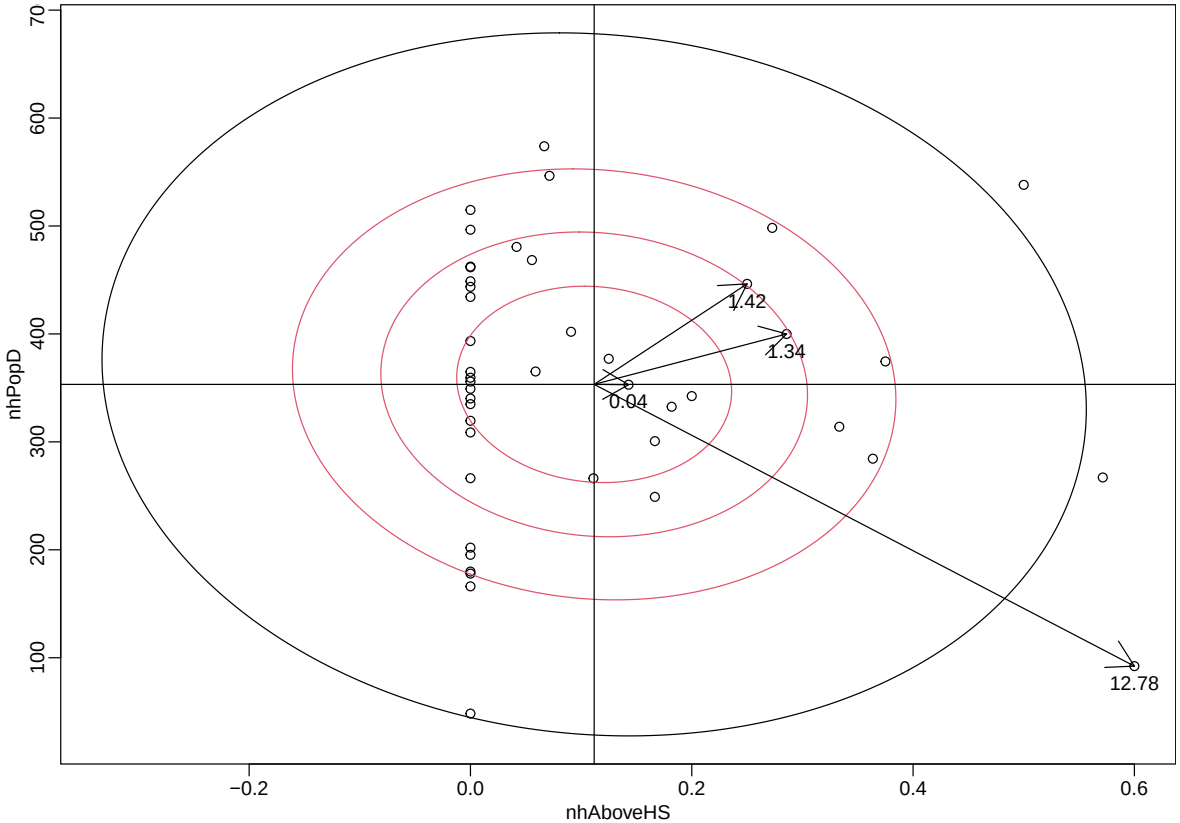
```
round(apply(Xsd, 2, mean), 8) ## should be 0
```

```
nhAboveHS    nhPopD
        0         0
```



Dimension reduction using the Mahalanobis Distance

The mahalanobis distance avoids the scale problem in the euclidean distance.[1] Here each circle are points of the same MH distance.




```
[1]  0.00 12.78
nhAboveHS
    3.692
```

[1] For more see here

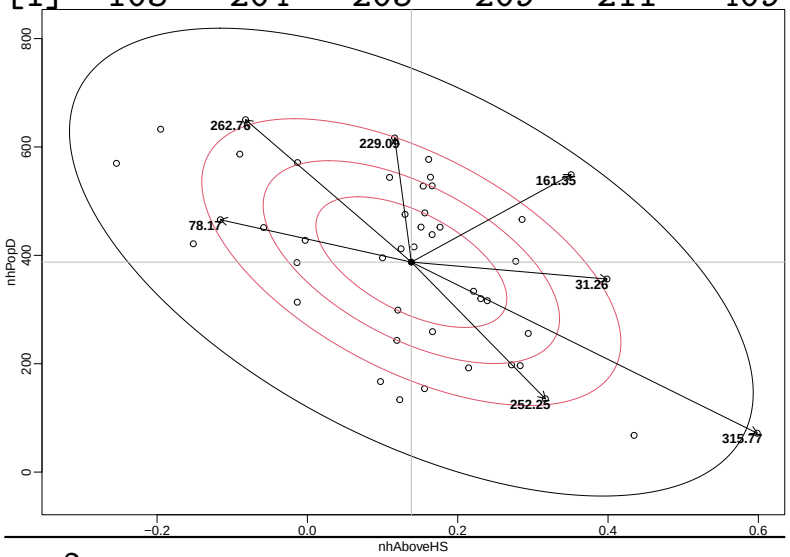
How is Mahalanobis distance different from Scaled Euclidean Distance?

To Review: The Mahalanobis distance (Mahalanobis, 1930), avoids the scale and correlation problem in the euclidean distance.[2] $dist_M = \sqrt{(\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{M}^{-1} (\mathbf{y} - \bar{\mathbf{y}})}$

where $\mathbf{M} = \begin{bmatrix} \mathbf{V}(x) & \mathbf{Cov}(x,y) \\ \mathbf{Cov}(x,y) & \mathbf{V}(y) \end{bmatrix}$

Here, using simulated data: The contour lines show points with the same Mahalanobis distance and the numbers are Euclidean distance.

