Matching

Stephen Pettigrew

April 15, 2015

Outline

Outline

Things

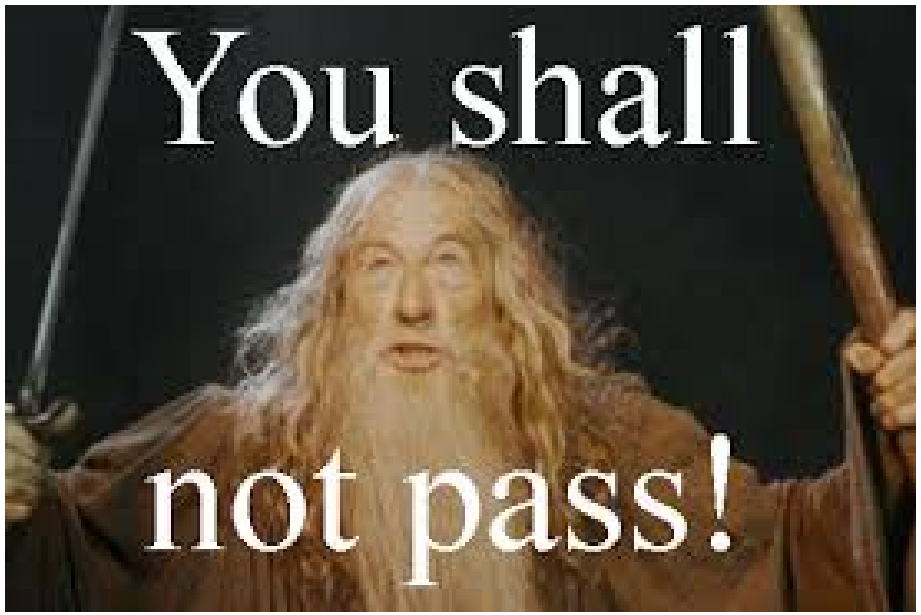
- Last pset and assessment 7 due next week
- Sole's office hours moved from Friday to Tuesday (3-5pm, CGIS-K cafe)
- RSVP for the party by tonight

Final three sections

- This week: matching
- Next week: multiple equation models and missing data imputation
  Two weeks from now: open (non-filmed) office hours for you to come ask questions about your final papers

Final papers

- You should have posted your final paper abstracts on Canvas. Now go through and provide feedback to your classmates about their abstracts
- Final paper is due on April 29. If you want an extension, you can have it until May 4 at 5pm. If you don't turn in your paper by then...



Outline

General Strategy of Matching

1. Determine the variables you want to match on. It's important to match on any potential confounders as well as any imbalanced covariates.
2. Choose at matching method (exact, Mahalanobis distance, propensity score, coarsened exact matching, or others).
3. Match the treatment and control observations in your data according to the variables and method you chose. Prune any observations that don't have good enough matches.
4. Assess the matching procedure by rechecking the balance of your dataset. Iterate through steps 1 through 3 until you're comfortable with the balance in your dataset.
5. Use parametric analysis (regress, t-test, etc.) to estimate your treatment effect of interest.
6. Perform sensitivity tests to check the assumptions of the matching or modeling procedure.

Things to think about while matching

tinder™

1. Which distance metric to use
2. How to turn distances into matches
3. How to prune the data as you match
4. With/without replacement

Choosing a distance metric

You want to match together each treatment unit to the control unit (or units) that is most similar to the treatment unit based on pretreatment covariates.

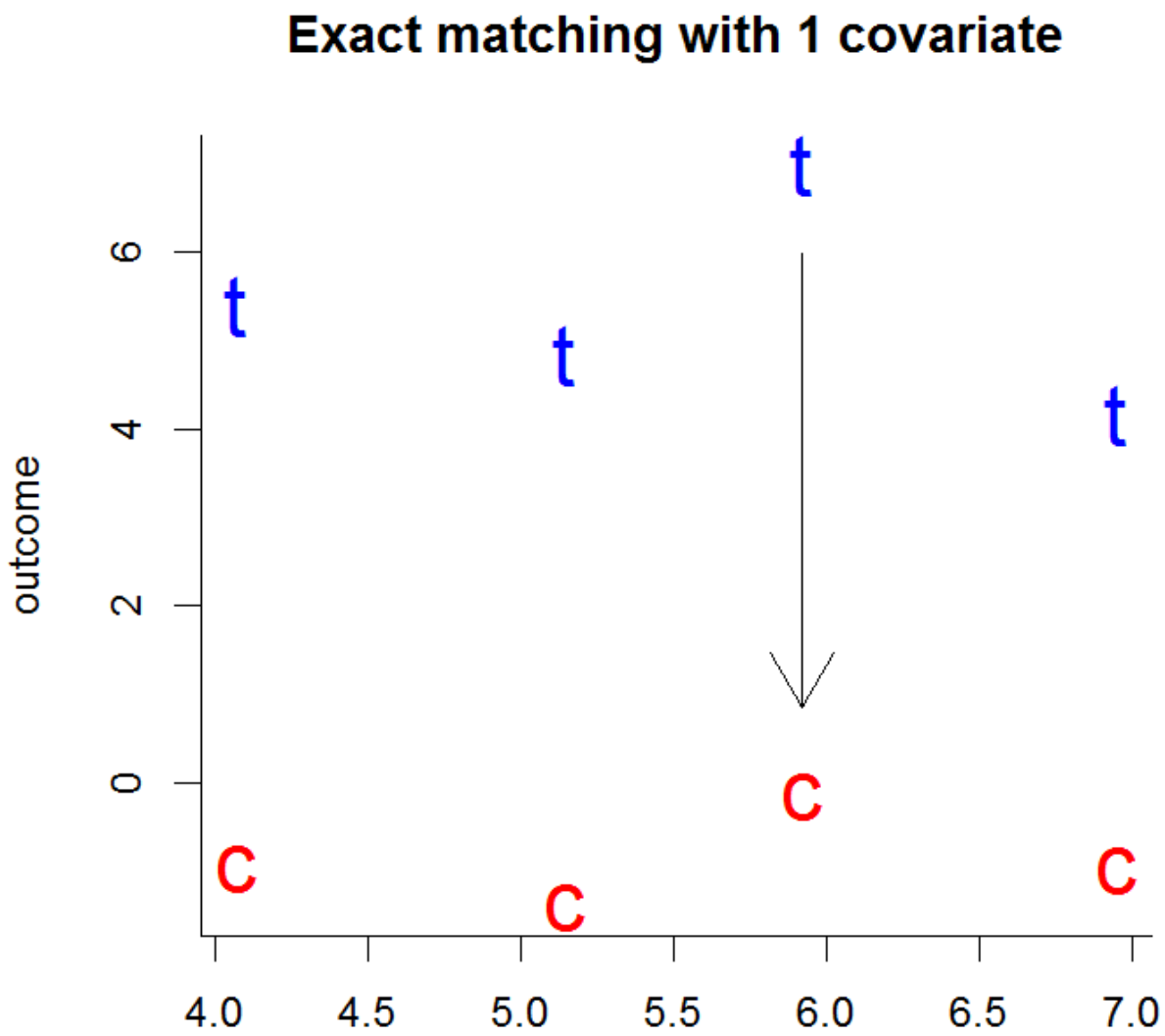
By doing this, you're essentially trying to find control units which can serve as a counterfactual for each treatment unit.

