

Precept Nine: Model Dependence and Matching

Rebecca Johnson

April 12th, 2017

Outline

- ▶ Problem of model dependence to motivate matching
- ▶ Intuition behind matching
- ▶ Implementation of matching (using example of Mahalanobis distance); focus on choices at each step:
 1. Check the need for matching/imbalance before matching
 2. Choose a number of matches
 3. Choose a distance metric/matching algorithm
 4. Find matches (drop non-matches)
 5. Check balance
 6. Repeat 2-5 until balance is acceptable
 7. Calculate effect of treatment on the outcome in matched dataset

Motivating example: observational study that tries to estimate the treatment effect of ads on political contributions

Dollars on the Sidewalk: Should U.S. Presidential Candidates Advertise in Uncontested States?

Carly Urban Montana State University

Sarah Niebler Dickinson College

Framing article in potential outcomes notation...

- ▶ Y_i : outcome for observation i ...in this case:
 - ▶ i : geographic units (zip codes) composed of residents contributing to campaigns
 - ▶ Y : total campaign contributions (in 1,000s) for the 2008 election campaign given to one of the two major-party candidates (Obama/Biden and McCain/Palin)
- ▶ D_i : binary indicator for treatment status for observation i ...in this case:
 - ▶ $D_i = 1$: zip code receives treatment of campaign ads meant for adjacent zip codes in battleground states that share a media market with zip code of interest (spillover ads for shorthand)¹
 - ▶ $D_i = 0$: zip code does not receive treatment
- ▶ X_i : vector of zip-code level covariates like income, race, and education level of residents

¹More precisely, the authors take a continuous measure of ad dollars spent in different media markets and discretize it to count as a 'treatment effect' of ads if $> \$1000$ was spent; they check results' robustness to this threshold

Two assumptions behind identifying the treatment effect of ads

1. Selection on observables (also called unconfoundedness, ignorability, no unmeasured confounding)

- ▶ In general: $(Y_1, Y_0) \perp D | X$
- ▶ Specific case (in math):

$$(Contributions(ads), Contributions(no ads)) \perp Ads | X$$

- ▶ Specific case (in words): the potential campaign contributions from treated zip codes and from control zip codes are independent from treatment status (receiving ads) once we condition on covariates like resident income, education, etc.

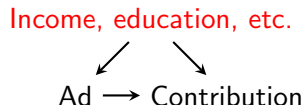
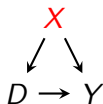
2. Common support:

- ▶ In general: $0 < Pr(D = 1 | X) < 1$ with probability one
- ▶ Specific case (in words): for zip codes with any level of covariate(s), there is a positive probability of receiving the treatment of ads (or not receiving the treatment of ads)

How regression fits into this framework (which then motivates how matching fits into this framework)

$$(Y_1, Y_0) \perp D | X$$
$$(Cont(ads), Cont(no\ ads)) \perp Ads | X$$

- ▶ **Regression:** choose a set of covariates X that we think helps us satisfy this assumption (so X to condition on to break the association between whether a zip code receives ads and its potential campaign contributions)
- ▶ In DAG terms, choose what goes into the elements of the vector that compose the node **in red**- or as lecture slide 34 put it, "when selection on observables holds, we still need to **adjust for X_i** "



Problem of model dependence (Ho et al., 2007)

- ▶ Begin with the ATT (using D_i to represent treatment):

$$\frac{1}{\sum_{i=1}^n D_i} \sum_{i=1}^n D_i [\mu_1(X_i) - \mu_0(X_i)]$$

- ▶ Rewrite the **treatment** (receive ads) and **control** (don't receive ads) outcomes in terms of a model that links the mean of the outcome variable to covariates like the town's median income, etc.:

$$\mu_1(X_i) = E[Y_i(1)|D_i = 1, X_i] = g(\alpha + \beta + X_i\gamma),$$

$$\mu_0(X_i) = E[Y_i(0)|D_i = 0, X_i] = g(\alpha + X_i\gamma)$$

- ▶ In experiments, the **parts in red** drop out due to randomization, but without experiments, we have researcher discretion in various choices

Choices researcher can make

$$E[Y_i(1)|D_i = 1, X_i] = g(\alpha + \beta + X_i\gamma),$$

$$E[Y_i(0)|D_i = 0, X_i] = g(\alpha + X_i\gamma)$$

- ▶ $g(\cdot)$: Which parametric model? (parametric: choose a distribution for Y and link function to relate X to Y ; e.g., choose linear regression v. poisson in present case)
- ▶ X : after choosing a parametric model, what to include in X (e.g., if we include the town's median income, also include education?)
- ▶ After choosing a parametric model *and* choosing X , choices about how to represent the relationship between X and Y (e.g., income alone or income + income²)
- ▶ Why do we not like discretion? Potential for *subconscious* cherry-picking of models to support priors/find significant results

Empirical illustration from ads-contribution example

Income, education, etc.