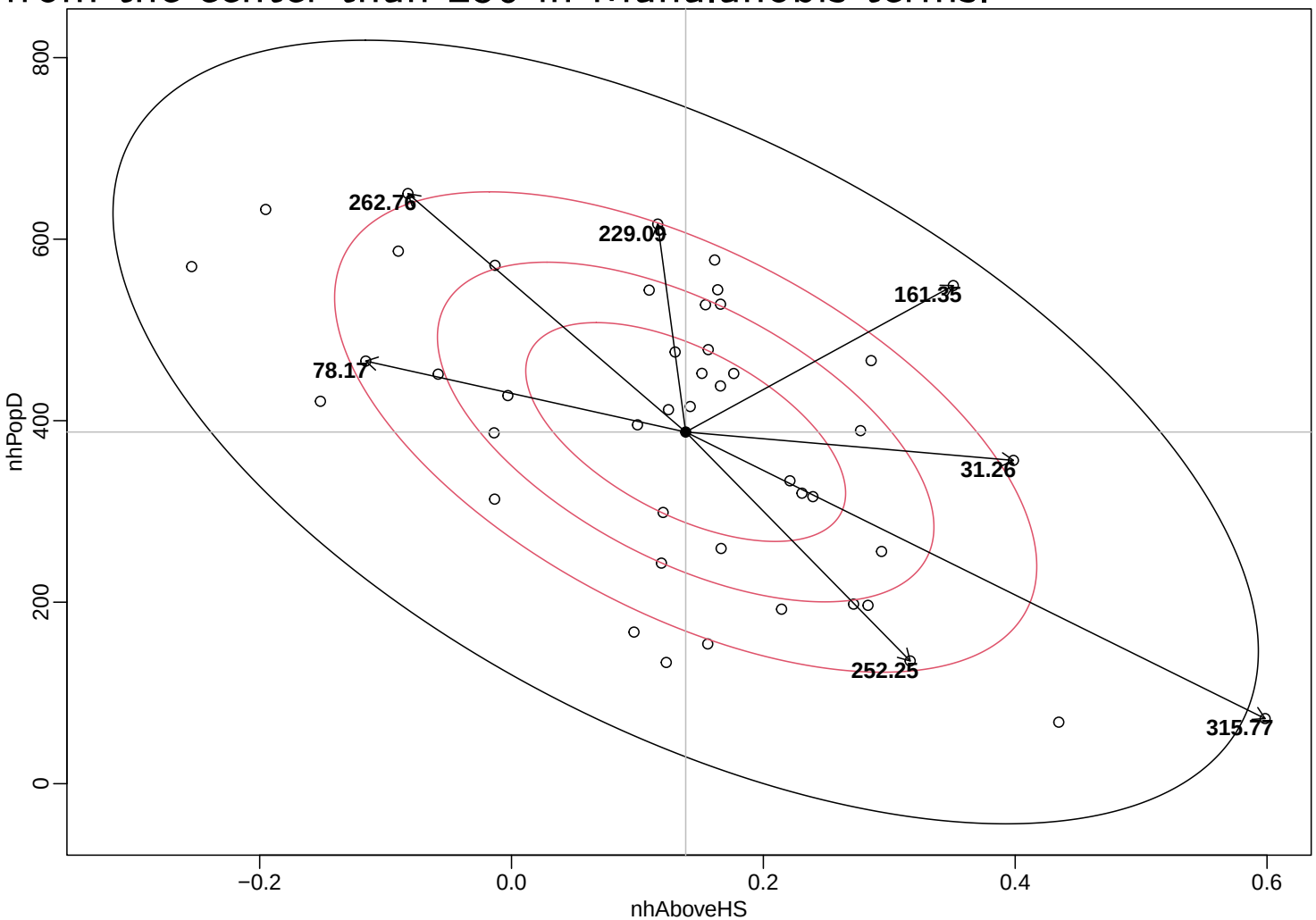
```
           nhAboveHS  nhPopD
nhAboveHS     1.0000 -0.5592
nhPopD       -0.5592  1.0000
[1] "103" "204" "205" "209" "211" "409" "810"
```



[2] For more see https://stats.stackexchange.com/questions/62092/bottom-to-top-explanation-of-the-mahalanobis-distance

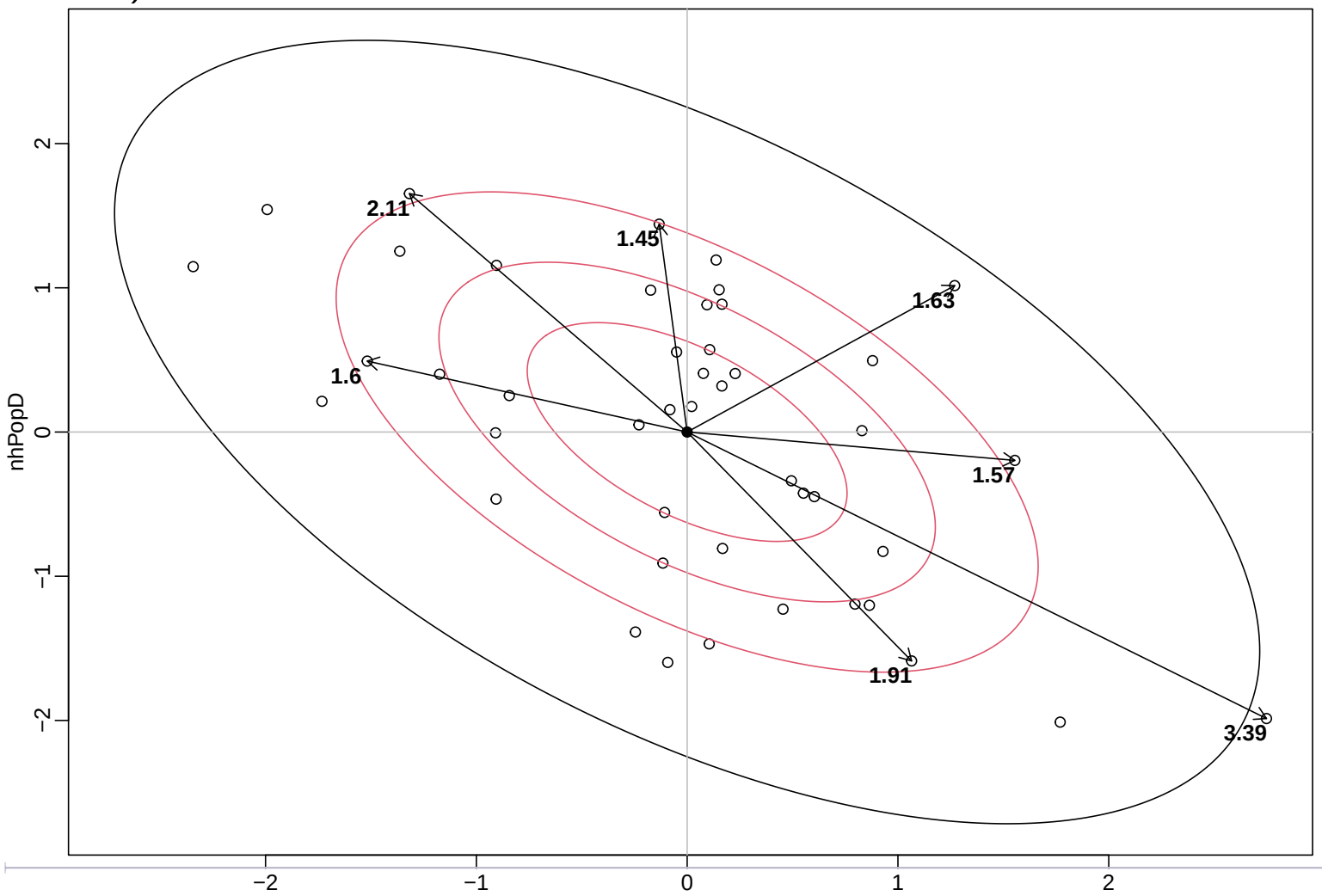
Dimension reduction using the Mahalanobis distance