Exact matching

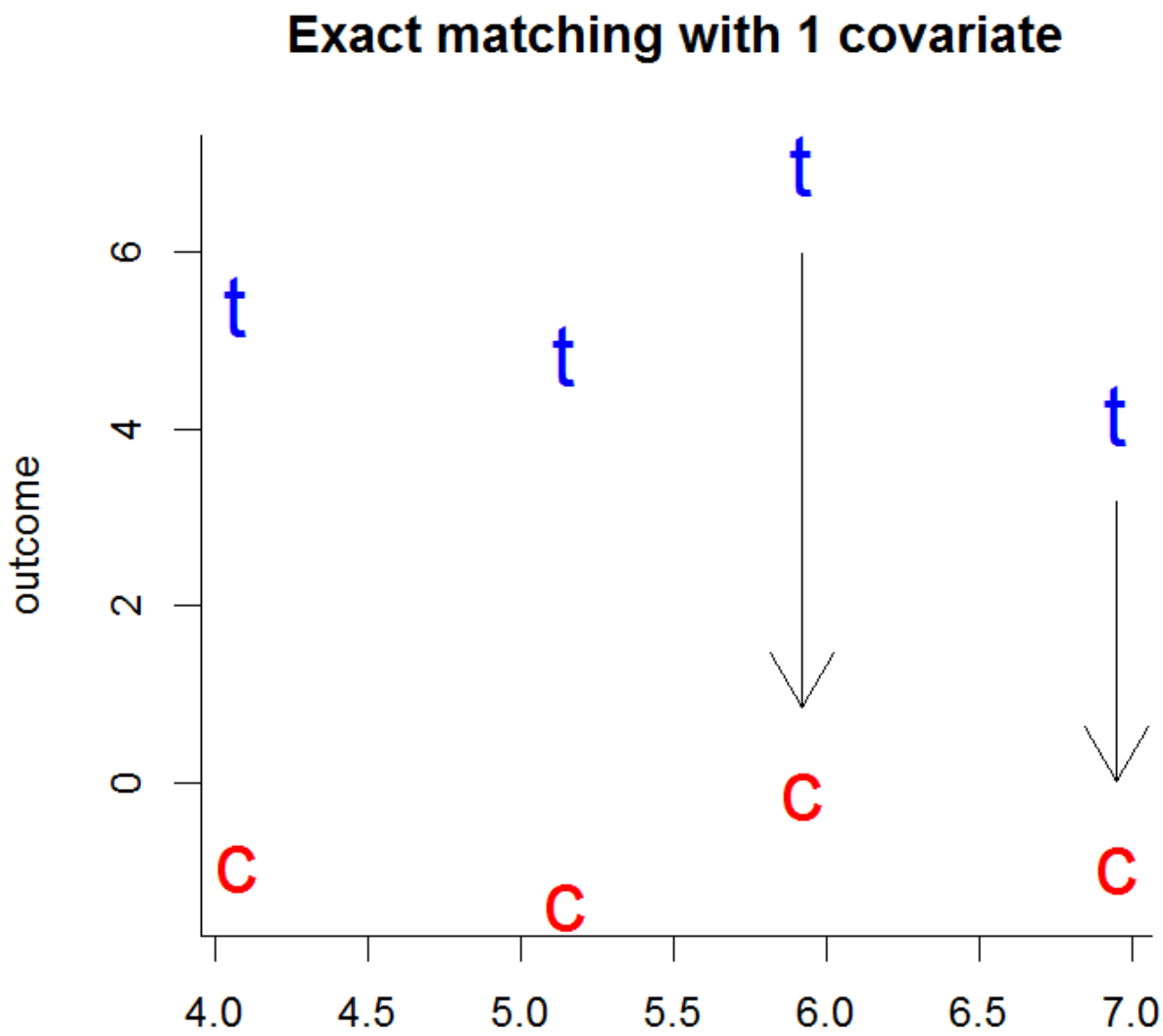
The most straightforward way to do this is by matching each treated unit to a control unit that have *exactly* the same covariate values.

This is called exact matching and can be thought of as the gold-standard for matching.