- ▶ Begin with the following predictors plausibly correlated with both whether a town receives political ads and the town's potential total campaign contributions:
 1. Median household income
 2. Median household income²
 3. Percent black
 4. Percent hispanic
 5. Percent college graduates
 6. Percent college graduates²
 7. Population density

To illustrate model dependence even after we decide that X should include some combination of these 7 predictors...

1. Enumerate all combinations² of 7 predictors of size 2, 3, ... 7
2. Run a linear regression of contributions on the treatment of ads (1 = yes; 0 = no) for each combination found in step 1...e.g.:

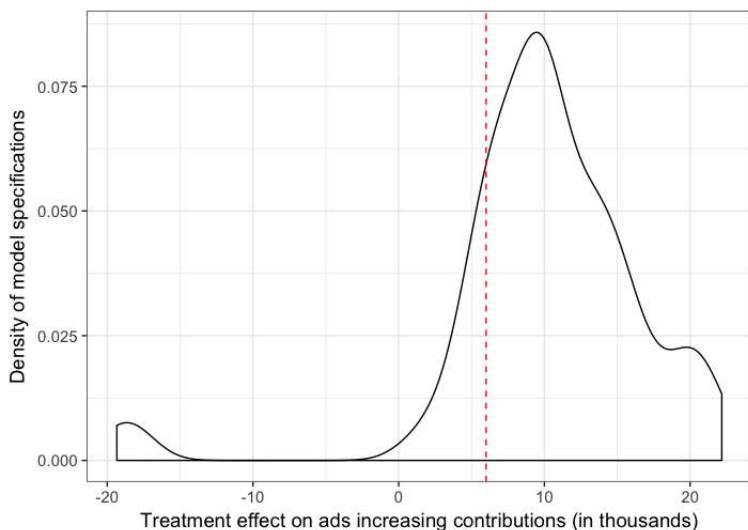
$$\text{contributions}_i = \alpha + \beta_1 \times \text{ad}(1 = \text{yes}; 0 = \text{no})_i + \beta_2 \times \text{income}_i \\ + \beta_3 \times \text{percblack}_i + \beta_4 \times \text{density}_i$$

$$\text{contributions}_i = \alpha + \beta_1 \times \text{ad}(1 = \text{yes}; 0 = \text{no})_i + \beta_2 \times \text{income}_i \\ + \beta_3 \times \text{percblack}_i + \beta_4 \times \text{perchisp}_i + \beta_5 \times \text{density}_i$$

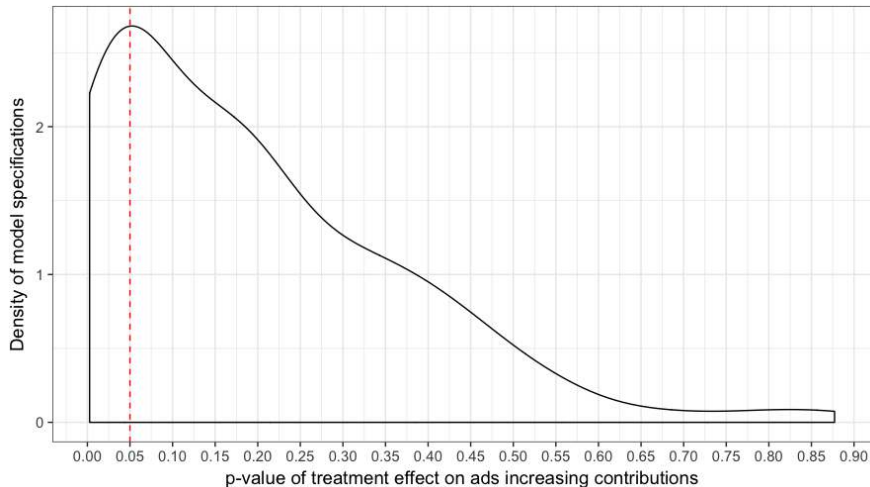
3. Extract coefficient and p-value for treatment variable of interest (**ads** (1 = yes; 0 = no)) and plot the distribution across the combinations from 1
 - **Higher model dependence (bad!)**: estimate of 'treatment effect' changes a lot across specifications

²We're simplifying this slightly because, for instance, we might not want to count a combination as valid if it includes income² but not income

Results: variation in estimated treatment effect across model specifications



Results: variation in significance level of estimated treatment effect across model specifications



How to generate using R (and like last week, a refresher on `lapply` for PSet 6...)