The contour lines show points with the same Mahalanobis distance, the numbers are Euclidean distance. Notice that the point with Euclidean distance of 161 is farther from the center than 250 in Mahalanobis terms.



Dimension reduction using the Mahalanobis distance

The contour lines show points with the same Mahalanobis distance and the numbers are Euclidean distance (**now on the standardized variables**). (notice that 1.63 is farther from the center in Mahalanobis terms than 2.11, but 2.11 is farther in Euclean terms.)



Matching on the Mahalanobis Distance

Here using the rank based Mahalanobis distance following DOS Chap. 8 (but comparing to the ordinary version).

```
mhdist <- match_on(nhTrt ~ nhPopD + nhAboveHS, data = meddat, method = "rank_mahalanobis")
mhdist[1:3, 1:3]
```

```
         control
treatment   401   402   403
      101 1.860 1.067 2.404
      102 1.999 1.296 1.744
      103 1.356 1.591 2.044
```

```
mhdist2 <- match_on(nhTrt ~ nhPopD + nhAboveHS, data = meddat)
mhdist2[1:3, 1:3]
```

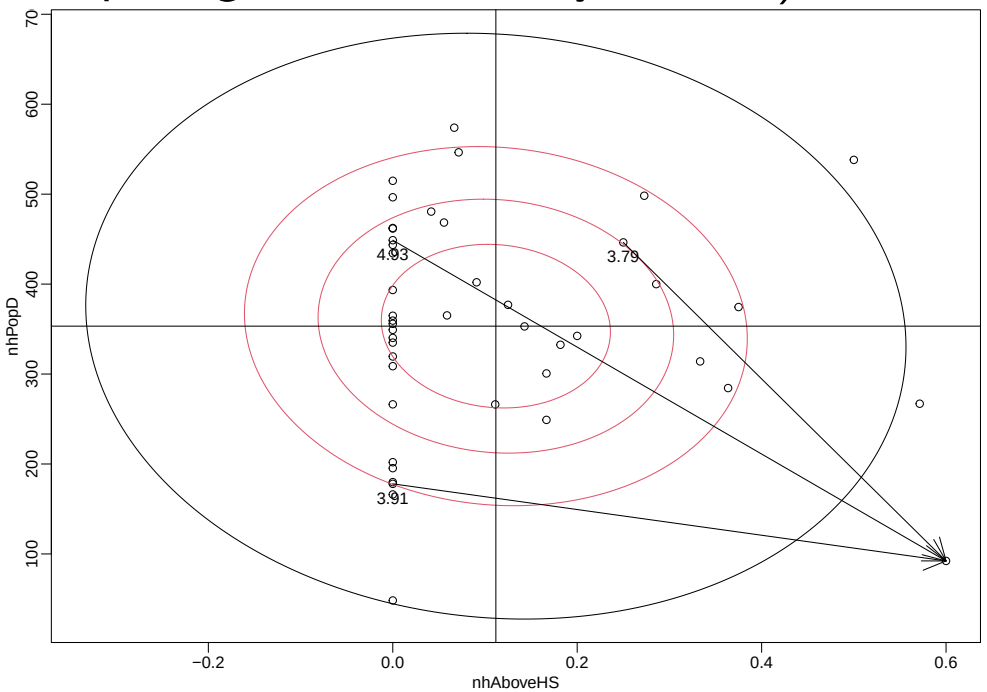
```
         control
treatment   401    402   403
      101 1.235 0.8513 1.665
      102 1.739 1.4626 1.482
      103 1.140 1.0738 1.609
```

```
mhdist2[, "407"]
```

```
  101   102   103   104   105   106   107   108   109   110   111   112   201   202   203   204
4.929 3.909 4.768 5.002 4.905 4.076 5.180 4.319 3.912 4.277 3.841 3.890 5.195 3.788 3.872 4.382 4
```

Matching on the Mahalanobis Distance

Here using the rank based Mahalanobis distance following DOS Chap. 8 (but comparing to the ordinary version).




```
mhdist2[tpts, "407"]
```

```
  101   102   202
4.929 3.909 3.788
```