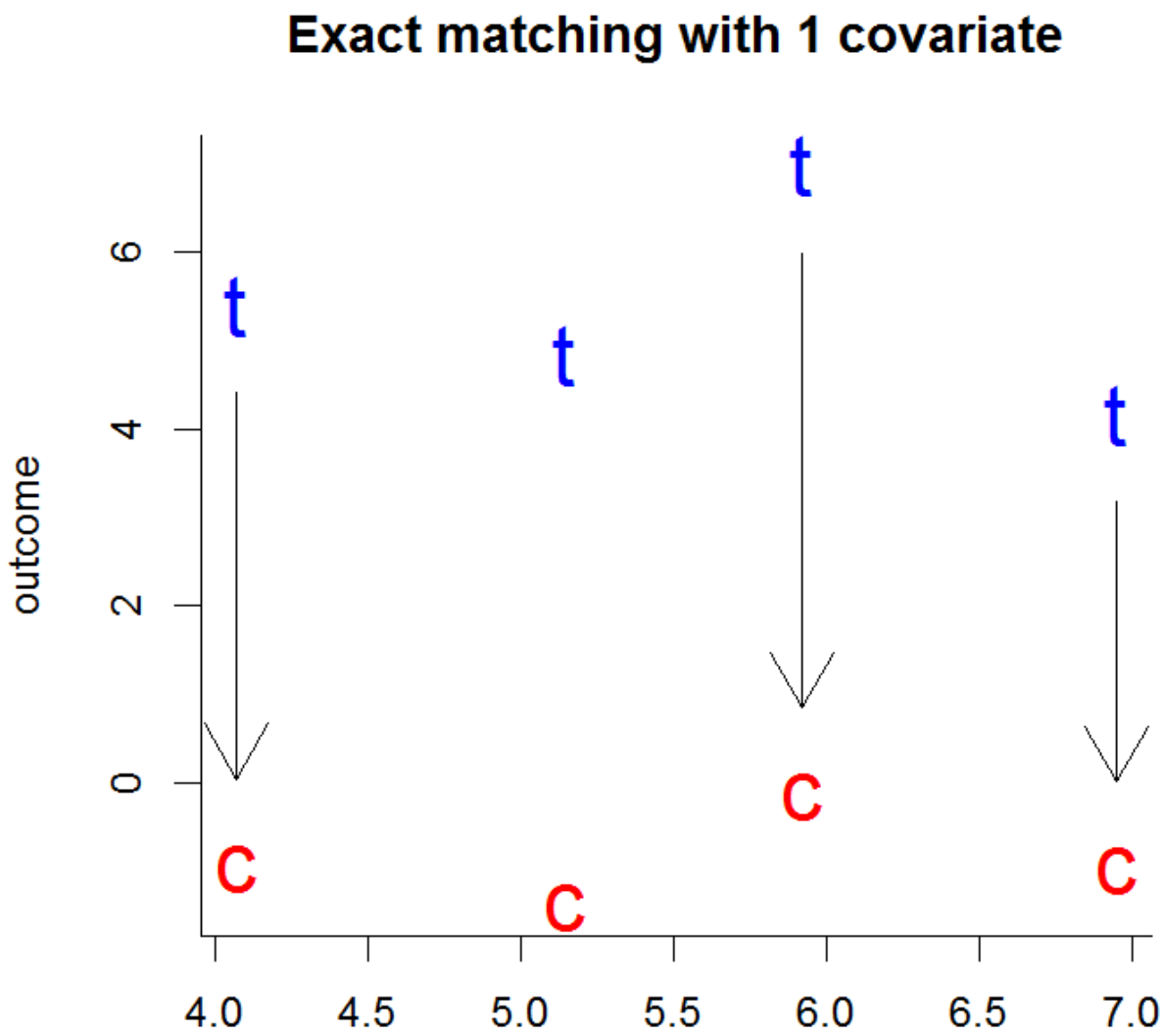
Exact matching with one covariate



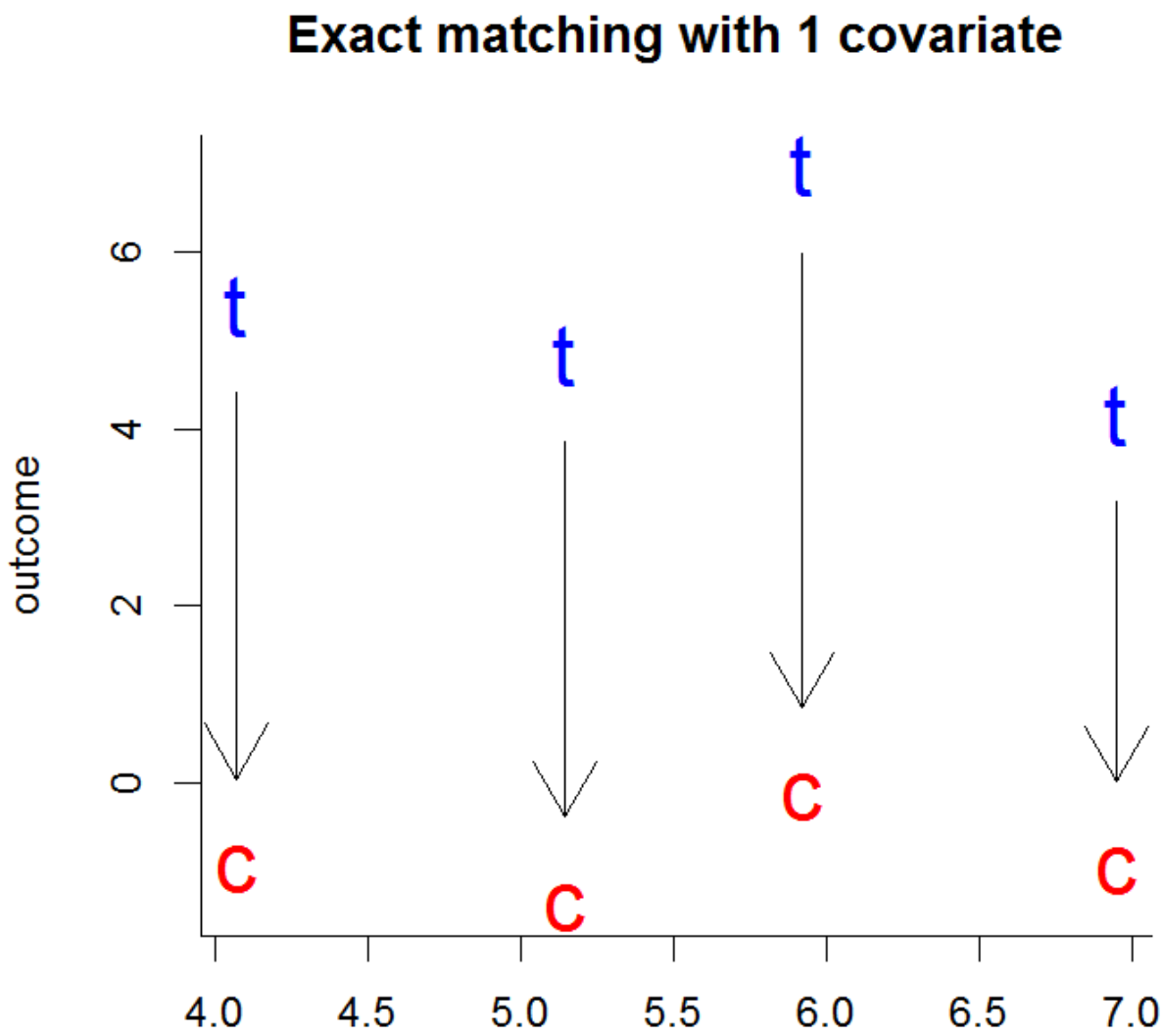
Exact matching with one covariate



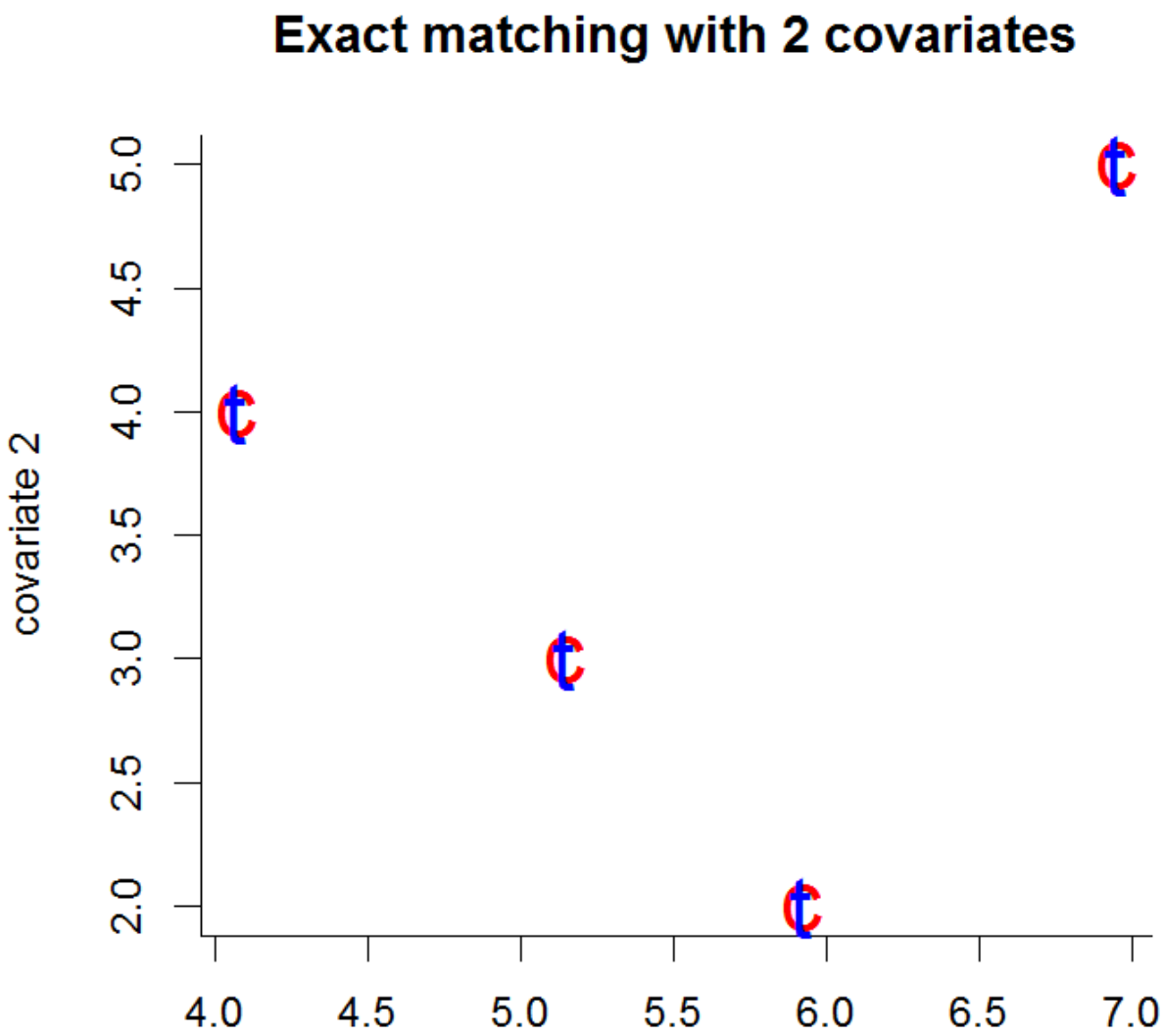
Exact matching with one covariate



Exact matching with one covariate



Exact matching with two covariates



Limitations of exact matching

When can we not do exact matching?

- When treatment/control units lack a perfect match in the other treatment/control condition (although we could prune these observations)
- More commonly, when you have continuous covariates, since the value of two observations can never be exactly the same

In these cases, we'll need to choose a metric for evaluating distance between units, and then use it to match

Distance metrics

There's lots of difference ways to measure distance, here are a few:

1. **Exact**:
   - Distance $= 0$ if $X_i = X_j$
   - Distance $= \infty$ if $X_i \neq X_j$
   - Ideal, but hard for a lot of variables

2. **Mahalanobis**:
   - Distance$(X_i, X_j) = \sqrt{(X_i - X_j)' S^{-1} (X_i - X_j)}$, where S^{-1} is the matrix of covariances between the variables
   - Doesn't work very well when X is high dimensional because it tries to take into account all interactions between the covariates.
   - You can emphasize the importance of a variable by tweaking the S^{-1} matrix

Distance metrics

3. **Propensity score**
   - Estimate a logit model where the outcome variable is whether the unit was in treatment or control group
   - Estimate $\pi_i \equiv \Pr(T_i = 1|X) = \frac{1}{1+e^{-X_i\beta}}$
   - Distance$(X_i, X_j) = |\pi_i - \pi_j|$
   - This overcomes the high-dimensionality problem by summarizing covariates with one number which is interpretable as the probability that the unit was in treatment group
   - Downside: doesn't ensure balance on your covariates, only on the propensity to be treated