Step one: enumerate all combinations of seven predictors

```
##start with a vector of predictors
predictors
[1] "Inc"      "PercentHispanic"  "PercentBlack"
"density"   "per_collegegrads" "Inc_sq" "College_sq"

##use combn in conjunction with apply to choose 2, 3...7
##predictors from that length 7 vector
all_predictors <- sapply(m,
function(x) {combn(predictors, m = x)})
```

Results in following count of combinations for a regression with 2, 3, ... 7 predictors:

Predictors	Combinations of the 7 possible variables
2	21
3	35
4	35
5	21
6	7
7	1

Step two: run a linear regression of contributions on the treatment of ads (1 = yes; 0 = no) for each combination found in step 1

```
##create regression formulas for each combination (each comb.
## is a matrix column so we apply over columns (2))
all_predictors_form <- lapply(all_predictors,
                             function(x) {apply(x, 2,
                                                  function(y) paste(y, collapse = "+"))})

##add treatment to each regression formula
all_predictors_form_with_tx <- lapply(all_predictors_form,
                                     function(x) {lapply(x, function(y)
                                                         as.formula(paste("Cont ~ Treatment1 +",
                                                         y))))})

##feed each formula to lm
all_models_results <- lapply(all_predictors_form_with_tx,
                             function(x) lapply(x,
                                                  function(y) (lm(y, data = nj))))
```

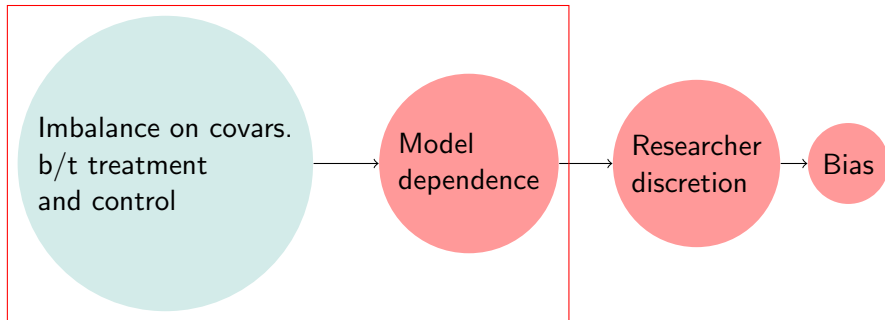
Step three: extract coefficient and p-value for treatment variable of interest (ads (1 = yes; 0 = no) and plot the distribution across the combinations from step 1

```
##extract coefficient on tx variable (ads)
all_tx_coef <- lapply(all_models_results,
                      function(x) lapply(x,
                      function(y) coefficients(y)["Treatment1"])))

##extract the p value
all_tx_p <- lapply(all_models_results,
                   function(x) lapply(x,
                   function(y)
                   summary(y)$coefficients["Treatment1",
                   4])))

##plotting code in solutions .rmd
```

Contributors to model dependence we saw in wide variation in treatment values and wide variation in p values (Lecture slide 44)



Leads us to matching as a pre-processing technique for addressing imbalance between treatment and covariates



Imbalance on covars.
b/t treatment
and control

General matching procedure

1. **Check the need for matching/imbalance before matching**
2. **Choose a number of matches**
 - 2.1 Choose how many controls to match to each treatment unit (e.g., $M = 1$; $M > 1$)
 - 2.2 When matching controls to treatment, choose:
 - 2.2.1 *Sampling without replacement*: each control serves as a match for a max of 1 treatment unit (can also serve as match for 0 treatment units)
 - 2.2.2 *Sampling with replacement*: each control can serve as a match for more than one treatment unit (still can serve as a match for 0 treatment units)
3. **Choose a distance metric/matching algorithm**
4. **Find matches (drop non-matches)**
 - ▶ If chose $M > 1$ or *sampling with replacement*: each control will have a weight that represents how much it contributes to data once included rather than a **1** = included; **0** = not included
5. **Check balance**
6. **Repeat 2-5 until balance is acceptable**
7. **Calculate effect of treatment on the outcome in matched dataset**

We'll review each step in detail, but before, general intuition behind matching as pre-processing to correct imbalance between treatment and control units

Intuition: specify the correct counterfactual for each treatment unit/the treatment units as a whole

To illustrate, we'll focus on New Jersey within the treatment effect of ads on contributions data, and focus on a particular treated observation w/ which many of us are familiar...