Matching on the Mahalanobis Distance

```
mhdist <- match_on(nhTrt ~ nhPopD + nhAboveHS, data = meddat, method = "rank_mahalanobis")

fmMh <- fullmatch(mhdist, data = meddat)
summary(fmMh, min.controls = 0, max.controls = Inf)
```

```
Structure of matched sets:
6:1 2:1 1:1 1:2 1:3 1:5
  1   1  11   1   1   1
Effective Sample Size:  18.5
(equivalent number of matched pairs).
```

```
summary(unlist(matched.distances(fmMh, mhdist)))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0762  0.2812  0.5009  0.7797  1.1823  2.5131
```

```
quantile(as.vector(mhdist), seq(0, 1, .1))
```

```
     0%     10%     20%     30%     40%     50%     60%     70%     80%     90%    100%
0.07624 0.67875 1.18235 1.44850 1.71964 1.93039 2.14689 2.33615 2.52358 2.81432 4.05058
```

```
fmMh1 <- fullmatch(mhdist + caliper(mhdist, 1), data = meddat) # , min.controls = 1)
summary(fmMh1, min.controls = 0, max.controls = Inf)
```

```
Structure of matched sets:
1:0 6:1 3:1 2:1 1:1 1:2 0:1
  1   1   1   1   9   1   9
Effective Sample Size:  14.9
(equivalent number of matched pairs).
```

```
summary(unlist(matched.distances(fmMh1, mhdist)))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

1. Overview and Review
2. How to assess the randomization process in an experiment (to teach us how to assess research designs in observational studies).
3. Assessing comparisons in observational studies
4. Matching on Many Covariates: Using the Mahalnobis Distance to Scale Euclidean Distance
5. Matching on Many Covariates: Using Propensity Scores
6. Matching Tricks of the Trade: Calipers, Exact Matching
7. The separation problem in Logistic Regression

The propensity score

Given covariates $\mathbf{x}(= (x_1, \ldots, x_k))$, and a treatment variable Z, $Z(u) \in \{0, 1\}$, $\mathbf{Pr}(Z|\mathbf{x})$ is known as the (true) **propensity score** (PS).

$$\phi(\mathbf{x}) \equiv \log\left(\mathbf{Pr}(Z = 1|\mathbf{x})/\mathbf{Pr}(Z = 0|\mathbf{x})\right)$$

is also known as the PS. In practice, one works with an estimated PS, $\widehat{\mathbf{Pr}}(Z|\mathbf{x})$ or $\widehat{\phi}(\mathbf{x})$.

Theoretically, propensity-score strata or matched sets both

1. reduce extrapolation; and
2. balance each of $x_1, \ldots, x_k$.

They do this by making the comparison more "experiment-like", at least in terms of $x_1, \ldots, x_k$.

Theory Paul R. Rosenbaum and Rubin (1983) also tells us that in the **absence of hidden bias**, such a stratification supports unbiased estimation of treatment effects.

Propensity scoring in practice

- Fitted propensity scores help identify extrapolation.
- In practice, stratification on $\widehat{\phi}(\mathbf{x})$ helps balance each of $x_1, \ldots, x_k$ compared to no stratification.

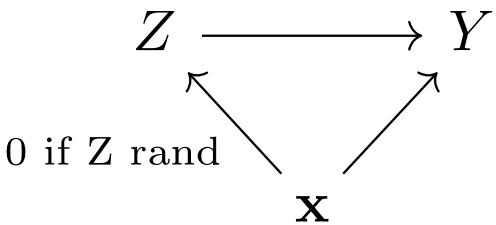
There are lots of cases in which adjustment with the propensity score alone fails to generate estimates that agree with those of randomized studies.
There are various reasons for this, starting with:

- lots of observational studies that don't measure quite enough $x$es or the right $x$es or the right $x$es in the right way
- **hidden biases** — propensity scores address bias on measured variables, not unmeasured ones.

Intuition about the propensity score

A propensity score is the output of a function of covariates as they relate to Z (the "treatment" or "intervention"). Why reduce the dimension of in this way rather than, say, using Mahalanobis distance?

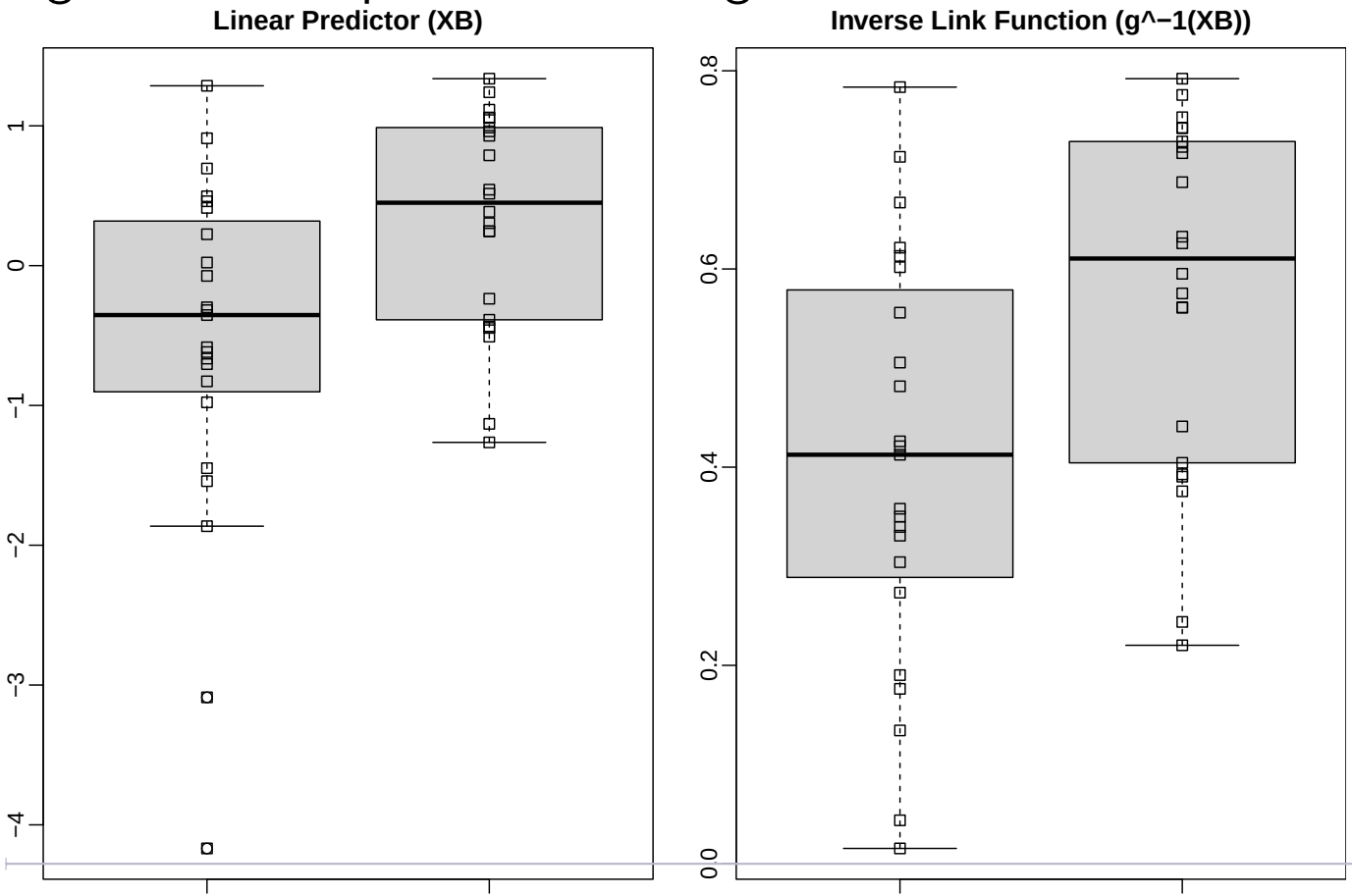
Recall that an experiment breaks the relationship between Z and $= \{x_1, x_2, ...\}$ but not between and Y or y_1, y_0.



Making strata of units who are similar on the propensity score reduces (or removes) the relationship between Z and the relevant $\mathbf{x}$ within strata (either the units have similar values for or the particular xs which do not have a strong (linear, additive) relationship with Z).

Matching on the propensity score

Make the score (Note that we will be using `brglm` or `bayesglm` in the future because of logit separation problems when the number of covariates increases.) We tend to match on the linear predictor rather than the version required to range only between 0 and 1. Recall how distance matrices required choices of distance metrics? We don't want to categorize two observations as "close" just because the logit function squashed them together near 0 or 1.



Matching on the propensity score: What do the distance matrix entries mean?

optmatch creates a scaled propensity score distance by default — scaling by, roughly, the pooled median absolute deviation of the covariate (or here, the propensity score). So, the distance matrix entries are like standard deviations — standardized scores.