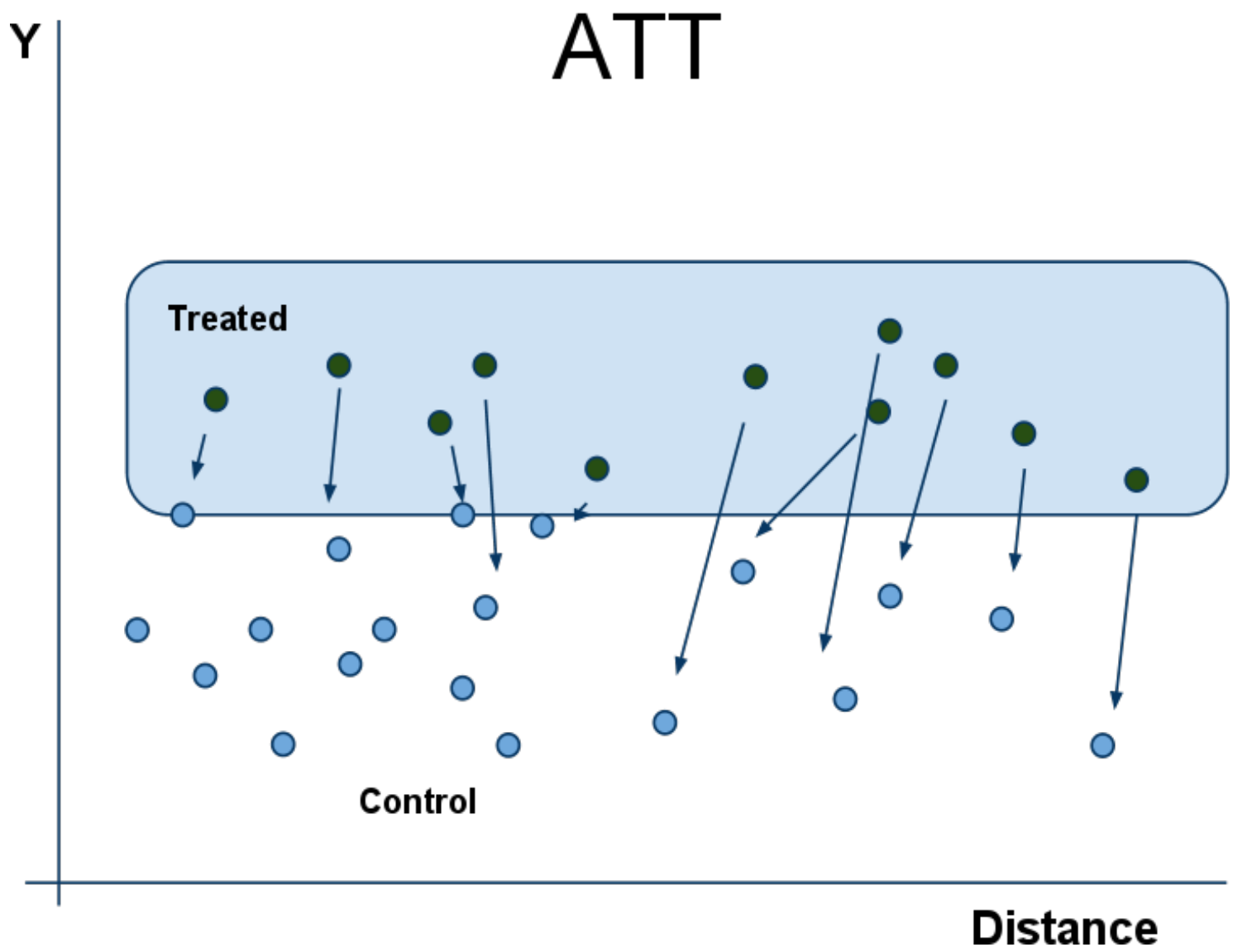
Distance ⇒ Matches

Once we have a distance metric, how do we determine matches?

1. 1:1 Nearest neighbor matching
   - Almost always is an estimate of the ATT – matches the treated group, then discards the remaining controls.
   - "Greedy": each control unit is only used once
2. Optimal matching
   - Instead of being greedy, minimizes a global distance measure
   - Reduces the difference between pairs, but not necessarily the difference between groups

Generally these methods give weights of zero or 1 to observations, depending on whether they are matched.

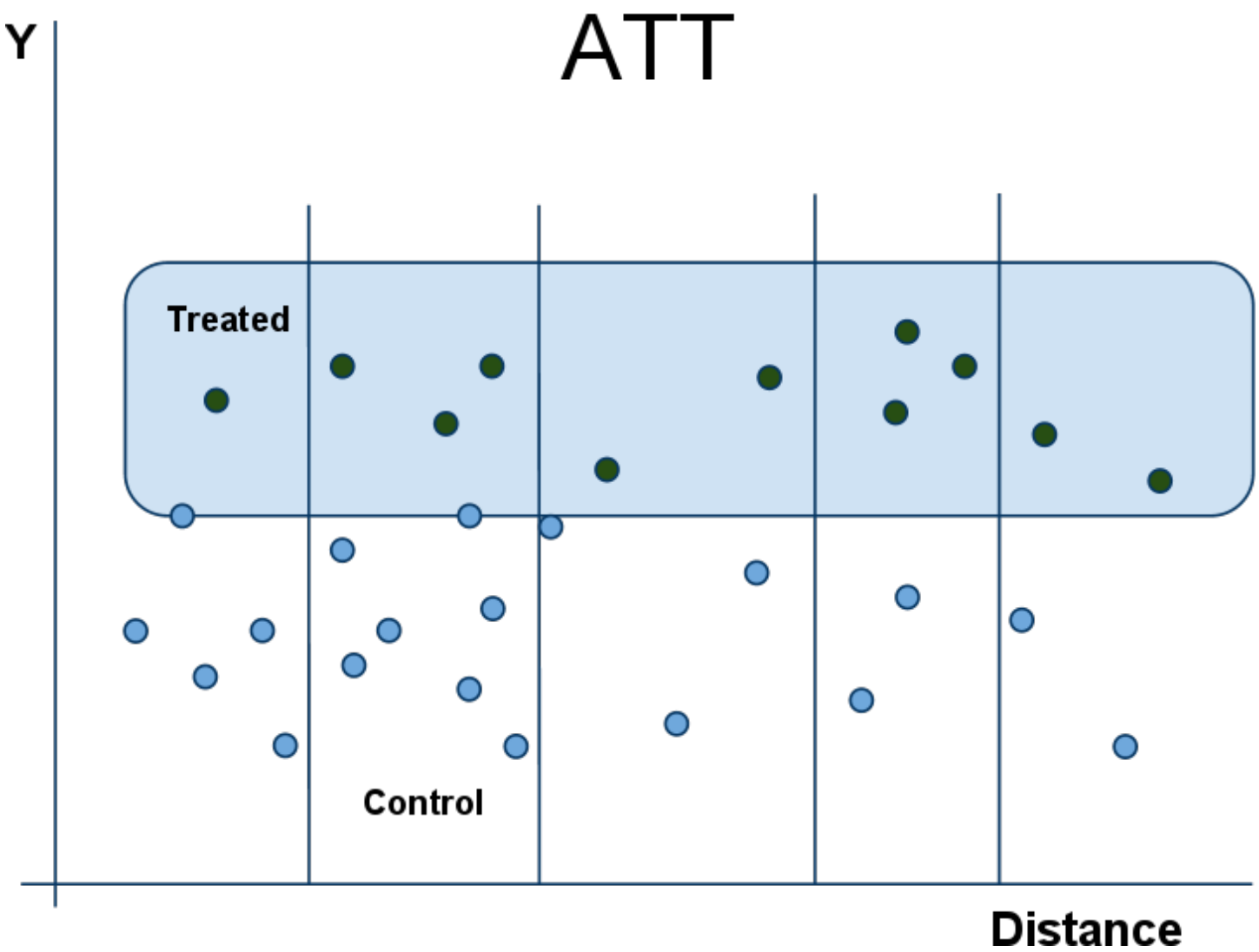
Visualizing Nearest Neighbor and Optimal Matches



Distance $\Rightarrow$ Matches

3. Full matching
   - Groups each treated unit with control units based on the distance metric
   - Estimates the ATT or ATE within each group, weighting each group by the number of observations.
4. Weighting based on propensity scores
   - Weighting based on propensity scores is essentially a subclassification scheme with extreme dimension reduction

Visualizing Subclassification



Distance in coarsened exact matching

- CEM is exact matching with coarsening.
- Similar to sub-classification, but the classification is not based on a distance metric, it's based on substantive knowledge of the covariates.
- This allows us to get the benefits of exact matching without the problems of high dimensionality.
- CEM also weights to get the ATT and the ATE depending on how many observations are in the coarsened categories.

Pros and Cons of 1 to 1 matching

- **Cons:**
  - You are discarding observations
  - Might lead to reduced power and bigger standard errors
- **Pros**:
  - You'll tend to get better matches
  - It might not lead to reduced power because power is often driven by the size of the smaller group (treated or control).
  - Power can be increased if you have better precision (reduced extrapolation)

Alternative to 1 to 1 matching: k-nearest neighbor matching. Match each treated unit to the k control units that are most similar, then average or weight over the potential outcomes of the control units.
Footnote: KNN matching is the basis for player evaluation metrics like PECOTA, KUBIAK, and VUKOTA in the sports analytics world.

Matching with or without replacement?

- **Pros of using replacement**:
  - You will get better matches
  - Particularly helpful when you have very few control individuals
- **Cons of using replacement:**
  - More complicated because matched controls are not independent.
  - Should be aware of how many times you are using one control.

Outline

The Goal of Balance

To what extent does the pre-match distribution of $X|T = 1$ look like the distribution of $X|T = 0$?