Princeton as a treatment unit

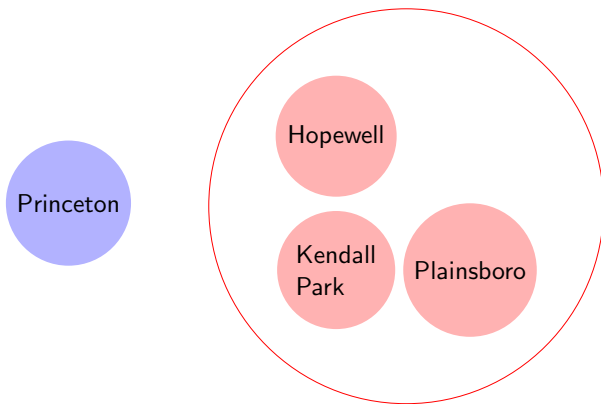


- Each i is a zip code, and associated with two zip codes in the data, both of which were treated with spillover ads likely from targeting of Pennsylvania media markets:

Contrib.	D_i	zip	Inc	% Hisp	% Black	pop. dens	% college grads
1155.20	1	08540	90.98	0.04	0.05	811.98	71.39
87.34	1	08542	56.15	0.25	0.12	8958.48	52.88

Focusing on 08540, approaches we could take to finding a counterfactual for Princeton ($D_i = 1$) from a pool of controls ($D_i = 0$)

Treatment zip code Control zip code

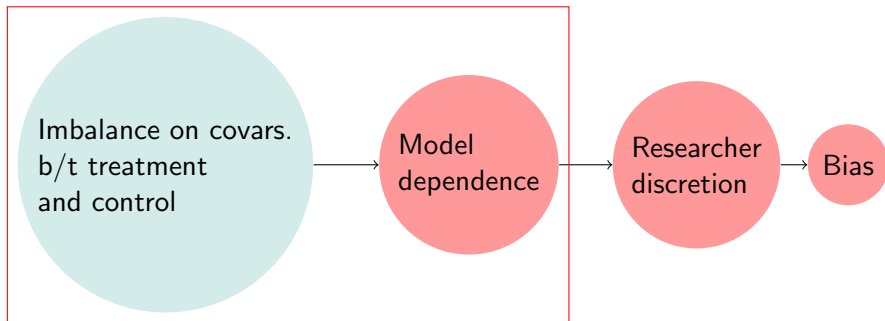


Matching by intuition/researcher discretion (not a real approach)!

City	Contrib.	D_i	zip	Inc	% Hisp	% Black	pop. dens	% college
Princeton	1155.20	1	08540	90.98	0.04	0.05	811.98	71.39
Hopewell	49.96	0	08525	90.44	0.02	0.02	216.46	53.58
Plainsboro	29.39	0	08536	70.65	0.05	0.08	3148.13	70.48
Kendall P.	9.23	0	08824	89.35	0.04	0.04	3003.76	44.54

- ▶ How do we decide, for instance, which of the following we should use as a match for Princeton?
 1. *Hopewell*: close in terms of income but much lower population density and college education levels
 2. *Plainsboro*: more similar college education but lower income/higher minority population
 3. *Weighted combination of 0.75 Hopewell, 0.25 Plainsboro*: Hopewell contributes similar income/demographics; Plainsboro contributes population density
- ▶ Could use intuition to decide the most plausible counterfactual unit for Princeton, but better to use a data-driven approach

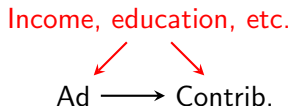
To decide which approach to use, return to original problem we're trying to solve with matching



And focus on step one from our general procedure

1. Check the need for matching/imbalance before matching
2. Choose a number of matches
3. Choose a distance metric/matching algorithm
4. Find matches (drop non-matches)
5. Check balance
6. Repeat 2-5 until balance is acceptable
7. Calculate effect of treatment on the outcome in matched dataset

Ways to check for imbalance



1. **Imbalance variable by variable:** iterate through each variable you think might be relevant for the paths in red above and...
 - ▶ Compare summary statistics for each covariate between the treatment group ($D_i = 1$; receives ad) and control group ($D_i = 0$; doesn't receive ad)
 - ▶ Visualize density of each variable between treatment and control
2. **Imbalance in joint distribution of variables:** how do we get a more global summary measure of imbalance that looks at imbalance across all the variables of interest? One we'll review:
 - ▶ Multivariate $\mathcal{L}_1$ (in cem package in R)

Ways to check for imbalance

1. **Imbalance variable by variable:** iterate through each variable you think might be relevant for the paths in red above and...
 - ▶ Compare summary statistics for each covariate between the treatment group ($D_i = 1$; receives ad) and control group ($D_i = 0$; doesn't receive ad)
 - ▶ Visualize density of each variable between treatment and control
2. **Imbalance in joint distribution of variables:** how do we get a more global summary measure of imbalance that looks at imbalance across all the variables of interest? One we'll review:
 - ▶ Multivariate $\mathcal{L}_1$ (in cem package in R)

Implementing in R

```
balance_function <- function(data, vars_of_interest,
                             treatment){
  ##restrict data to treatment + variables of interest
  variables <- data[, c(treatment, vars_of_interest)]

  ##summarize the mean and variance of each column in
  ##the data after grouping by treatment
  ##the weird group_by is just so it takes a string
  summary_eachvar <- as.data.frame(variables %>%
    group_by_(.dots = treatment) %>%
    summarize_each(funs(mean, var)))

  ##order in terms of column names
  ##and transpose
  return(t(summary_eachvar[, order(names(summary_eachvar))]))
}
```

