Matching for Adjustment and Causal Inference Class 4: Matching with more than two groups — Non-bipartite Matching

Jake Bowers

August 03, 2023

1. Overview and Review
2. Non-bipartite Matching: The Medellin Data
3. Non-bipartite Matching: An Application with the Study of Race and Place

So far: A workflow to create research designs from data I

The data exist, maybe data collected for purposes other than assessing theories. How do we protect ourselves from criticisms that we (a) did 100 hypothesis tests and chose the design/regression specification that suits us? and (b) our results describe our own preferences more than they describe the world?

- **Before looking at outcomes: List the main alternative explanations** (could crime have caused Metrocable stations; socio-economic status differences; ...). Can we operationalize these explanations?
- **Before looking at outcomes** we explain our designs to ourselves by comparing the design to our background subtantive understanding of the context for causality. (What are the drivers of the "treatment"? How **much** adjustment in substantive terms is required? What are the most compelling alternative explanations for the treatment→outcome relationship? (Alternative to the theoretical explanation that we are exploring/assessing))

So far: A workflow to create research designs from data II

- **Before looking at outcomes** we explain our designs to ourselves by comparing the design to an equivalently designed randomized experiment using the known distribution of the d^2 statistic under the null hypothesis of no covariate-to-treatment relationships across any covariates (see the Hansen and Bowers 2008 piece).
- We estimate (average) effects and test hypotheses about effects **as if the research design was randomized**.
- **After estimating effects/testing hypotheses** we again engage with alternative explanations by modeling how unobserved covariates might confound the relationship (Sensivity Analysis).

Notice: We could pre-register our design process, or even our designs themselves.

How might we do this? Adjustment by stratification

- Matching to generate optimal stratifications (decisions and strategies that are part of research design; matching on missingness and `fill.NAs`; `exactMatch`; `caliper`; `min.controls`; `effectiveSampleSize`); Or using `designmatch` or one of the other packages recommended by Rosenbaum 2020. The point is to create a categorical variable indicating set membership — the point of the design phase is **not** to produce estimated effects or p-values or confidence intervals.
- Assessing success of stratified research designs in adjustment;
- The As-If-Randomized mode of statistical inference for stratified research designs (treat a matched design as a block-randomized experiment).

Today: Stratification with more than two groups — Non-bipartite matching

1. What is the general idea of creating pairs or sets that differ on one key explanatory variable (or causal factor) but do not differ on others.
2. How do we assess this kind of stratification? What do balance tests mean? (How to interpret the output of `xBalance` (or `balanceTest`) in this case?)
3. What do "effects" mean in this case? How to estimate them?
4. We will use two applications to address these questions. See also the (Rabb et al., 2022) piece as another application.

1. Overview and Review
2. Non-bipartite Matching: The Medellin Data
3. Non-bipartite Matching: An Application with the Study of Race and Place

Non-bipartite Matching: The Medellin Data

Hypothetical Setup I

Imagine that there is a debate about whether housing insecurity is strongly related to violence. We have neighborhoods in Medellin where we have measured both violence scaled by the population of the place (`HomRate08`), whether people own their own home (`nhOwn`), and potential confounders like the proportion of people who are employed (`nhEmp`). However, we know that both housing insecurity as well as violence can be predicted from other background variables: maybe the relationships we would summarize between housing and violence would be confounded by those other relationships.

Designmatch setup I

We will use an approach to adjustment called **non-bipartite** matching) which doesn't require two groups. Rather it creates pairs of units (neighborhoods) in this case, which are as similar as possible in regards to many covariates.

```
covs <- c(
  "nhClass", "nhSisben", "nhPopD", "nhQP03", "nhPV03", "nhTP03",
  "nhBI03", "nhCE03", "nhNB03", "nhMale", "nhAgeYoung",
  "nhAgeMid", "nhMarDom", "nhSepDiv", "nhAboveHS", "nhHS", "HomRate03"
)

covmat <- dplyr::select(meddat, one_of(covs))

## Mahalanobis distances for each neighborhood
meddat$covmh <- mahalanobis(
  x = covmat,
  center = slam::col_means(covmat),
  cov = cov(covmat)
)

## Absolute mahalanobis distances between neighborhoods
mhdist_mat <- outer(meddat$covmh, meddat$covmh, FUN = function(x, y) {
  abs(x - y)
})
dimnames(mhdist_mat) <- list(meddat$nh, meddat$nh)
```

Designmatch use I

Now, we can match on those distances:

Designmatch use II

```
## Turns out that the designmatch software doesn't like too many decimals, and prefers
## mean-centered distances. This doesn't really matter in substantive terms but is important in
## regards to getting the software to work
matchdist_mat <- round(100 * mhdist_mat / mean(mhdist_mat), 1)

## Restrict allowable matches. This is like a caliper but on two dimensions.
nearlist <- list(
  covs = as.matrix(meddat[, c("HomRate03", "nhAboveHS")]),
  pairs = c(HomRate03 = 5, nhAboveHS = .5)
)

## For larger problems you will want to install gurobi using an academic
## license. After installing the license, then I do something like the following
## where the details of the version numbers will differ
## install.packages("/Library/gurobi952/macos_universal2/R/gurobi_9.5-2_R_4.2.0.tgz",repos=NULL)
## also had to use a different version of designmatch for now:

## Only run this next one one time
### renv::install("bowers-illinois-edu/designmatch")
library(designmatch)
# library(slam)
library(highs)
# library(gurobi)
solverlist <- list(name = "highs", approximate = 0, t_max = 1000, trace = 1)
```

Designmatch use III

The function `nmatch` does the optimization. It is not full-matching, but is pair-matching.

```
mh_pairs <- nmatch(
  dist_mat = matchdist_mat,
  near = nearlist,
  subset_weight = 1,
  solver = solverlist
)
```

```
  Building the matching problem...
  HiGHS optimizer is open...
  Finding the optimal matches...
  Optimal matches found
```

```
## mh_pairs
```

Designmatch use IV

```
#' Function to convert the output of nmatch into a factor variable for use in analysis
nmatch_to_df <- function(obj, origid) {
  ## We want a factor that we can merge onto our
  ## existing dataset. Here returning a data.frame so that
  ## we can merge --- seems less error prone than using
  ## rownames even if it is slower.
  matchesdat <- data.frame(
    bm = obj$group_id,
    match_id = c(obj$id_1, obj$id_2)
  )
  matchesdat$id <- origid[matchesdat$match_id]
  return(matchesdat)
}
```

```
mh_pairs_df <- nmatch_to_df(mh_pairs, origid = meddat$nh)
nrow(mh_pairs_df)
```

```
[1] 14
```

```
## So, in matched set 1 (bm==1) we see two neighborhoods:
mh_pairs_df %>% filter(bm == 1)
```

```
  bm match_id  id
1  1        3 103
2  1       39 803
```