```
          control
treatment    401    402    403    404
      101 1.4072 0.5772 1.8520 0.3207
      102 0.1616 0.9917 0.2831 1.2481
      103 1.1896 0.3595 1.6343 0.1031
      104 1.4858 0.6557 1.9305 0.3993
```

What do those distances mean?

```
[1] 0.9014
[1] 0.9253
[1] 0.9133
   401    402    403
1.2858 0.5274 1.6921
   401    402    403
1.4072 0.5772 1.8520
   401    402    403
1.4072 0.5772 1.8520
```

Matching on the propensity score

The following design balances the two covariates used in the creation of the propensity score well. It does not balance the baseline outcome well (not that we assumed it would, but demonstrating here that the covariates used for the creation of the design need not necessarily be all of those used to **evaluate** the design).

```
Structure of matched sets:
5:1 3:1 2:1 1:1 1:2 1:3 1:5
  1   1   1   8   2   1   1
Effective Sample Size:  18.3
(equivalent number of matched pairs).

Balance test overall result:
  chisquare df p.value
       1.16  2    0.56
     chisquare df p.value
fmPs     1.159  2  0.5603
     chisquare df p.value
fmPs     9.151  3 0.02734
```

Compare to Mahalanobis distance:

```
Structure of matched sets:
5+:1  2:1  1:1  1:2  1:3 1:5+
   1    1   11    1    1    1
Effective Sample Size:  18.5
(equivalent number of matched pairs).
        chisquare df  p.value
unstrat    13.585  3 0.003529
fmPs        9.151  3 0.027344
fmMh        6.115  3 0.106124
```

1. Overview and Review
2. How to assess the randomization process in an experiment (to teach us how to assess research designs in observational studies).
3. Assessing comparisons in observational studies
4. Matching on Many Covariates: Using the Mahalnobis Distance to Scale Euclidean Distance
5. Matching on Many Covariates: Using Propensity Scores
6. Matching Tricks of the Trade: Calipers, Exact Matching
7. The separation problem in Logistic Regression

Calipers

The optmatch package allows calipers (which forbids certain pairs from being matched).[3] Here, for example, we forbid comparisons which differ by more than 2 propensity score standardized distances.

```
     0%     10%     20%     30%     40%     50%     60%     70%     80%     90%    100%
0.01981 0.20101 0.40724 0.61788 0.90555 1.10614 1.39937 1.74453 2.08675 2.92582 6.02617
         control
treatment    405   407     408   409    410    411
      105 0.3911 5.579 1.40367 4.397 1.0973 1.9227
      106 1.6671 4.303 0.12776 3.121 0.1786 0.6468
      107 0.1871 5.783 1.60771 4.601 1.3014 2.1267
      108 1.0756 4.895 0.71918 3.712 0.4128 1.2382
      109 1.8847 4.086 0.08987 2.903 0.3962 0.4291
      110 1.1381 4.832 0.65670 3.650 0.3503 1.1757
       control
treated    405 407     408 409    410    411
    105 0.3911 Inf 1.40367 Inf 1.0973 1.9227
    106 1.6671 Inf 0.12776 Inf 0.1786 0.6468
    107 0.1871 Inf 1.60771 Inf 1.3014    Inf
    108 1.0756 Inf 0.71918 Inf 0.4128 1.2382
    109 1.8847 Inf 0.08987 Inf 0.3962 0.4291
    110 1.1381 Inf 0.65670 Inf 0.3503 1.1757
```

[3] You can implement penalties by hand.

Calipers

The optmatch package allows calipers (which forbid certain pairs from being matched).[4] Here, for example, we forbid comparisons which differ by more than 2 standard deviations on the propensity score. (Notice that we also use the `propensity.model` option to `summary` here to get a quick look at the balance test:)

```
Structure of matched sets:
5:1 3:1 2:1 1:1 1:2 1:3 1:4 0:1
  1   1   1   8   2   1   1   1
Effective Sample Size:  18.3
(equivalent number of matched pairs).