Results: see that units that receive ads are more rural (lower pop density), have a slightly lower median income and fewer college grads

	Control	Treatment
density_mean	4352.91	2502.20
density_var	34677536.23	74245806.12
Inc_mean	68.19	51.60
Inc_var	591.50	255.21
per_collegegrads_mean	34.79	21.46
per_collegegrads_var	276.84	193.16
PercentBlack_mean	0.08	0.12
PercentBlack_var	0.03	0.03
PercentHispanic_mean	0.10	0.06
PercentHispanic_var	0.02	0.01

Will want to use matching to see better balance between treatment and control on these variables!

Ways to check for imbalance

1. **Imbalance variable by variable:** iterate through each variable you think might be relevant for the paths in red above and...
 - ▶ Compare summary statistics for each covariate between the treatment group ($D_i = 1$; receives ad) and control group ($D_i = 0$; doesn't receive ad)
 - ▶ Visualize density of each variable between treatment and control
2. **Imbalance in joint distribution of variables:** how do we get a more global summary measure of imbalance that looks at imbalance across all the variables of interest? One we'll review:
 - ▶ Multivariate $\mathcal{L}_1$ (in `cem` package in R)

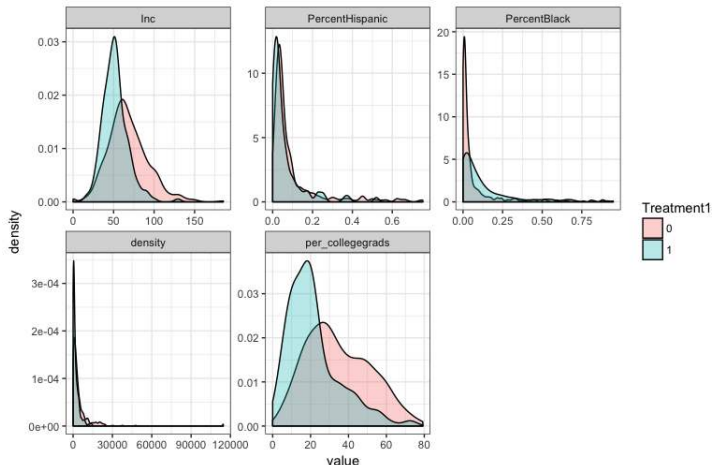
Implementing in R

```
density_function <- function(data, vars_of_interest,
                             treatment){
  ##restrict data to treatment + variables of interest
  wide_data <- data[, c(treatment, vars_of_interest)]

  ##reshape to long format to use facet_wrap in
  ##plotting
  data_long <- melt(variables, id.vars = treatment)

  ##plot with facet_wrap- subsetting inside
  ##as.factor is again due to ggplot and strings
  ggplot(data_long, aes(x = value)) +
    geom_density(aes(fill = as.factor(data_long[[treatment]])),
                 alpha = 0.3) +
    facet_wrap(~ variable, scales = "free") +
    labs(fill = treatment) +
    theme_bw()
}
```

Results



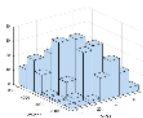
Notice differences in not only mean but also distribution of income and percent college grads; density difference is still there but not showing up as clearly on graph because of some very high density zip codes

Ways to check for imbalance

1. **Imbalance variable by variable:** iterate through each variable you think might be relevant for the paths in red above and...
 - ▶ Compare summary statistics for each covariate between the treatment group ($D_i = 1$; receives ad) and control group ($D_i = 0$; doesn't receive ad)
 - ▶ Visualize density of each variable between treatment and control
2. **Imbalance in joint distribution of variables:** how do we get a more global summary measure of imbalance that looks at imbalance across all the variables of interest? One we'll review:
 - ▶ Multivariate $\mathcal{L}_1$ (in `cem` package in R)

Motivation and background for multivariate $\mathcal{L}_1$ (Ho et al., 2007, p. 220)

- ▶ **Motivation:** previous summary statistics and plots went variable by variable to check imbalance, but we may want a more global measure of imbalance across all predictors of interest
- ▶ Can break down into two parts:
 1. $\mathcal{L}_1$: jargon for measure of distance ($\sum_{i=1} |a_i - b_i|$)
 2. *Multivariate*: rather than iterating over each variable and defining the distance, want to calculate one measure of distance for all the variables
- ▶ **General idea:** calculate distance between heights in multidimensional histogram for covariates of interest (k) for treatment observations and multidimensional histogram for covariates of interest (k) in control observations