Designmatch use V

```
mh_pairs_df$nh <- mh_pairs_df$id

# The nmatch_to_df function creates a column labeled "bm" which contains
meddat2 <- inner_join(meddat, mh_pairs_df, by = "nh")
meddat2 <- droplevels(meddat2)
stopifnot(nrow(meddat2) == nrow(mh_pairs_df))

## Number of matches:
# meddat2$bm is the matched set indicator.
stopifnot(length(unique(meddat2$bm)) == nrow(meddat2) / 2)
nrow(mh_pairs_df)
```

```
[1] 14
```

```
nrow(meddat2)
```

```
[1] 14
```

```
## Notice some observations were not matched:
nrow(meddat)
```

```
[1] 45
```

Assessing the design I

Now, what we are trying to do is break the relationship between covariates and the main explanatory variables (just as we might in a pair randomized study): the neighborhood higher on the explanatory variable shouldn't be systematically more or less likely to be the neighborhood higher on any given covariate in such a study. We assess this below:

```
## Make a new variable that is 1 for the neighborhood higher in home ownership
## and 0 for the neighborhood who is lower. (Similarly for Employment)
## We'd like to show that the covariates are not related to either home
## ownership or employment within pair.
meddat2 <- meddat2 %>%
  group_by(bm) %>%
  mutate(
    rank_own = rank(nhOwn) - 1,
    rank_emp = rank(nhEmp) - 1
  ) %>%
  arrange(bm) %>%
  ungroup()

## Notice pair bm=1
meddat2 %>% dplyr::select(bm, nh, nhOwn, rank_own, nhEmp, rank_emp)
```

Assessing the design II

```
# A tibble: 14 x 6
      bm    nh nhOwn rank_own nhEmp rank_emp
   <int> <int> <dbl>    <dbl> <dbl>    <dbl>
 1     1   103 0.5          0 0.333        1
 2     1   803 0.667        1 0.167        0
 3     2   105 0.542        0 0.5          1
 4     2   402 0.625        1 0.25         0
 5     3   107 0.857        1 0.286        0
 6     3   807 0.7          0 0.35         1
 7     4   108 0.727        1 0.455        1
 8     4   801 0.636        0 0.364        0
 9     5   201 0.455        0 0.364        1
10     5   401 0.571        1 0.286        0
11     6   202 0.5          0 0.25         0
12     6   405 0.733        1 0.267        1
13     7   207 0.556        1 0.222        0
14     7   415 0.5          0 0.25         1
```

```
## Check for sets with a tie
table(meddat2$rank_own)
```

```
0 1
7 7
```

Assessing the design III

```
## Since balanceTest demands binary treatment, we remove ties for now.
meddat3 <- meddat2 %>% filter(rank_own != .5)
table(meddat3$rank_own)
```

```
0 1
7 7
```

```
## We are trying to break the relationships between the covariates and the two
## explanatories. Let's look at one of them here.

## Since we have a smaller dataset, we need to use fewer covariates if we want to use the large sa
newcovs <- c("nhClass", "HomRate03", "nhTP03", "nhAgeYoung", "nhAboveHS", "nhEmp")

balfmla_new <- reformulate(newcovs, response = "rank_own")
## Using only the matched data and also conditional within sets
xb_own <- balanceTest(update(balfmla_new, . ~ . + strata(bm)), data = meddat3, p.adjust = "none")
xb_own$overall
```

```
   chisquare df p.value
bm     6.658  6  0.3537
--     5.004  6  0.5433
```

Assessing the design IV

```
xb_own_vars <- data.frame(xb_own$results[, c("Control", "Treatment", "adj.diff", "std.diff", "p")
## xb_own_vars$padj <- p.adjust(xb_own_vars$p, method = "holm") ## already adjusted using holm adj
options(digits = 3)
arrange(xb_own_vars, p) %>% zapsmall(digits = 5)
```

```
           Control Treatment adj.diff std.diff     p
nhEmp        0.344     0.276  -0.0684   -0.788 0.148
nhAgeYoung   0.288     0.378   0.0907    0.822 0.187
nhAboveHS    0.130     0.064  -0.0662   -0.592 0.320
HomRate03    1.005     1.404   0.3995    0.441 0.400
nhTP03       0.525     0.584   0.0591    0.281 0.655
nhClass      2.286     2.286   0.0000    0.000 1.000
```

```
stopifnot(xb_own$overall[, "p.value"] > .3)
```

An equivalent way to do what balanceTest is doing

```
library(formula.tools)
library(coin)
coin_fmla <- ~ rank_own | bmF
lhs(coin_fmla) <- rhs(balfmla_new)
meddat3$bmF <- factor(meddat3$bm)
coin_test <- independence_test(coin_fmla, data = meddat3, teststat = "quadratic")
coin_test_perm <- independence_test(coin_fmla, data = meddat3, teststat = "quadratic", distributi
```

Outcome Analysis

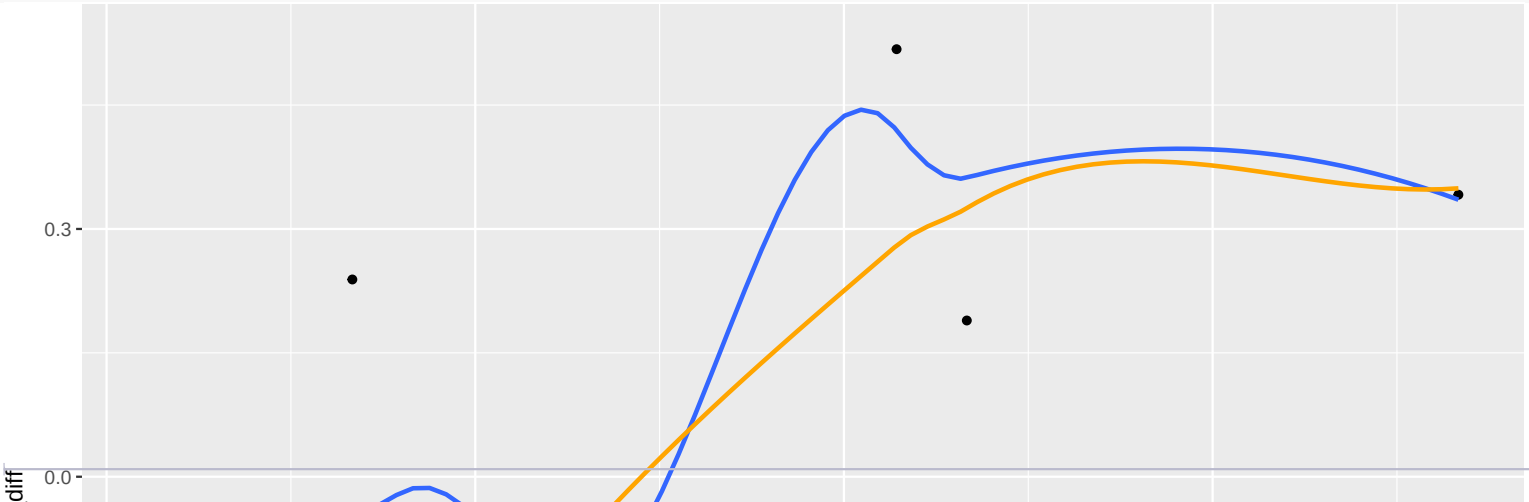
Now, assuming we are happy with the design, we describe the relationships between home ownership and violence in 2008 at the neighborhood level.

```
## Ways to assess the relationship between home ownership and the outcome
## conditional on sets. These are all the same.

## We will start with estimating the difference between the high and low home
## ownership neighborhoods and then move to estimating the smooth linear
## relationship between differences in proportion home ownership and the
## outcome.

## First, the most transparent way, but most typing is to convert the data
## into the strata level and create averages.
meddat2$bmF <- factor(meddat2$bm)
pair_diffs <- meddat2 %>%
  filter(rank_own != .5) %>%
  group_by(bmF) %>%
  summarize(
    hr = mean(HomRate08),
    hr_diff = HomRate08[rank_own == 1] - HomRate08[rank_own == 0],
    own_diff = nhOwn[rank_own == 1] - nhOwn[rank_own == 0],
    own_diff_raw = diff(nhOwn),
    hr_diff_raw = diff(HomRate08), .groups = "drop"
  )

## Simply the mean of the differences within pair between the higher and lower
## home ownership neighborhoods. We will see that this is exactly the same as
## the other estimates.
est1 <- mean(pair_diffs$hr_diff)
```