If they are very close, then we have matched well. Example: exact matching leads to identical multivariate distributions:

$$f(X|T = 1) = f(X|T = 0)$$

Balance tables

Papers that do causal inference or present results of experiments often present a **balance table**

Usually shows summary statistics of covariates separated out by control or treatment groups

Example from Gerber, Green, Larimer (2008):

TABLE 1. Relationship between Treatment Group Assignment and Covariates (Household-Level Data)

	Control	Civic Duty	Hawthorne	Self	Neighbors
	Mean	Mean	Mean	Mean	Mean
Household size	1.91	1.91	1.91	1.91	1.91
Nov 2002	.83	.84	.84	.84	.84
Nov 2000	.87	.87	.87	.86	.87
Aug 2004	.42	.42	.42	.42	.42
Aug 2002	.41	.41	.41	.41	.41
Aug 2000	.26	.27	.26	.26	.26
Female	.50	.50	.50	.50	.50
Age (in years)	51.98	51.85	51.87	51.91	52.01
N =	99,999	20,001	20,002	20,000	20,000

Note: Only registered voters who voted in November 2004 were selected for our sample. Although not included in the table, there were no significant differences between treatment group assignment and covariates measuring race and ethnicity.

Balance tables

Often balance tables also include information about the variance of the covariates. You can use this information to do a difference in means t-test.

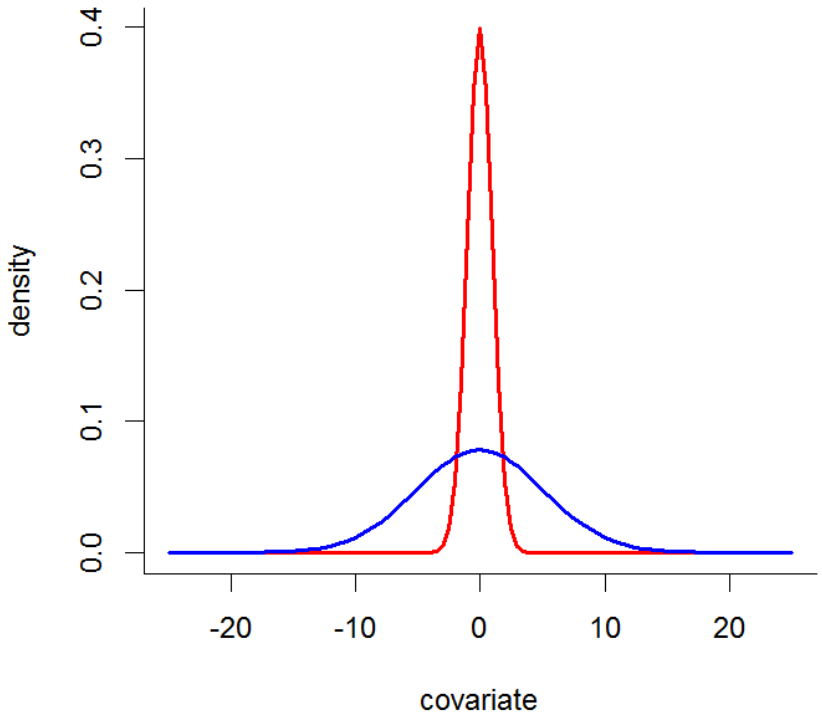
In R:

```
t.test(data$covariate1[data$treatment == 1],
       data$covariate1[data$treatment == 0])
```

What does a statistically significant result from this t-test indicate?

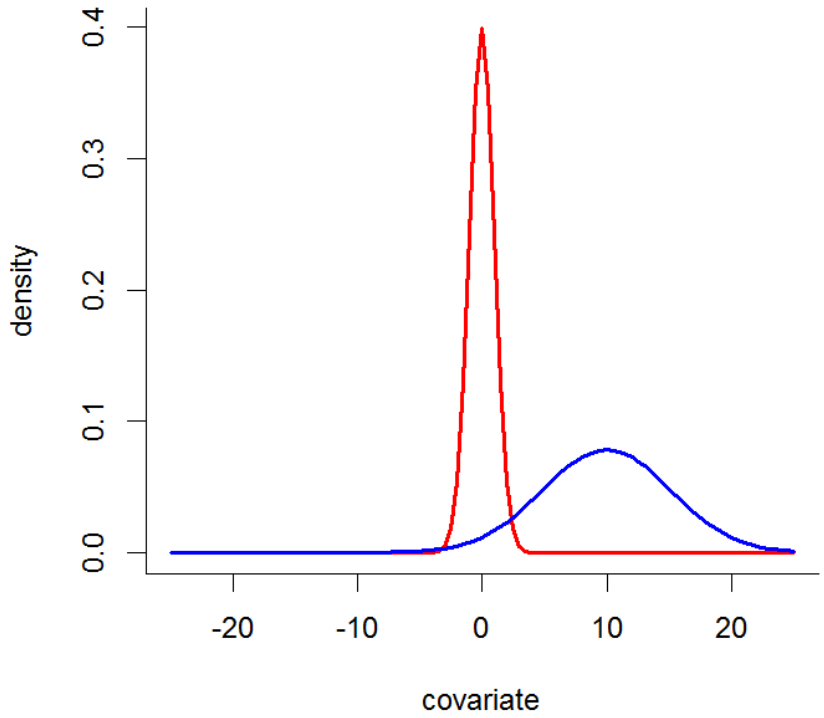
Shortcomings of balance tables

What's wrong with this picture?



Shortcomings of balance tables

What's wrong with this picture?



Shortcomings of balance tables

This looks balanced, right?

	number of treatment units	number of control units
black	2.00	2.00
white	2.00	2.00
female	2.00	2.00
male	2.00	2.00

Shortcomings of balance tables

It looks like it's balanced, but it's definitely not:

Treated Units

	female	male
black	2.00	0.00
white	0.00	2.00

Control Units

	female	male
black	0.00	2.00
white	2.00	0.00

Full dataset

	race	sex	treat
1	black	m	1.00
2	black	m	1.00
3	white	f	1.00
4	white	f	1.00
5	black	f	0.00
6	black	f	0.00
7	white	m	0.00
8	white	m	0.00

Multivariate balance: $\mathcal{L}_1$

The idea is to divide the distributions of $X|T = 1$ and $X|T = 0$ each into k bins, sort of like a big multivariate (or univariate) histogram. Bin sizes are usually determined automatically.

We then have a set of frequencies $f_1, ..., f_k$ where f_i is the proportion of treated observations which fall in bin i; likewise $g_1, ..., g_k$ are the proportions of control observations falling in bin i.

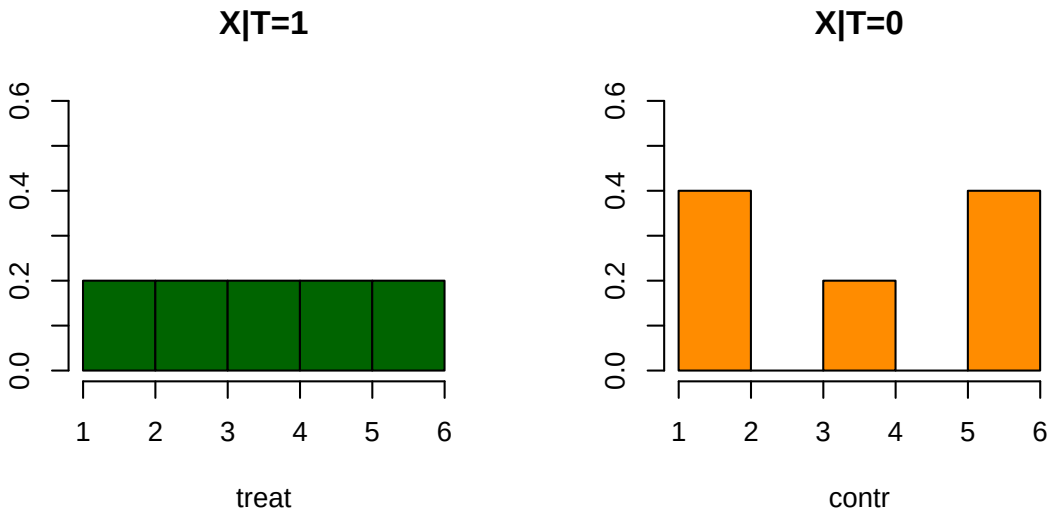
Then

$$\mathcal{L}_1(f, g) = \frac{1}{2} \sum_{i=1,...,k} |f_i - g_i|.$$

If balance is perfect this equals 0; if completely imperfect, 1.