Motivation and background for multivariate $\mathcal{L}_1$

► More specifically:

1. Coarsen the k covariates into bins (function can also do automatically)
2. Separately for tx and control group, generate cross-tabs that represent cell counts of different combinations of variables/bins
3. Repeat for the control group
4. End up with two length- k vectors of counts (treatment vector; control vector)– calculate absolute value of difference and average across variables, where f is treated and g is control:³

$$\mathcal{L}_1(f, g) = \frac{1}{2} \sum_{\ell_1 \dots \ell_k} |f_{\ell_1 \dots \ell_k} - g_{\ell_1 \dots \ell_k}|$$

► **Interpretation:**

- **Higher** $\mathcal{L}_1$: greater distance across variables; more imbalance and more need for matching
- **Lower** $\mathcal{L}_1$: smaller distance across variables; less imbalance and more need for matching

³Note: the source for equation uses notation ℓ but we should not confuse that with log-likelihood!

Implementation in R

```
##load CEM package
library(cem)

##restrict to vars of interest
nj_varsofint <- nj_withz[, c("Treatment1", vars_of_interest)]

##run function
imbalance_pre <- imbalance(group = nj_varsofint$Treatment1,
data = nj_varsofint, drop = c("Treatment1"))
```

Result: $\mathcal{L}_1 = 0.787$

Where we go from here

- ▶ Summary statistics, density plots, and global imbalance all show imbalance between treatment group (receives spillover ads) and control group (lack of spillover ads) on covariates that might also be correlated with campaign contributions
- ▶ Suggests need for matching/next steps in the process

Moving to steps 2-4

1. Check the need for matching/imbalance before matching
2. Choose a number of matches
 - 2.1 Choose how many controls to match to each treatment unit (e.g., $M = 1$; $M > 1$)
 - 2.2 When matching controls to treatment, choose:
 - 2.2.1 *Sampling without replacement*: each control serves as a match for a max of 1 treatment unit (can also serve as match for 0 treatment units)
 - 2.2.2 *Sampling with replacement*: each control can serve as a match for more than one treatment unit (still can serve as a match for 0 treatment units)
3. Choose a distance metric/matching algorithm
4. Find matches (drop non-matches)
 - ▶ If chose $M > 1$ or *sampling with replacement*: each control will have a weight that represents how much it contributes to data once included rather than a **1** = included; **0** = not included
5. Check balance
6. Repeat 2-5 until balance is acceptable
7. Calculate effect of treatment on the outcome in matched dataset

Brief note: why are we collapsing these steps?

1. Important *conceptual* differences in each step and each requires different decisions
 - ▶ Number of matches...decisions:
 - 1.1 One control or more than one control for treatment?
 - 1.2 Match controls to treatments with or without replacement?
 - ▶ Choose a distance metric/matching algorithm...decisions:
 - 1.1 Algorithm that minimizes distances between covariates or algorithm that uses covariates to estimate a single score and minimizes distances between scores (propensity score matching)?
 - 1.2 Within each approach, many variations (e.g. estimate propensity score with logistic regression or more complex prediction model)
2. *Implementation-wise* in R, we often specify these decisions in the same command by passing the command different arguments

Matching algorithm to anchor these steps

1. **Method 1: Mahalanobis distance (covered in Weds. lecture)**
2. *If time in code*: Propensity score (covered in Monday lecture)
3. *Next week*: coarsened exact matching (CEM) (covered in Monday lecture)

For many methods

- ▶ Covariates to match on (same as those in Urban and Niebler (2013)):
 1. Median household income
 2. Percent black
 3. Percent hispanic
 4. Percent college graduates
 5. Population density
- ▶ Package to use: MatchIt
- ▶ When choosing m , keeping in mind the count of treatment and control observations which can constrain how many control units we match to each treatment unit if matching without replacement:

Control	Treatment
(Treatment1 = 0)	(Treatment1 = 1)
395	186

Method 1: Mahalanobis distance

► General idea in words:

- For two cities, calculate the distance between two covariates
- When creating the summary measure of distance across covariates, make small distances between *loosely correlated* covariates count more towards a smaller distance than small distances between two *highly correlated* covariates

► In math:

$$Distance(X_i, X_j) = \sqrt{(X_i - X_j)S^{-1}(X_i - X_j)}$$

- S in present data: less important to have small distance between income/college since two are highly correlated; more important between density and college since two are less correlated:⁴

	Inc	% Hisp	% Black	Density	% College
Inc	1.00	-0.41	-0.42	-0.34	0.83
% Hisp	-0.41	1.00	0.32	0.52	-0.34
% Black	-0.42	0.32	1.00	0.39	-0.36
density	-0.34	0.52	0.39	1.00	-0.19
% College	0.83	-0.34	-0.36	-0.19	1.00

⁴For instance, you might have low density suburbs with mansions or low density rural areas