Graphing the possibly non-linear/heterogeneous relationships

This next allows us to explore the within pair differences — here we look at how differences in proportion home ownership within pair relate to differences in homocide rate within pair.

```
## More exploring about the pair-differences
g1 <- ggplot(data = pair_diffs, aes(x = own_diff, y = hr_diff)) +
  geom_point() +
  geom_smooth(method = "loess", se = FALSE, method.args = list(family = "gaussian", deg = 2, span
  geom_smooth(
    method = "loess", se =
      FALSE, method.args = list(family = "symmetric", span = .8, deg = 1), col = "orange"
  )

g1
```



Outcome analysis 2: Size of the difference within pairs

So far our analysis asked, "Did the neighborhood in the pair with higher home ownership have less or more violence, on average, than the neighborhood in the pair with less home ownership." This ignores the size of the difference in proportion owning a home and in exchange allows us to simplify the question. That said, we can also look how the mean neighborhood violence differs given different magnitude of differences within pair. What about when we are looking at the difference in violence associated linearly with continuous differences in home ownership? (i.e. looking at how differences in violence are associated with differences in home ownership in proportions). Notice below that we have the same methods as above (only that the `difference_in_means` doesn't work because we don't have a binary explanatory variable.)

Outcome analysis 2: Size of the difference within pairs

In each case the interpretation is about average differences in outcome for a one unit difference in the explanatory variable (which is really large, it is the maximum difference between any two neighborhoods on the explanatory.)

```
## Still restricting attention to pairs that are not identical so that we can be
## using the same observations for both analyses.
est1cont <- lm_robust(hr_diff ~ own_diff - 1, data = pair_diffs)

est3cont <- lm_robust(HomRate08 ~ nhOwn, fixed_effects = ~bmF, data = meddat2, subset = rank_own
est4cont <- lm_robust(HomRate08 ~ nhOwn + bmF, data = meddat2, subset = rank_own != .5)

meddat2 <- meddat2 %>%
  group_by(bmF) %>%
  mutate(own_md = nhOwn - mean(nhOwn)) %>%
  ungroup()
est5cont <- lm_robust(hr_md ~ own_md, data = meddat2, subset = rank_own != .5)

meddat2 %>%
  filter(bmF == "1") %>%
  dplyr::select(nhOwn, rank_own, own_md, HomRate08, hr_md) %>%
  head()
```

```
# A tibble: 2 x 5
  nhOwn rank_own  own_md HomRate08   hr_md
  <dbl>    <dbl>   <dbl>     <dbl>   <dbl>
1 0.5          0 -0.0833     0.308 -0.0946
2 0.667        1  0.0833     0.497  0.0946
```

```
pair_diffs %>% filter(bmF == "1")
```

Summary of non-bipartite matching

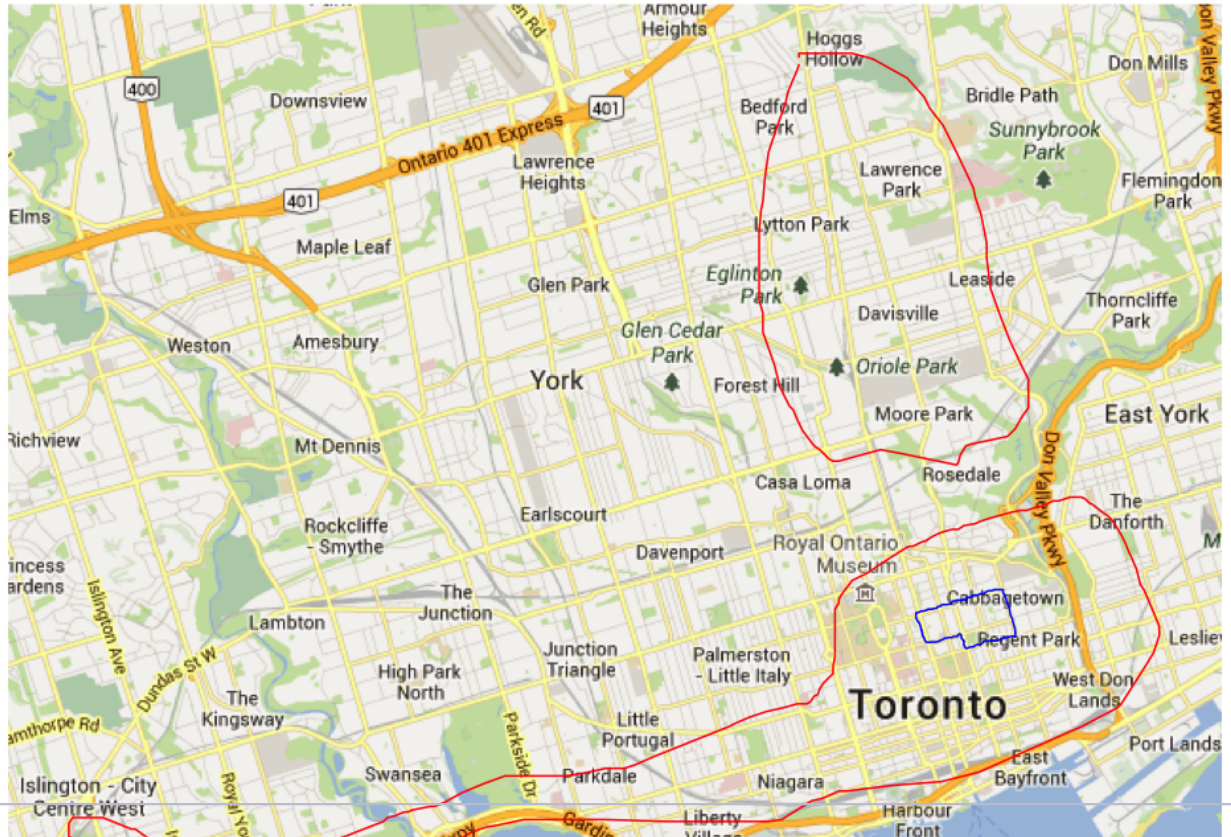
- We can make pairs of units within which we can claim to have broken the relationship between many background covariates and another causal driver, intervention, or treatment even if that Z variable has many values. This is called **non-bipartite matching**.
- We can compare these relationships to (1) our substantive and contextual knowledge and (2) the kind of $X \rightarrow Z$ relationships we would see had Z been randomly assigned within pair (imagine Z having multiple values and the higher value being assigned at random within pair).
- We can compare how $Z \rightarrow Y$ conditional on pair in a variety of ways: estimation and testing comparing the higher-vs-lower treatment value member of a pair or by averaging over the size of the higher-vs-lower treatment value differences (say, using OLS to focus on the linear relationship). We can also visualize the relationships to assess linearity and/or learn more.