Multivariate balance: $\mathcal{L}_1$

Here is a univariate example:



$$L_1(f, g) = \frac{1}{2}(.2 + .2 + 0 + .2 + .2) = .4.$$

Outline

Introducing the Data

LaLonde dataset: an evaluation of a job training program administered in 1976. The data contain a few hundred observations which were part of a randomized experiment, as well as several thousand (non-randomized, control) observations which were drawn from the CPS. Main outcome of interest is `re78`, retained earnings in 1978; sole treatment is the job training program (`treated`).

A variety of covariates on which to match:

- `age`, `education` (in years), `nodegree`
- `black`, `hispanic`, `married`
- `re74`, `re75`
- `u74`, `u75` both indicators of unemployment

Get the Data

```
install.packages("MatchIt")
install.packages("cem")
library(MatchIt)
library(cem)
library(Zelig)

## load the dataset from cem package
data(LL)
```

Look at the Data Before Matching

Naive calculation of the average treatment effect:

```
mean(LL$re78[LL$treated == 1]) - mean(LL$re78[LL$treated == 0])
[1] 886.3038
```

Estimation of ATE using regression:

```
summary(lm(re78 ~ treated + age + education + black + married +
        nodegree + re74 + re75 + hispanic + u74 + u75, data = LL))
```

	Estimate	Std. Error	t value	Pr(> |t|)
(Intercept)	3185.6806	2637.8616	1.21	0.2276
treated	823.6546	468.4621	1.76	0.0791
age	10.2419	37.0664	0.28	0.7824
education	200.5936	180.5211	1.11	0.2669
black	-1419.1896	801.9466	-1.77	0.0772
married	48.5436	652.2012	0.07	0.9407
nodegree	-299.6107	747.9791	-0.40	0.6889
re74	0.1274	0.0753	1.69	0.0909
re75	0.0647	0.0914	0.71	0.4794
hispanic	299.2882	1050.4602	0.28	0.7758
u74	1529.9294	937.8173	1.63	0.1033
u75	-1005.8379	917.0246	-1.10	0.2731

Checking (univariate) Imbalance in R

The most straightforward way to check univariate imbalance is to perform a t-test on the difference in for the covariate of interest between the treatment and control groups.

```
test <- t.test(LL$age[LL$treat == 1], LL$age[LL$treat == 0])

Welch Two Sample t-test

data:  LL$age[LL$treat == 1] and LL$age[LL$treat == 0]
t = 0.3565, df = 631.223, p-value = 0.7216
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.807995  1.166403
sample estimates:
mean of x mean of y
 24.62626  24.44706
```

Checking (multivariate) Imbalance in R

```
pre.imbalance <- imbalance(group=LL$treated,
                           data=LL,
                           drop=c("treated","re78"))

Multivariate Imbalance Measure: L1=0.735
Percentage of local common support: LCS=12.4%

Univariate Imbalance Measures:

               statistic   type           L1 min 25%       50%
age         1.792038e-01 (diff) 4.705882e-03   0   1   0.00000
education   1.922361e-01 (diff) 9.811844e-02   1   0   1.00000
married     1.070311e-02 (diff) 1.070311e-02   0   0   0.00000
...
re74       -1.014862e+02 (diff) 5.551115e-17   0   0  69.73096
re75        3.941545e+01 (diff) 5.551115e-17   0   0 294.18457
...
```

Exact Matching

```
exact.match <- matchit(formula= treated ~ age + education
    + black + married + nodegree + re74 + re75 + hispanic +
    u74 + u75, data = LL, method = "exact")


Call:
matchit(formula = treated ~ age + education + black + married +
    nodegree + re74 + re75 + hispanic + u74 + u75, data = LL,
    method = "exact")

Exact Subclasses: 36

Sample sizes:
          Control Treated
All           425     297
Matched        74      55
Unmatched     351     242
```

Exact Matching

Look at the matched dataset:

```
exact.data <- match.data(exact.match)
```

```
head(exact.data)
          u74 u75   weights subclass
16110   1   1 1.0000000        1
16141   1   1 1.0000000        2
... 16148   1   1 1.0000000        3
16156   1   1 0.6727273        3
16164   1   1 1.0000000        4
16182   1   1 4.0363636       14
```

What is the Treatment Effect?

Using regression:

```
lm(re78 ~ treated+ age + education + black
          + married + nodegree + re74 + re75
          + hispanic + u74 + u75,
    data = exact.data,
    weights=exact.data$weights)
```

Selected output:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    3951.7      670.1   5.897 3.14e-08 ***
treated        1306.1     1026.2   1.273    0.205
```

What is the Treatment Effect?

Using the formula estimator for the ATE:

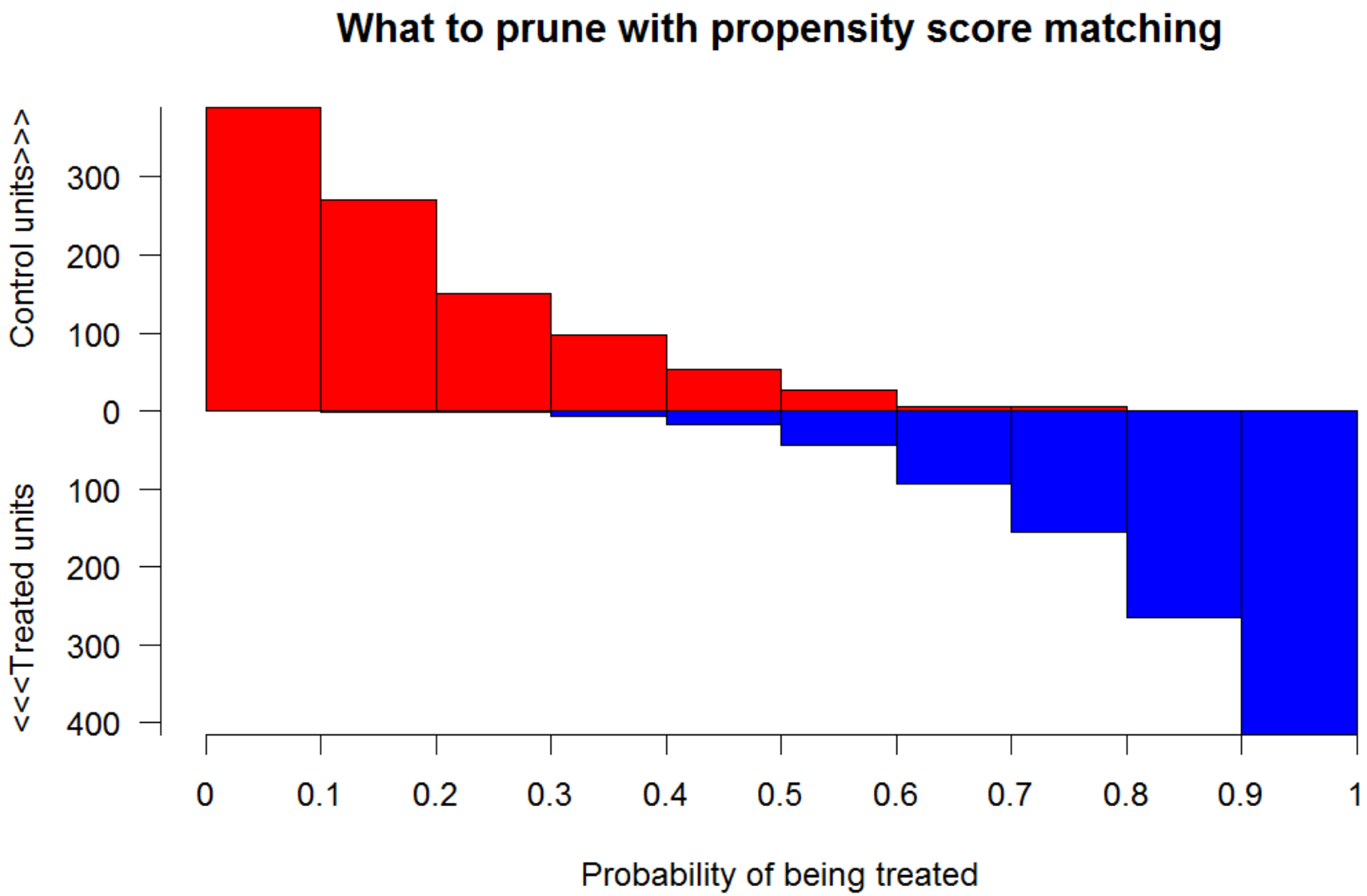
```
y.treat <-
weighted.mean(exact.data$re78[exact.data$treated == 1],
              exact.data$weights[exact.data$treated == 1])


y.cont <-
weighted.mean(exact.data$re78[exact.data$treated == 0],
              exact.data$weights[exact.data$treated == 0])