Method 1: Mahalanobis distance- implementing via MatchIt...each control only matches with one treatment unit

► Choices:

- *How to calculate distance*: mahalanobis distance
distance = "mahalanobis"
- *Method for matching treatment units after calculation of distance*:
nearest neighbor
method = "nearest"
- *How many units to match with each treatment*: 2
ratio = 2
- *Can controls to match with multiple treatment units?*: no
replace = FALSE

► Putting it together:

```
mahal.out_nomult <- matchit(Treatment1 ~ Inc +  
PercentHispanic + PercentBlack + density + per_collegegrads,  
data = nj_withz, method = "nearest", distance = "mahalanobis",  
ratio= 2, replace = FALSE)
```

Method 1: Mahalanobis distance- implementing via MatchIt...each control can match with multiple treatment units

► Choices:

- *How to calculate distance*: mahalanobis distance
distance = "mahalanobis"
- *Method for matching treatment units after calculation of distance*:
nearest neighbor
method = "nearest"
- *How many units to match with each treatment*: 2
ratio = 2
- **Can controls to match with multiple treatment units?**: yes
replace = TRUE

► Putting it together:

```
mahal.out_mult <- matchit(Treatment1 ~ Inc + PercentHispanic +  
                          PercentBlack + density + per_collegegrads,  
                          data = nj_withz, method = "nearest",  
                          distance = "mahalanobis",  
                          ratio= 2, replace = TRUE)
```

Method 1: useful quantities to extract from the output of `matchit`

In this case, we can use to assess differences produced by choice to allow for controls to only match one or to match multiple treatment units

1. Summary table of number of control units matched
2. Weights on observations: need for balance function to check means, density plot function for visualization, and eventually regression we feed the matched data to
 - ▶ **Treatment units:** weights should be 1– remember from lecture that discarding treatment units changes our quantity of interest to the feasible ATT, so for now, we're keeping all treatment units
 - ▶ **Control units:** weights will depend on method:
 - ▶ *Match without replacement:* each control unit either does not get matched with a treatment ($w = 0$) or gets matched with one treatment ($w = 1$)
 - ▶ *Match with replacement:* each control unit may not get matched ($w = 0$) but may also have a non-zero weight depending on how many units they are matched with $w > 0$

1. Summary table of matched counts

- ▶ How many of control units are matched in each method? we know the answer for the only matching one (treatment $\times$ 2), but are interested in whether *allowing* multiple matches reduces to each control still only being used once:

```
mahal.out_nomult$nn
```

```
mahal.out_mult$nn
```

- ▶ Results:
 - ▶ Control matches with one unit

	Control	Treated
All	395.00	186.00
Matched	372.00	186.00
Unmatched	23.00	0.00

- ▶ Control can match with multiple units:

	Control	Treated
All	395.00	186.00
Matched	140.00	186.00
Unmatched	255.00	0.00

2. Weights to use for remainder of analysis

- ▶ How to find:

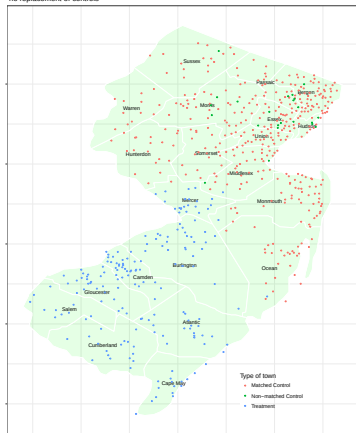
```
##find weights
mahal.out_nomult_w <- mahal.out_nomult$weights
mahal.out_mult_w <- mahal.out_mult$weights
##append weights to data
nj_withz <- nj_withz %>%
  mutate(weights_mahalomult = mahal.out_nomult_w,
         weights_mahalmult = mahal.out_mult_w)
```

- ▶ Example of how which observations are used as controls changes between methods:

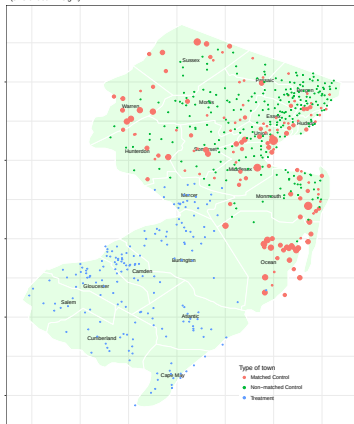
D_i	city	zip	Weights no multiple	Weights allow multiple
0	Spring Lake	07762	1.00	0.75
1	Camden	08104	1.00	1.00
1	Wildwood	08260	1.00	1.00
0	Woodbridge	07095	1.00	0.00
0	Ramsey	07446	1.00	0.00

Visual comparison of control towns used as matches versus not used as matches depending on whether same town can match multiple treatments

Control units matched and not matched:
Mahalanobis distance, 2 controls:treatment,
no replacement of controls



Control units matched and not matched:
Mahalanobis distance, 2 controls:treatment,
allow replacement of controls
(size of dot = weight)



Returning to our general steps...