1. Overview and Review
2. Non-bipartite Matching: The Medellin Data
3. Non-bipartite Matching: An Application with the Study of Race and Place

How do perceptions of place influence attitudes?

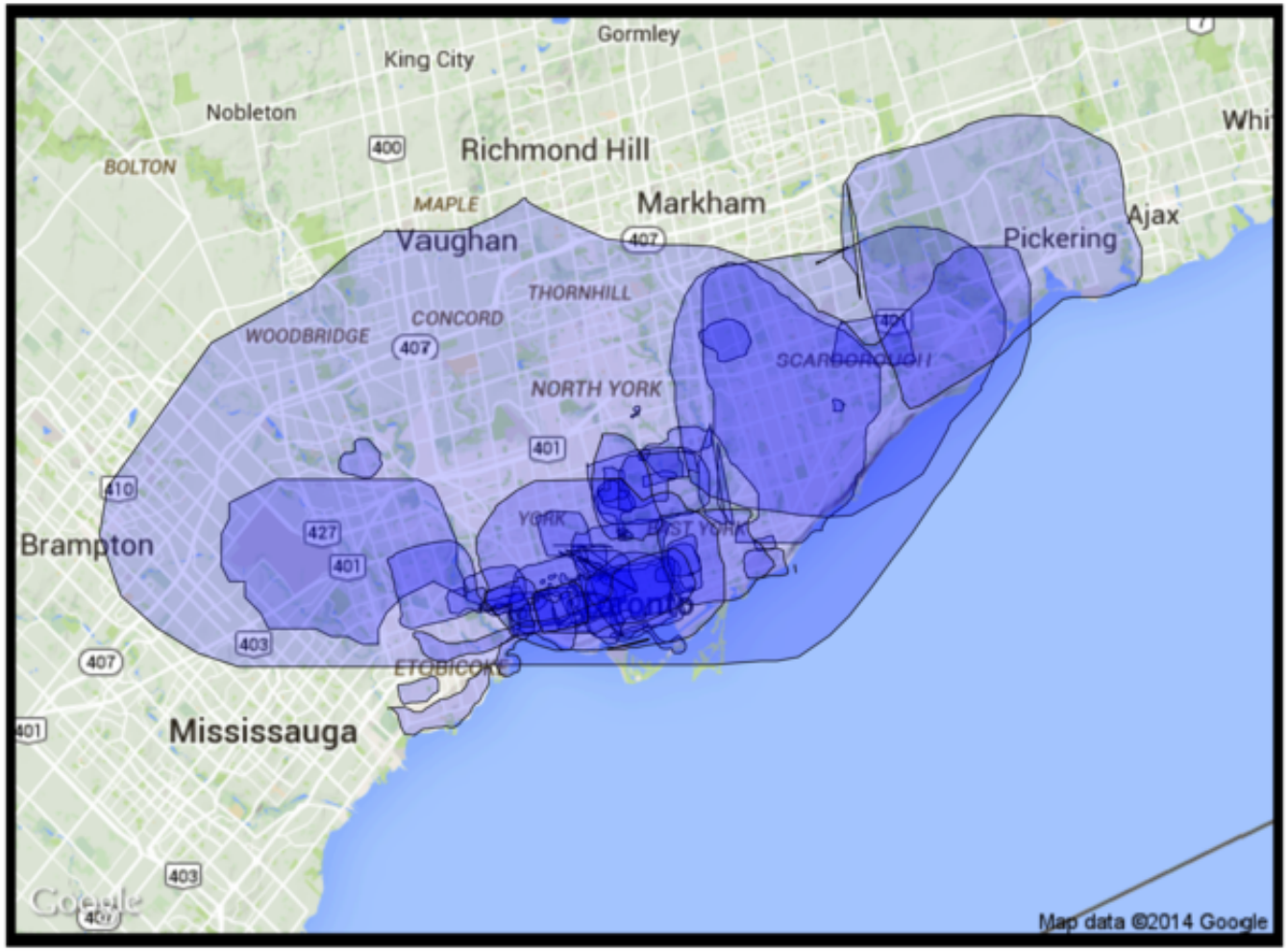
Wong et al. (2012) set out to measure perceptions of environments using an internet survey of Canadians during 2012 where each respondent drew a map of their "local community" and then reported their understanding of the demographic breakdown of this place.

```
## White English Speaking Canadians only
load(url("http://jakebowers.org/ICPSR/canadamapdat.rda"))
## summary(canadamapdat)
```



Capturing perceptions

Here are 50 maps drawn by people based in Toronto.



Capturing perceptions

And here is the question people were asked (groups in random order).