Balance test overall result:
  chisquare df p.value
       1.25  2   0.536
Structure of matched sets:
1:1 0:1
 22   1
Effective Sample Size:  22
(equivalent number of matched pairs).

Balance test overall result:
  chisquare df p.value
       12.2  2  0.0022
```

[4] You can implement penalties by hand.

Calipers

Another example: We may want to match on mahalanobis distance but disallow any pairs with extreme propensity distance and/or extreme differences in baseline homicide rates (here using many covariates all together).

```
          control
treatment    401    402    403
      101 0.4147 0.3854 0.9707
      102 0.3175 0.4826 0.8735
      103 0.3601 1.1602 0.1960
        0%        10%        20%        30%        40%        50%        60%        70%        80
 0.0007996  0.1008848  0.2147839  0.3451456  0.4711063  0.6495775  0.8518273  1.1360636  1.511547
     0%     10%     20%     30%     40%     50%     60%     70%     80%     90%    100%
0.07624 0.67875 1.18235 1.44850 1.71964 1.93039 2.14689 2.33615 2.52358 2.81432 4.05058
       control
treated    405   407    408   409    410    411
    105 0.3911 5.579 1.4037 4.397 1.0973 1.9227
    106 1.6671 4.303 0.1278 3.121 0.1786 0.6468
    107 0.1871 5.783 1.6077 4.601 1.3014 2.1267
    108 1.0756 4.895    Inf 3.712 0.4128    Inf
    109    Inf 4.086    Inf 2.903 0.3962    Inf
    110 1.1381 4.832 0.6567 3.650 0.3503 1.1757
         control
treatment    405   407     408   409   410    411
      105 0.5009 3.313 2.63301 2.685 1.017 1.3631
      106 0.9068 3.047 3.24223 2.355 1.102 0.8459
      107 0.1105 3.515 3.08576 2.851 1.252 1.3904
      108 2.4370 2.782 0.76237 2.488 1.674 2.2701
      109 3.2021 2.538 0.07624 2.459 2.217 2.7334
      110 2.5727 2.720 0.60990 2.462 1.755 2.3388
```

Calipers

Now, use this new matrix for the creation of stratified designs — but possibly excluding some units (also showing here the `tol` argument. The version with the tighter tolerance produces a solution with smaller overall distances)

```
Structure of matched sets:
1:0 5:1 2:1 1:1 1:2 1:4 1:6
  2   1   3   5   2   1   1
Effective Sample Size:  16.6
(equivalent number of matched pairs).
Structure of matched sets:
1:0 5:1 2:1 1:1 1:2 1:4 1:6
  2   1   3   5   2   1   1
Effective Sample Size:  16.6
(equivalent number of matched pairs).

Balance test overall result:
  chisquare df p.value
       3.29  2   0.193
[1] 0.344
[1] 0.344
```

Exact Matching

We often have covariates that are categorical/nominal and for which we really care about strong balance. One approach to solve this problem is match **exactly** on one or more of such covariates. If `fullmatch` or `match_on` is going slow, this is also an approach to speed things up.

```
Structure of matched sets:
9:1 4:1 1:1 1:2 1:5 1:7
  1   1   5   2   1   1
Effective Sample Size:  14.5
(equivalent number of matched pairs).

Balance test overall result:
  chisquare df p.value
       4.52  2   0.104
 Group                                                Members
  hi.1 101, 103, 104, 105, 107, 201, 207, 209, 210, 404
 hi.10           203, 407, 409, 411, 414, 801, 802, 803
 hi.11                                          204, 813
 hi.12                                          205, 402
 hi.14                                          208, 412
 hi.17                                          211, 415
  hi.5                                    106, 401, 410
  hi.7                                          108, 810
  hi.9                    202, 403, 408, 413, 807, 808
  lo.1                         102, 109, 111, 112, 812
  lo.3                                    110, 405, 811
```

Exact Matching

	Class	hi		lo	
	Trt	0	1	0	1
fmEx1					
hi.1		1	9	0	0
hi.10		7	1	0	0
hi.11		1	1	0	0
hi.12		1	1	0	0
hi.14		1	1	0	0
hi.17		1	1	0	0
hi.5		2	1	0	0
hi.7		1	1	0	0
hi.9		5	1	0	0
lo.1		0	0	1	4
lo.3		0	0	2	1

1. Overview and Review
2. How to assess the randomization process in an experiment (to teach us how to assess research designs in observational studies).
3. Assessing comparisons in observational studies
4. Matching on Many Covariates: Using the Mahalnobis Distance to Scale Euclidean Distance
5. Matching on Many Covariates: Using Propensity Scores
6. Matching Tricks of the Trade: Calipers, Exact Matching
7. The separation problem in Logistic Regression

What about using many covariates? The separation problem in logistic regression

What if we want to match on more than two covariates? Let's step through the following to discover a problem with logistic regression when the number of covariates is large relative to the size of the dataset.