1. Check the need for matching/imbalance before matching
2. Choose a number of matches
3. Choose a distance metric/matching algorithm
4. Find matches (drop non-matches)⁵
5. Check balance
6. Repeat 2-5 until balance is acceptable
7. Calculate effect of treatment on the outcome in matched dataset

⁵Note: the observations with weight = 0 like Woodbridge and Ramsey will be dropped from analyses as long as we specify to use weights

Checking balance in matched data

- ▶ Just as we used three balance metrics *prior* to matching to assess balance/the need for matching, we can return to the same three ways of assessing balance:
 1. Compare summary statistics for each covariate between the treatment group ($D_i = 1$; receives ad) and control group ($D_i = 0$; doesn't receive ad)
 2. Visualize density of each variable between treatment and control
 3. Multivariate $\mathcal{L}_1$ (in `cem` package in R)
- ▶ What changes? We're now feeding it the *matched* data...which means:
 - ▶ If we did a simpler matching algorithm where each control observation is either kept with $w = 1$ or discarded due to $w = 0$, we can prune the data to these observations and run the above functions without a `weights` argument
 - ▶ How do we do this? Either filter out all observations where `weights = 0` or in the original `matchit` command, specify `discard = "control"`
 - ▶ If we did an algorithm where some control observations have $w \neq \{0, 1\}$, we need to make sure to run the balance functions with a `weights` argument

Checking balance with *weighted* mean for allowing each control to match with more than one treated unit

```
variables <- nj_withz[, c(treatment, vars_of_interest, weights)]  
weighted_summary <- t(as.data.frame(variables %>%  
  group_by(.dots = treatment) %>%  
  summarize_each(funs(weighted.mean(.,  
    w = weights_mahalmult))))))
```

See .rmd solutions for how to update the balance function we showed on a previous slide to deal with weighted.means

Compare balance to data before matching

	Control Matched	Treatment Matched	Control NonMatched	Treatment NonMatched
Inc	55.36	51.60	68.19	51.60
PercentHispanic	0.07	0.06	0.10	0.06
PercentBlack	0.09	0.12	0.08	0.12
density	2564.38	2502.20	4352.91	2502.20
per_collegegrads	24.86	21.46	34.79	21.46

$$\text{New } \mathcal{L}_1 = 0.691$$

Improved balance, especially on the density covariate!

Returning to our general steps...

1. Check the need for matching/imbalance before matching
2. Choose a number of matches
3. Choose a distance metric/matching algorithm
4. Find matches (drop non-matches)
5. Check balance
6. Repeat 2-5 until balance is acceptable- If doing in real research, would probably add more covariates to try to improve the balance further!
7. Calculate effect of treatment on the outcome in matched dataset

Step 7: calculate the effect of treatment on the outcome in matched dataset

We will discuss in Monday's lecture two options; for now, we're implementing second:

- ▶ Estimate simple (weighted) difference in means between treatment and control
- ▶ What Ho et al. recommend: feed the matched data to the parametric model one was planning on estimating in the first place; in this case, that's a simple linear regression of campaign contributions on the treatment indicator of ads, controlling for various covariates
- ▶ **Important to note:** we should *not* be looking at the outcome variable until this step because we don't want our matching choices to be driven by what will produce significance effects between the matched treatment and control

Step 7: calculate the effect of treatment on the outcome in matched dataset

```
summary(lm(Cont ~ Treatment1 + Inc +  
          PercentHispanic + PercentBlack + density +  
          per_collegegrads, data = variables,  
          weights = weights_mahalmult))
```

Matching gets us a more conservative estimate on the treatment/null results for this particular state

Estimate with non-matched data

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-84.4950	13.8492	-6.10	0.0000
Treatment1	20.2604	7.4060	2.74	0.0064
Inc	0.2901	0.2533	1.15	0.2526
PercentHispanic	78.3532	31.4541	2.49	0.0130
PercentBlack	69.1403	21.1332	3.27	0.0011
density	0.0003	0.0005	0.50	0.6150
per_collegegrads	2.4498	0.3312	7.40	0.0000

Estimate with matched data

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-53.3672	17.1821	-3.11	0.0021
Treatment1	11.1512	7.2010	1.55	0.1225
Inc	-0.0003	0.3535	-0.00	0.9994
PercentHispanic	75.3112	49.5881	1.52	0.1298
PercentBlack	16.6408	25.8659	0.64	0.5205
density	0.0002	0.0006	0.34	0.7316
per_collegegrads	2.4375	0.3904	6.24	0.0000