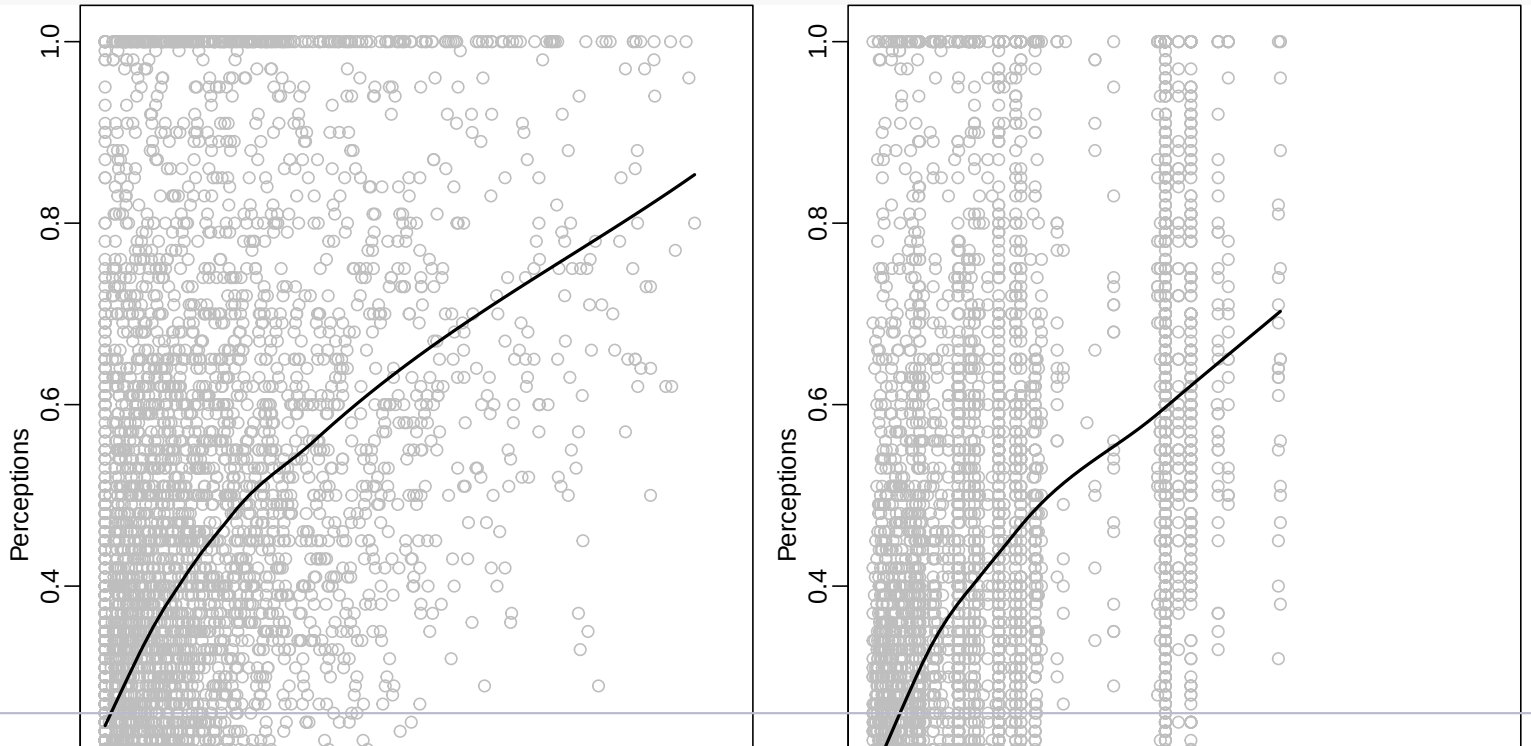
Just your best guess – what percentage of the population of this community is:

Chinese	0%	0	100%
Unemployed	0%	0	100%
Conservative Party supporters	0%	0	100%
Canadian Aboriginals	0%		100%

Capturing perceptions

White, Engish-speaking, Canadian respondents' reports about "visible minorities" in their hand drawn "local communities".

```
par(mfrow = c(1, 2))
with(canadamapdat, scatter.smooth(vm.da, vm.community.norm2,
  col = "gray", ylab = "Perceptions", xlab = "Census Neighborhood (DA)",
  xlim = c(0, 1), ylim = c(0, 1), lpars = list(lwd = 2)
))
with(canadamapdat, scatter.smooth(vm.csd, vm.community.norm2,
  col = "gray", ylab = "Perceptions", xlab = "Census Municipality (CSD)",
  xlim = c(0, 1), ylim = c(0, 1), lpars = list(lwd = 2)
))
```



Codebook: Mainly for Rmd file

The variables are: age in years, income as a scale, sex in categories, a social.capital scale coded to run 0 to 1, country of ancestry in categories, csd.pop is population of the Census Subdivision (like a municipality), vm.csd is 2006 proportion visible minority in the CSD, vm.da is proportion visible minority in the Census Dissemination Area (a small area containing 400–700 persons), and vm.community.norm2 is the proportion of visible minorities reported by respondents in their map of their local community, community_area_km is the area within their drawing in square km.

How to make the case for perceptions?

If we could randomly assign different perceptions to people, we could claim that differences of perceptions matter (above and beyond and independent of objective characteristics of the context).

What is an observational design that would do this? Match people on objective context (and maybe covariates) who differ in perceptions.

But objective context is continuous not binary: rather than matching m "treated" to $n - m$ "controls", we want to compare all n with all n respondents.

```
## Exclude people who did not offer a perception or an outcome
wrkdat0 <- canadamapdat[!is.na(canadamapdat$vm.community.norm2) &
  !is.na(canadamapdat$social.capital01), ]
## Take a random sample so that the lecture compiles
set.seed(12345)
wrkdat <- droplevels(sample_n(wrkdat0, 500))
wrkdat$vmdaPct <- wrkdat$vm.da * 100 ## express in pct
```

Create $n \times n$ distance matrices

Our main design compares white, English-speaking, Canadians with similar neighborhood proportions of visible minorities (as measured by the Canadian Census in 2006).

```
scalar.dist <- function(v) {
  ## Utility function to make n x n abs dist matrices
  outer(v, v, FUN = function(x, y) {
    abs(x - y)
  })
}

vmdaDist <- round(scalar.dist(wrkdat$vmdaPct), 1)
dimnames(vmdaDist) <- list(row.names(wrkdat), row.names(wrkdat))
## The nbpmatching way (Mahalanobis \equiv standardized in one dimension) takes a while:
## obj.com.dist.mat2<-distancematrix(gendistance(wrkdat[,"vmdaPct",drop=FALSE]))
## compare to tmp<-scalar.dist(wrkdat$vmdaPct/sd(wrkdat$vmdaPct))
wrkdat$vmdaPct[1:4]
```

```
[1]  0.00 23.53 17.80  1.63
```

```
diff(wrkdat$vmdaPct[1:4])
```

```
[1]  23.53  -5.73 -16.17
```

```
vmdaDist[1:4, 1:4]
```

```
     1    2    3    4
1  0.0 23.5 17.8  1.6
2 23.5  0.0  5.7 21.9
3 17.8  5.7  0.0 16.2
4  1.6 21.9 16.2  0.0
```

Non-bipartite match

```
canada_nearlist <- list(
  covs = as.matrix(wrkdat[, c("csd.pop", "community_area_km")]),
  pairs = c(csd.pop = 100000, community_area_km = 5)
)

## Try not to match two people with the same perceptions --- that doesn't add anything to our anal
canada_farlist <- list(
  covs = as.matrix(wrkdat[, "vm.community.norm2"]),
  pairs = c(vm.community.norm2 = .1)
)

canada_pairs <- nmatch(
  dist_mat = vmdaDist,
  near = canada_nearlist,
  far = canada_farlist,
  subset_weight = 1,
  solver = solverlist
)
```

```
  Building the matching problem...
  HiGHS optimizer is open...
  Finding the optimal matches...
  Optimal matches found
```

```
## Version using nonbimatch
## vmdaDistMat <- distancematrix(vmdaDist)
## nbp1match<-nonbimatch(vmdaDistMat)
## nbp1<-get.sets(nbp1match$matches,remove.unpaired=TRUE)
wrkdat$id <- row.names(wrkdat)
canada_pairs_df <- nmatch_to_df(canada_pairs, origid = wrkdat$id)
```