y.treat - y.cont
[1] 1306.075
```

Propensity score matching

The idea with propensity score matching is that we use a logit model to estimate the probability that each observation in our dataset was in the treatment or control group.

Then we use the predicted probabilities to prune out dataset such that, for every treated unit, there's a control unit that can serve as a viable counterfactual.

Pruning based on propensity scores



Calculating propensity scores

The model:

$$T_i \sim Bern(\pi_i)$$

$$\pi_i = \frac{1}{1 + e^{-X_i\beta}}$$

Estimate our coefficients using `glm()` or `zelig()`:

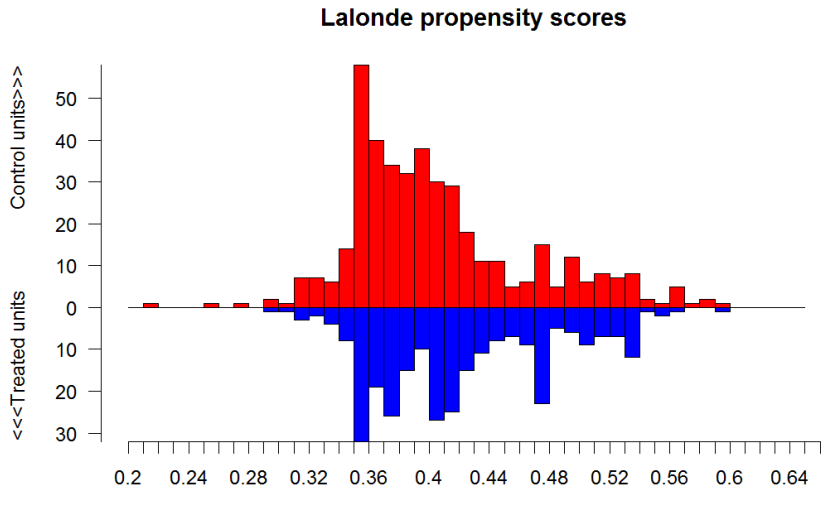
```
pscores.logit <- glm(treated ~ age + education + black
                     + married + nodegree + re74 + re75
                     + hispanic + u74 + u75,
                     family = "binomial",
                     data = LL)
```

Calculating propensity scores

Get the propensity score for each observation, which are the same as the predicted probabilities, π_i:

```
fittedvalues <- pscores.logit$fitted
pscore.treat <- fittedvalues[LL$treated == 1]
pscore.control <- fittedvalues[LL$treated == 0]
```

Determine what observations should be pruned by comparing the overlap in the propensity scores for the treated and control groups:



So far we've looked at how you can use propensity scores to prune your data, but we haven't looked at matching using propensity scores.

Let's return go back to looking at the full dataset and see how to do that

Nearest neighbor matching with propensity scores

```
nearest.match <- matchit(formula = treated ~ age + education
       + black + married + nodegree + re74 + re75 + hispanic +
         u74 + u75, data = LL,
         method = "nearest",
         distance = "logit",
         discard="control")
```

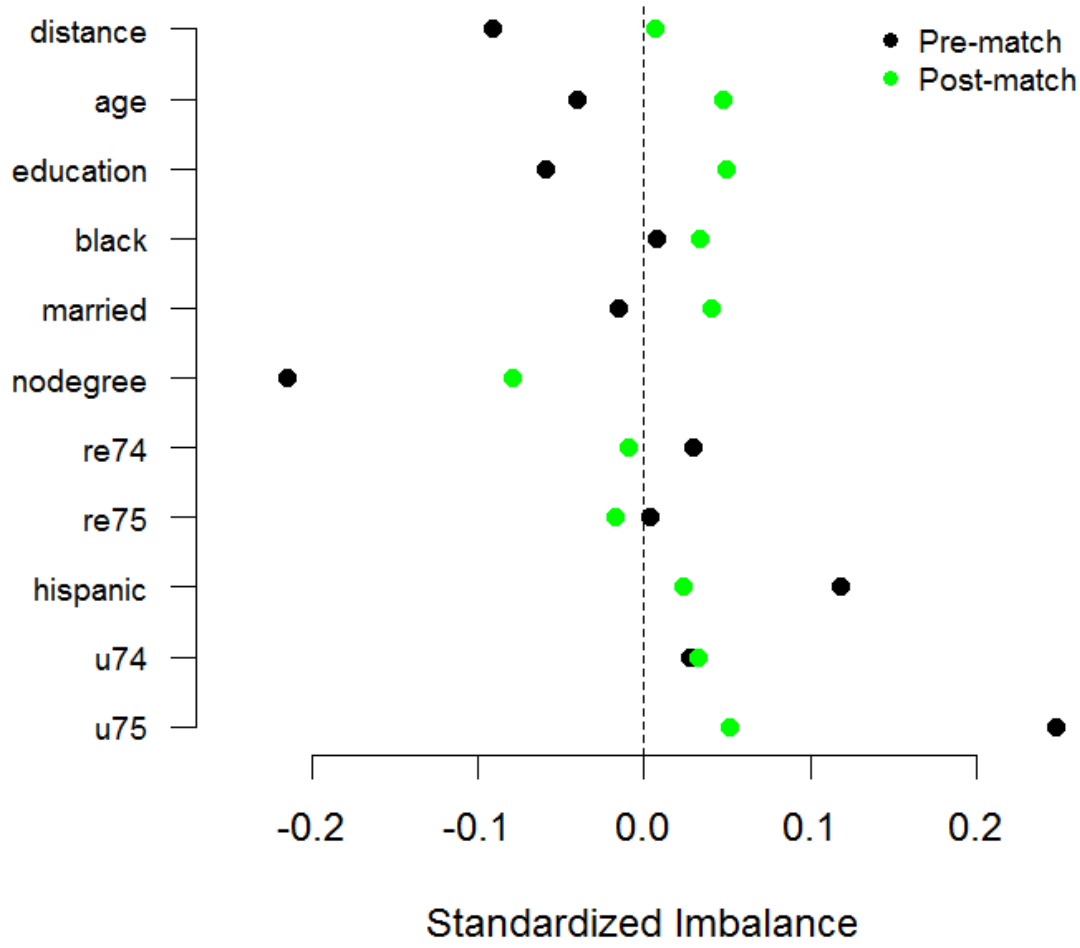
Check balance post-matching:

```
data.matched <- match.data(nearest.match)
imbalance(group=data.matched$treated, data=data.matched,
          drop=c("treated", "re78", "distance", "weights"))
```

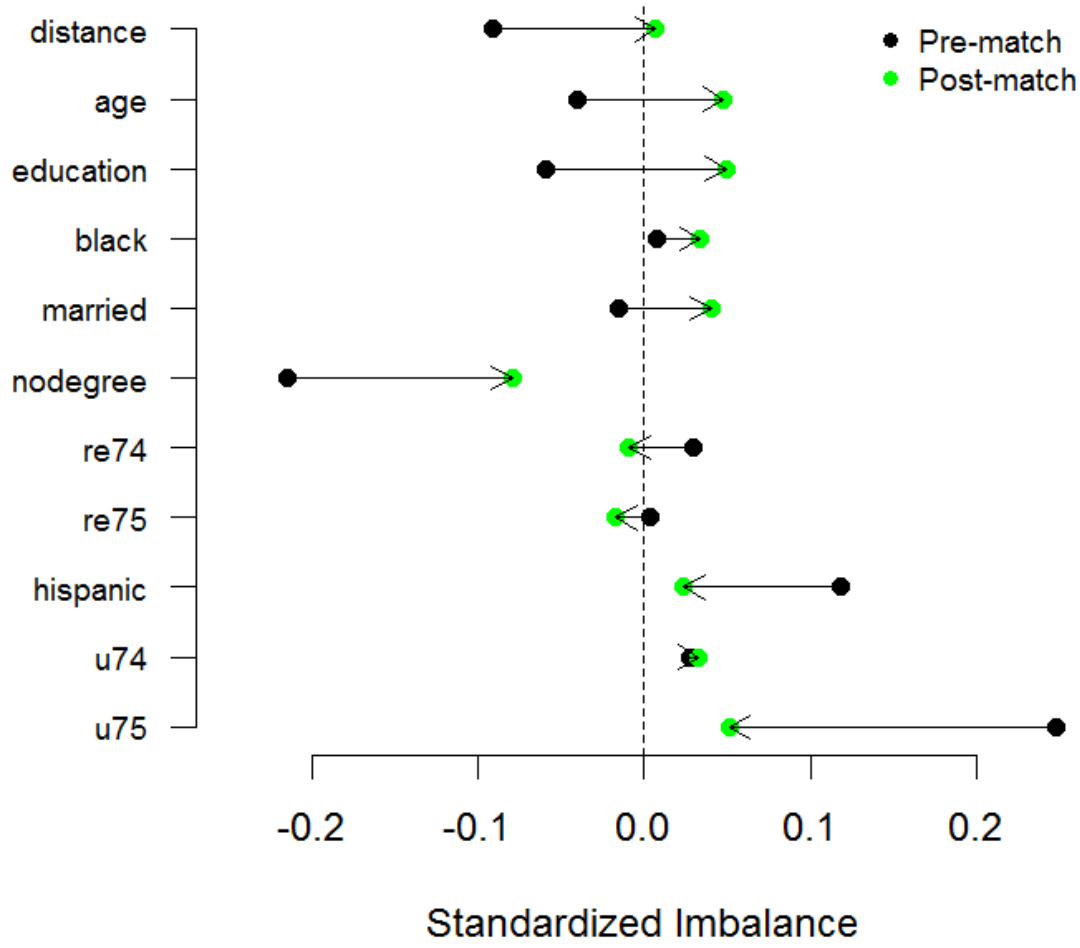
OR