```
library(splines)
library(arm)
thecovs <- unique(c(names(meddat)[c(5:7, 9:24)], "HomRate03"))
balfmla <- reformulate(thecovs, response = "nhTrt")
psfmla <- update(balfmla, . ~ . + ns(HomRate03, 2) + ns(nhPopD, 2) + ns(nhHS, 2))
glm0 <- glm(balfmla, data = meddat, family = binomial(link = "logit"))
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
glm1 <- glm(psfmla, data = meddat, family = binomial(link = "logit"))
```

```
Warning: glm.fit: algorithm did not converge

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

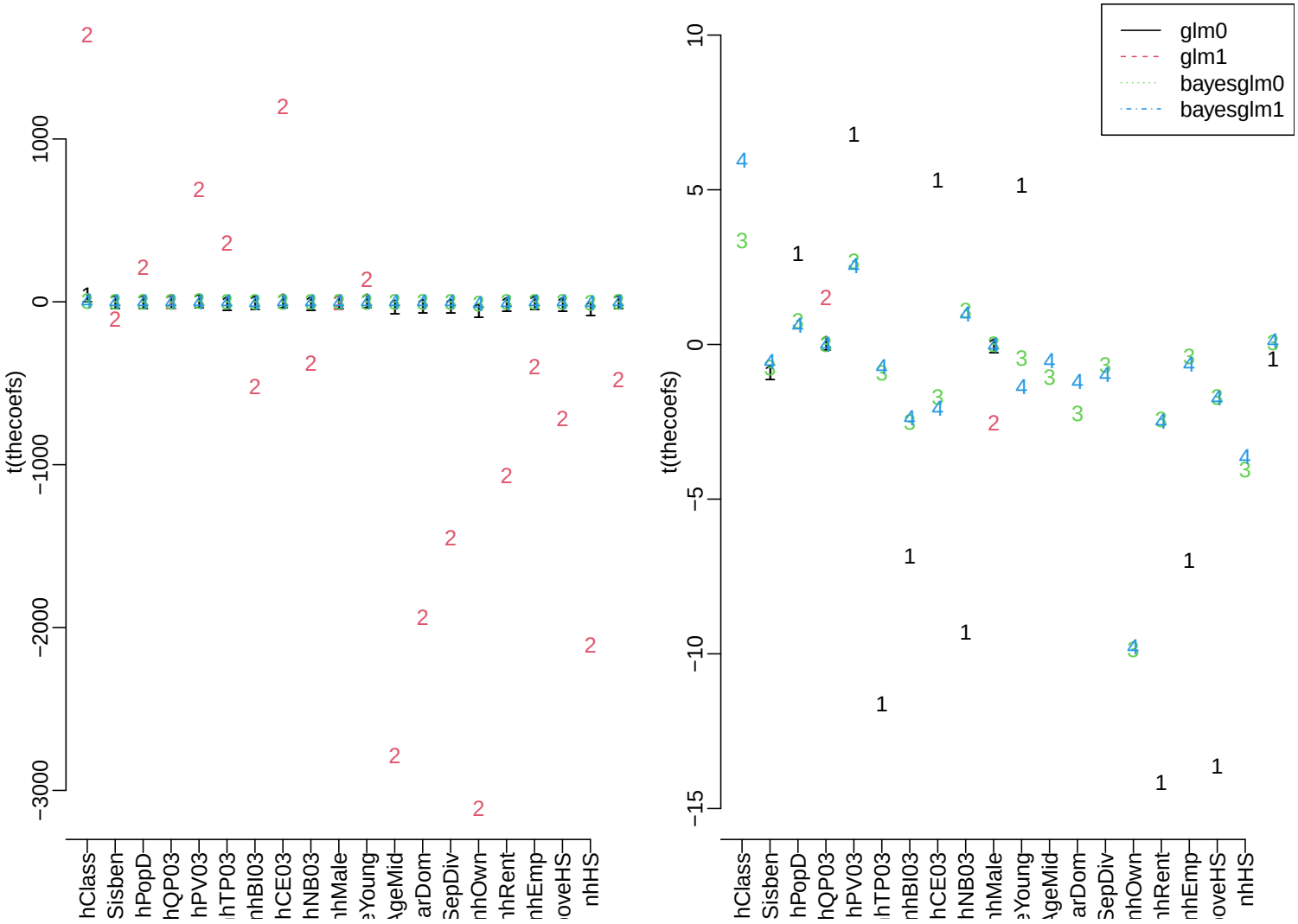
```
bayesglm0 <- bayesglm(balfmla, data = meddat, family = binomial(link = "logit"))
bayesglm1 <- bayesglm(psfmla, data = meddat, family = binomial(link = "logit"))
psg1 <- predict(glm1, type = "response")
psg0 <- predict(glm0, type = "response")
psb1 <- predict(bayesglm1, type = "response")
psb0 <- predict(bayesglm0, type = "response")
```

The separation problem

Logistic regression is excellent at discriminating between groups ...often **too excellent** for us (Gelman et al., 2008). First evidence of this is big and/or missing coefficients in the propensity score model. See the coefficients below (recall that we are predicting `nhTrt` with these covariates in those models):

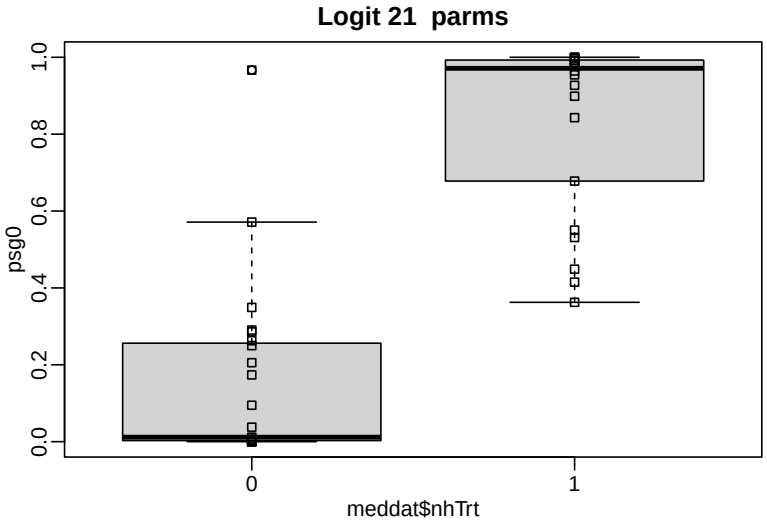
	(Intercept)	nhClass	nhSisben	nhPopD	nhQP03
glm0	45.497	-0.9139	2.9400	0.02975	6.812
glm1	1638.021	-104.6673	213.5140	1.51385	691.711
bayesglm0	3.347	-0.7622	0.7501	0.01002	2.687
bayesglm1	5.959	-0.5386	0.6139	0.00404	2.547