Inspect the solution

```
wrkdat2[order(wrkdat2$nbp1), c("nbp1", "vmdaPct", "vm.community.norm2")][1:6, ]
```

```
    nbp1 vmdaPct vm.community.norm2
1      1    17.8               0.25
164    1    18.2               0.60
2      2     0.0               0.40
132    2     0.0               0.63
3      3    16.8               0.45
71     3    16.7               0.61
```

```
## table(wrkdat2$nbp1)
nbp1vmdiffs <- tapply(wrkdat2$vmdaPct, wrkdat2$nbp1, function(x) {
  abs(diff(x))
})
nbp1percdiffs <- tapply(wrkdat2$vm.community.norm2, wrkdat2$nbp1, function(x) {
  abs(diff(x))
})
summary(nbp1vmdiffs)
```

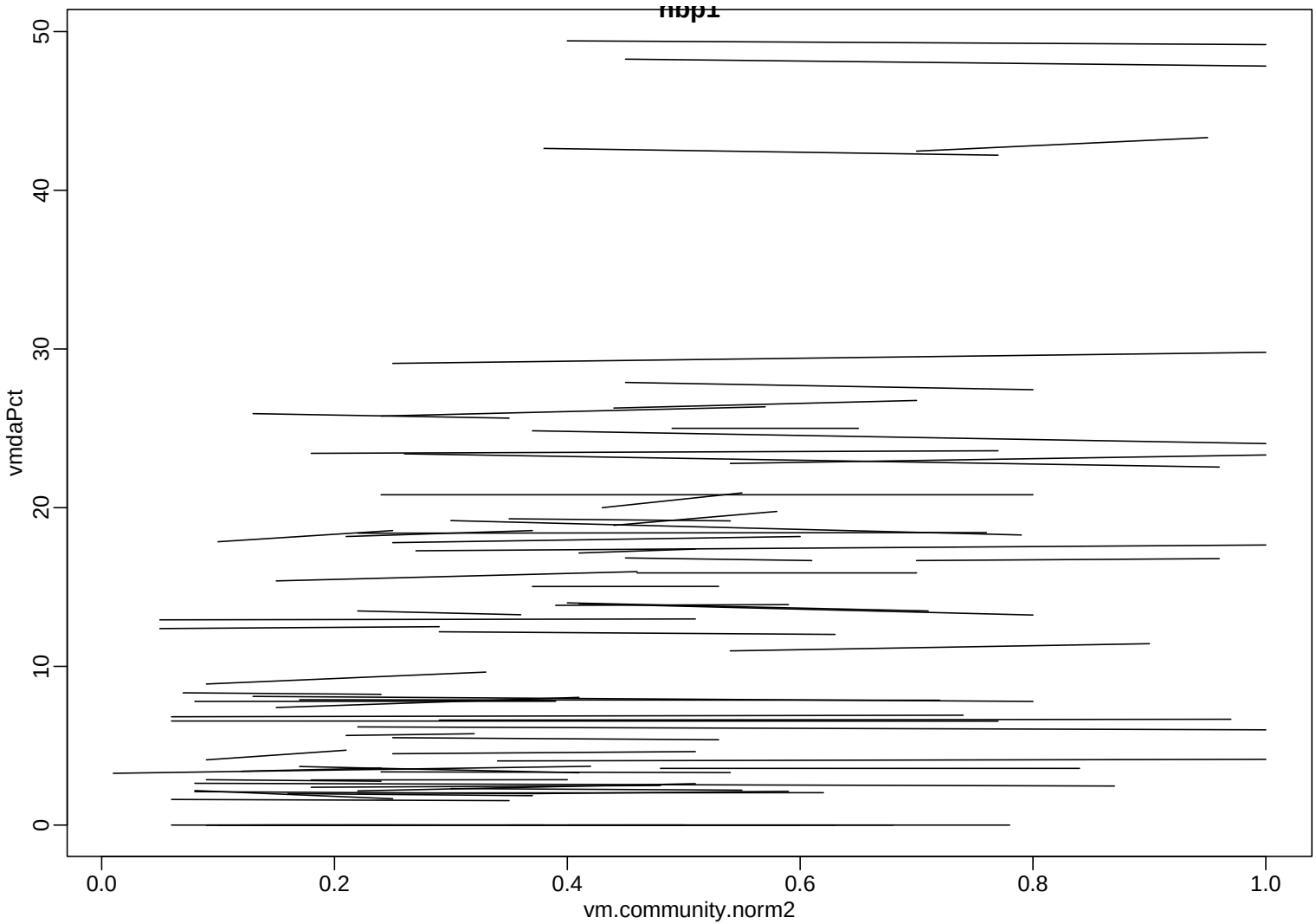
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.023   0.133   0.259   0.450   0.930
```

```
summary(nbp1percdiffs)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.100   0.193   0.290   0.339   0.440   0.790
```

```
source(url("http://jakebowers.org/Matching/nonbimatchingfunctions.R"))
```

Inspect the solution



Assess balance

No treatment and control groups to compare. But we can still compare the **relationships** between the adjusted variable (`vmdaPct`) and other covariates conditional on pair. Here using `xBalance` because it can handle continuous treatments.

```
        chisquare df        p.value
unstrat      87.7 20 0.000000000186
nbp1         23.4 20 0.268460289702
```

vars	stat z	stat p
age	0.5962	0.551
income.coded	1.1822	0.237
educationbachelor's degree	-0.9720	0.331
educationcompleted secondary / high school	0.6371	0.524
educationcompleted technical, community college, CEGEP, College Classique	0.2952	0.768
educationmaster's degree	-0.3426	0.732
educationprofessional degree or doctorate	0.6229	0.533
educationsome secondary / high school	1.0000	0.317
educationsome technical, community college, CEGEP, College Classique	0.8143	0.415
educationsome university	-0.4494	0.653
educationNA	-0.4995	0.617
x.years	-1.2336	0.217
sexFemale	-0.0511	0.959
sexMale	0.3773	0.706
sexNA	-0.9712	0.331
csd.pop	0.3416	0.733
vm.csd	-1.5041	0.133
community_area_km	-1.0396	0.299
age.NATRUE	-0.5857	0.558

Assess balance: Approach with higher-vs-lower

No treatment and control groups to compare. But we can still compare the **relationships** between which person is higher versus lower on the adjusted variable (vmdaPct) and other covariates conditional on pair.

```
rank.pairs <- function(x, block) { ## Identify the low and high subj in each pair
  unsplit(lapply(split(x, block), function(x) {
    rank(x)
  }), block)
}
```

```
    nbp1 vmdaPct vmdaPct_ranked
1      1    17.8              0
164    1    18.2              1
2      2     0.0              1
132    2     0.0              0
3      3    16.8              1
71     3    16.7              0
     chisquare df p.value
nbp1      21.2 20   0.387
--        18.8 20   0.534
```

```
                                                                                stat
vars                                                                             Control   Treatme
  age                                                                            53.9500     53.10
0.05268    -0.85000
  income.coded                                                                    6.5467      6.37
0.05342    -0.17167
  educationbachelor's degree                                                      0.3210      0.32
  educationcompleted secondary / high school                                      0.0123      0.06
  educationcompleted technical, community college, CEGEP, College Classique      0.0988      0.15
```

Strength of the treatment

The difference in "treatment" within sets varies — and so we expect the size of the effect to vary. For example, consider the ratio of objective context differences to perceived context differences:

```
summary(nbp1vmdiffs)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.023   0.133   0.259   0.450   0.930
```

```
summary(nbp1percdiffs)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.100   0.193   0.290   0.339   0.440   0.790
```

```
percDist <- scalar.dist(wrkdat2$vm.community.norm2 * 100)
da <- vmdaDist[1:5, 1:5]
perc <- percDist[1:5, 1:5]
da / perc
```

```
     1    2     3      4       5
1  NaN 1.57  0.89 0.8000  0.1000
2 1.57  NaN  1.14 1.6846  3.5833
3 0.89 1.14   NaN 0.9000 15.7000
4 0.80 1.68  0.90    NaN  0.0211
5 0.10 3.58 15.70 0.0211     NaN
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.100   0.193   0.290   0.339   0.440   0.790
```

Assess hypotheses about effects

Test the hypothesis of no relationship between perceptions as measured by `vm.community.norm2` and `social capital`.

```
library(coin)
wrkdat2$nbp1F <- factor(wrkdat2$nbp1)
test1 <- independence_test(social.capital01 ~ vm.community.norm2 | nbp1F, data = wrkdat2[!is.na(w
test1
```

```
    Asymptotic General Independence Test

data:  social.capital01 by vm.community.norm2
     stratified by nbp1F
Z = -3, p-value = 0.002
alternative hypothesis: two.sided
```

Describe the differences within pairs

Does the person who perceives more visible minorities in their community tend to be higher (or lower) in `social.capital` than the other person in the pair?

```
wrkdat2$scRank <- with(wrkdat2, rank.pairs(social.capital01, nbp1))
wrkdat2$vmCRank <- with(wrkdat2, rank.pairs(vm.community.norm2, nbp1))
wrkdat2[order(wrkdat2$nbp1), c("nbp1", "social.capital01", "scRank", "vm.community.norm2", "vmCRa
```

```
    nbp1 social.capital01 scRank vm.community.norm2 vmCRank
1      1            0.667      2               0.25       1
164    1            0.333      1               0.60       2
2      2            0.750      2               0.40       1
132    2            0.583      1               0.63       2
3      3            0.500      1               0.45       1
71     3            0.667      2               0.61       2
```

```
with(wrkdat2, tapply(scRank, vmCRank, mean))
```

```
   1    2
1.63 1.37
```

Summarize mean differences within pairs

If perceptions matters for social capital then we would expect pairs differing greatly in subjective context to display greater differences in social capital than pairs that differ a little.

```
## By default, this rescales each observation to be the distance from the group mean.
align.by.block <- function(x, block, fn = mean, thenames = NULL) {
  newx <- unsplit(lapply(split(x, block), function(x) {
    x - fn(x)
  }), block)
  if (!is.null(names)) {
    names(newx) <- thenames
  }
  return(newx)
}
```

```
wrkdat2$scMD <- with(wrkdat2, align.by.block(social.capital01, nbp1))
wrkdat2$vmcn2MD <- with(wrkdat2, align.by.block(vm.community.norm2, nbp1))
wrkdat2[order(wrkdat2$nbp1), c("social.capital01", "scMD", "vm.community.norm2", "vmcn2MD", "nbp1
```

```
    social.capital01    scMD vm.community.norm2 vmcn2MD nbp1
1              0.667  0.1667               0.25  -0.175    1
164            0.333 -0.1667               0.60   0.175    1
2              0.750  0.0833               0.40  -0.115    2
132            0.583 -0.0833               0.63   0.115    2
```

```
## notice that aligning or pair-mean-centering the data preserves the within
## set relationships
## summary(tapply(wrkdat2$scMD,wrkdat2$nbp1,function(x){ abs(diff(x)) }))
## summary(tapply(wrkdat2$social.capital01,wrkdat2$nbp1,function(x){ abs(diff(x)) }))
lm1 <- lm_robust(scMD ~ vmcn2MD, data = wrkdat2[!is.na(wrkdat2$nbp1), ])
```

Summarize mean differences within pairs

```
             Estimate Std. Error   t value   Pr(>|t|) CI Lower CI Upper  DF
(Intercept)  2.96e-18    0.00695  4.25e-16 1.00000000  -0.0137   0.0137 162
vmcn2MD     -1.82e-01    0.03724 -4.90e+00 0.00000235  -0.2559  -0.1088 162
                   Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
vm.community.norm2   -0.182     0.0529   -3.45   0.0009   -0.288  -0.0771 81

 1  2
82 82
               Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
I(vmCRank - 1)  -0.0681     0.0198   -3.44 0.000916   -0.107  -0.0287 81
```

Summarize mean differences within pairs

If perceptions matter for social capital above and beyond objective context then we would expect pairs differing greatly in subjective context to display greater differences in social capital than pairs that differ a little.

```
lm2
```

```
             Estimate Std. Error   t value   Pr(>|t|) CI Lower CI Upper  DF
(Intercept)  2.96e-18    0.00695  4.25e-16 1.00000000  -0.0137   0.0137 162
vmcn2MD     -1.82e-01    0.03724 -4.90e+00 0.00000235  -0.2559  -0.1088 162
```

```
lm3
```

```
                   Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
vm.community.norm2   -0.182     0.0529   -3.45   0.0009   -0.288  -0.0771 81
```

```
pairdiffs <- wrkdat2 %>%
  filter(!is.na(vmCRank) & !is.na(social.capital01) & !is.na(nbp1)) %>%
  group_by(vmCRank) %>%
  summarize(mnsc = mean(social.capital01))
wrkdat2[order(wrkdat2$nbp1), c("social.capital01", "scRank", "scMD", "vm.community.norm2", "vmcn2
```

```
    social.capital01 scRank    scMD vm.community.norm2 vmcn2MD vmCRank nbp1
1              0.667      2  0.1667               0.25  -0.175       1    1
164            0.333      1 -0.1667               0.60   0.175       2    1
2              0.750      2  0.0833               0.40  -0.115       1    2
132            0.583      1 -0.0833               0.63   0.115       2    2
```