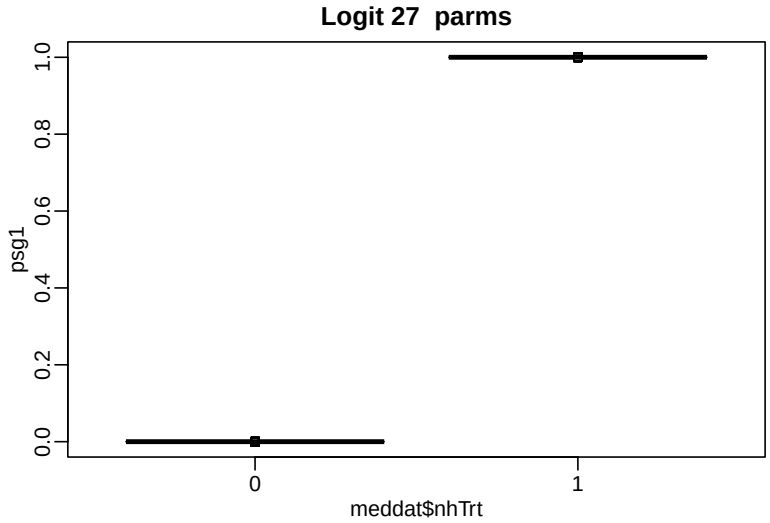
The separation problem

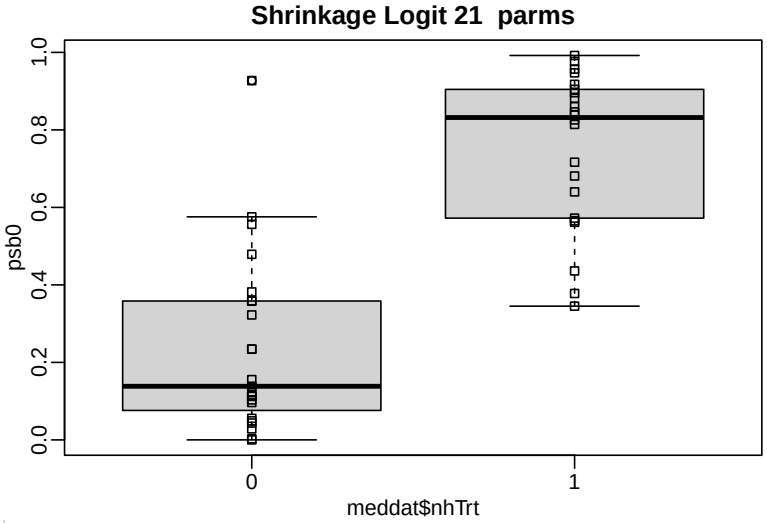


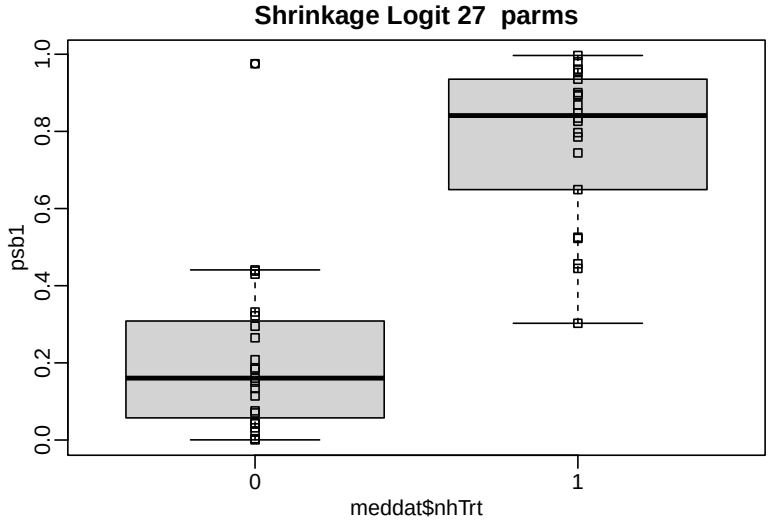
The separation problem in logistic regression

So, if we are interested in using the propensity score to compare observations in regards the multi-dimensional space of many covariates, we would probably prefer a dimensional reduction model like `bayesglm` over `glm`.









Back to

Decision Points in Creating Matched Designs

- Which covariates and their scaling and coding. (For example, exclude covariates with no variation!)
- Which distance matrices (scalar distances for one or two important variables, Mahalanobis distances (rank transformed or not), Propensity distances (using linear predictors)).
- (Possibly) which calipers (and how many, if any, observations to drop. Note about ATT as a random quantity and ATE/ACE as fixed.)
- (Possibly) which exact matching or strata
- (Possibly) which structure of sets (how many treated per control, how many controls per treated)
- Which remaining differences are tolerable from a substantive perspective?
- How well does the resulting research design compare to an equivalent block-randomized study (`xBalance`)?
- (Possibly) How much statistical power does this design provide for the quantity of interest?
- Other questions to ask about a research design aiming to help clarify comparisons.

Next time:

- Matching when we have more than one group (non-bipartite matching)

Remaining questions?

References

- Arceneaux, Kevin (Sept. 2005). "Using cluster randomized field experiments to study voting behavior". In: Annals of the Americal Academy of Political and Social Science 601, pp. 169–179.
- Gelman, Andrew et al. (2008). "A weakly informative default prior distribution for logistic and other regression models". In: The Annals of Applied Statistics 2.4, pp. 1360–1383.
- Hansen, Ben B and Jake Bowers (2008). "Covariate Balance in Simple, Stratified and Clustered Comparative Studies". In: Statistical Science 23.2, pp. 219–236.
- Mahalanobis, Prasanta Chandra (1930). "On test and measures of group divergence, Part I: Theoretical formulae". In.
- Rosenbaum, Paul R (2010). Design of Observational Studies. New York, NY: Springer.
- Rosenbaum, Paul R. and Donald B. Rubin (1983). "The Central Role of the Propensity Score in Observational Studies for Causal Effects". In: Biometrika 70.1, pp. 41–55.