```
lm4
```

```
               Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper DF
I(vmCRank - 1)  -0.0681     0.0198   -3.44 0.000916   -0.107  -0.0287 81
```

Summarize mean differences within pairs

```
summary(wrkdat2$vmcn2MD)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -0.395  -0.145   0.000   0.000   0.145   0.395
```

```
summary(wrkdat2$scMD)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.2917 -0.0417  0.0000  0.0000  0.0417  0.2917
```

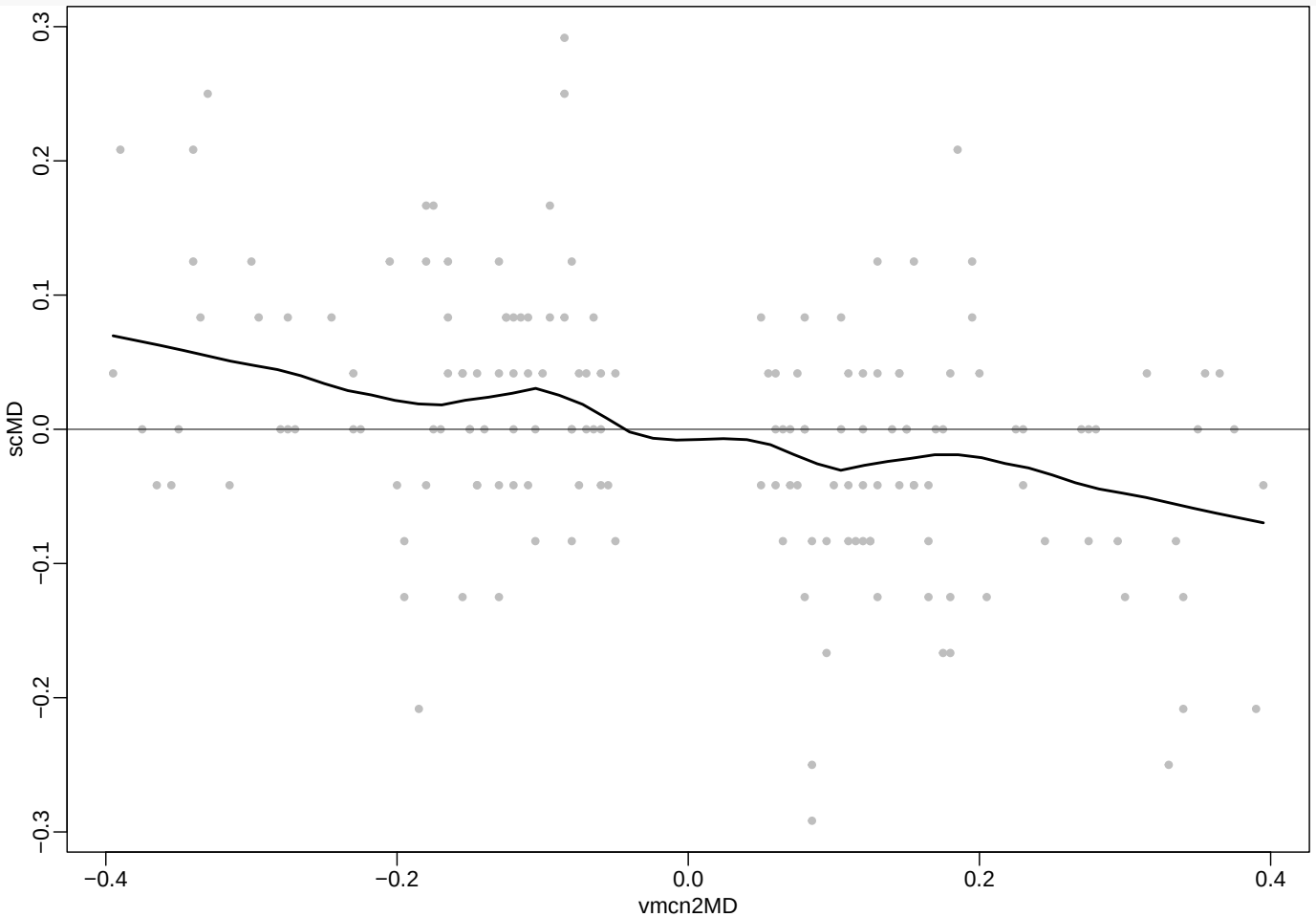
Within matched pair, the person who perceives more visible minorities within set tends to report lower social capital than the person who perceives fewer visible minorities within set.

The largest difference is about 0.4.
The model predicts that social capital would differ by about `r coef(lm1)[[2]]*.4` for such a difference. This is about `r coef(lm1)[[2]]*.4/sd(wrkdat2$scMD,na.rm=TRUE)` of a standard deviation of the social capital scale. Or about `r coef(lm1)[[2]]*.4/abs(diff(range(wrkdat2$scMD,na.rm=TRUE)))` of the range.

Summarize mean differences within pairs

Here is a look at the within-pair differences in perceptions of visible minorities as well as social capital.

```
with(wrkdat2, scatter.smooth(vmcn2MD, scMD, span = .3, cex = .7, col = "gray", pch = 19, lpars =
abline(h = 0, lwd = .5)
```



Summary of matching without groups

- Workflow in general is the same as matching with groups (covariates, distance matrices, optimization to select a stratification, assessment of the stratification by comparison to an experiment)
- Estimation is more flexible — could look simply at "higher versus lower" within pair, or could average over scores.

Another estimation approach

(Smith, 1997) presents a multi-level modelling approach to taking matched sets into account. The weights implied here are a bit different from the weights that we've discussed before (although with pairs they might be more or less the same). What is the data model? What additional assumptions are involved here?

```
  2.5 %  97.5 %
-0.2821 -0.0825

  1   2
158   3
  2.5 %  97.5 %
-0.2649 -0.0696
```

Other applications of non-bipartite matching?

See: DOS Chapter 11.
Also: this has a lot of applications in experimental design (see `blockTools` and (Ryan T Moore, 2012a,b)).

Next time:

- Sensitivity analysis: How different might our results be if units differed in their probability of treatment/selection/intervention **within strata**?

Remaining questions?

References

 Moore, Ryan T (2012a). "Multivariate continuous blocking to improve political science experiments". In: Political Analysis 20.4, pp. 460–479.

 — (2012b). blockTools: Block, assign, and diagnose potential interference in randomized exper R package.

 Rabb, Nathaniel et al. (July 2022). "The influence of social norms varies with "others" groups: Evidence from COVID-19 vaccination intentions". In: Proceedings of the National Academy of Sciences 119.29. DOI: https://doi.org/10.1073/pnas.2118770119.

 Smith, H.L. (1997). "Matching with Multiple Controls to Estimate Treatment Effects in Observational Studies". In: Sociological Methodology 27, pp. 325–353.

 Wong, Cara et al. (Oct. 2012). "Bringing the Person Back In: Boundaries, Misperceptions, and the Measurement of Racial Context". In: Journal of Politics 74.4, pp. 1153–1170.