```
pre.balance <- summary(nearest.match)$sum.all
post.balance <- summary(nearest.match)$sum.match
```

Balance checking



Balance checking



Estimating the ATT

```
nearest.data <- match.data(nearest.match)


## non-parametric estimate of the ATT
mean(nearest.data$re78[nearest.data$treated == 1]) -
     mean(nearest.data$re78[nearest.data$treated == 0])
[1] 1042.897

## A model-based estimate of the ATT
nearest.model <- lm(re78 ~ treated + age + education + black
  + married + nodegree + re74 + re75 + hispanic + u74 + u75,
  data = nearest.data)
```

Mahalanobis Matching

Implemented in exactly the same way as propensity score matching in R, except you'll use the `distance = ''mahalanobis''` option when you run `matchit()`

CEM: Automatic Coarsening

```
auto.match <- matchit(formula = treated ~ age + education
  + black + married + nodegree + re74 + re75 + hispanic +
  u74 + u75, data = LL, method = "cem")
```

```
Call:
matchit(formula = treated ~ age + education + black + married +
    nodegree + re74 + re75 + hispanic + u74 + u75, data = LL,
    method = "cem")

Sample sizes:
          Control Treated
All           425     297
Matched       222     163
Unmatched     203     134
Discarded       0       0
```

CEM: User Coarsening

```
re74cut <- seq(0, 40000, 5000)
re75cut <- seq(0, max(LL$re75)+1000, by=1000)
agecut <- c(20.5, 25.5, 30.5,35.5,40.5)


my.cutpoints <- list(re75=re75cut, re74=re74cut, age=agecut)


user.match <-  matchit(treated ~ age + education + black + married
                       + nodegree + re74 + re75 + hispanic + u74
                       + u75,
                       data = LL,
                       method = "cem",
                       cutpoints = my.cutpoints)
```

CEM: User Coarsening

```
user.data <- match.data(user.match)
auto.data <- match.data(auto.match)


auto.imb <- imbalance(group=auto.data$treated,
                      data=auto.data,
                      drop=c("treated","re78","distance",
                             "weights","subclass"))


user.imb <- imbalance(group=user.data$treated,
                      data=user.data,
                      drop=c("treated","re78","distance",
                             "weights","subclass"))
```

Balance checking

```
auto.imb$L1

Multivariate Imbalance Measure: L1=0.592
Percentage of local common support: LCS=25.2%


user.imb$L1

Multivariate Imbalance Measure: L1=0.437
Percentage of local common support: LCS=43.1%
```

CEM: Compare the Two

```
summary(auto.match)$nn
           Control Treated
All            425     297
Matched        222     163
Unmatched      203     134
Discarded        0       0
```

```
summary(user.match)$nn
           Control Treated
All            425     297
Matched        182     136
Unmatched      243     161
Discarded        0       0
```

CEM: Causal Effects

```
cem.match <- cem(treatment = "treated",
                 data = LL, drop = "re78",
                 cutpoints = my.cutpoints)


cem.match.att <- att(obj=cem.match, formula=re78 ~ treated,
                     data = LL, model="linear")


            G0  G1
All        425 297
Matched    182 136
Unmatched 243 161

Linear regression model on CEM matched data:

SATT point estimate: 448.556610 (p.value=0.447836)
95% conf. interval: [-708.263075, 1605.376295]
```

CEM: Causal Effects with a Model

```
cem.match.att2 <- att(obj=cem.match2, formula=re78 ~ treated +
                        age + education + black + married +
                        nodegree + re74+  re75 + hispanic +
                        u74 + u75,
                      data = LL, model="linear")
```

```
            G0  G1
All        425 297
Matched    182 136
Unmatched 243 161

Linear regression model on CEM matched data:

SATT point estimate: 474.936852 (p.value=0.423551)
95% conf. interval: [-686.678239, 1636.551944]
```

Outline

Intuition for the matching frontier

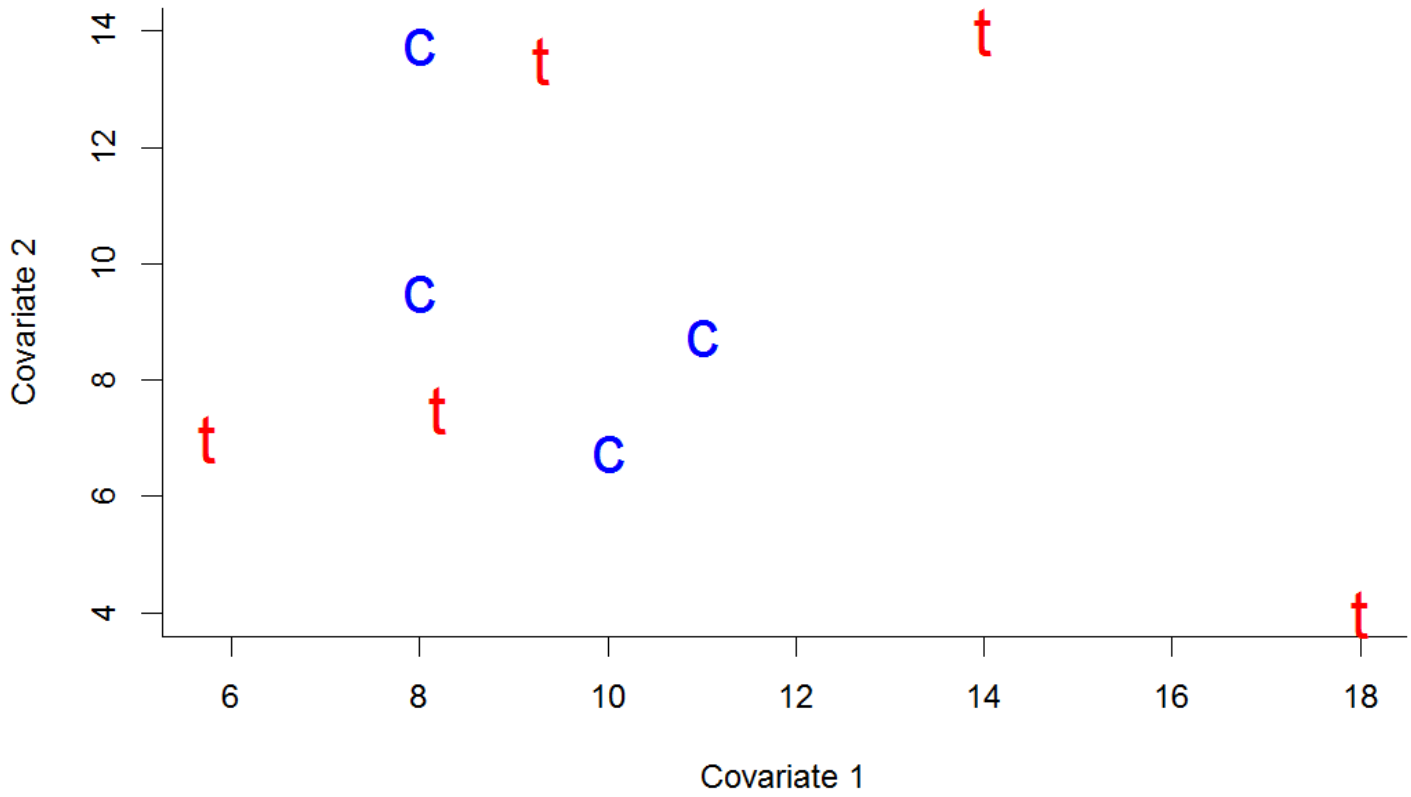
When you prune data by matching, sample size and imbalance trade off with each other

If you don't prune anything, you'll have a big sample size (allowing for better precision) but your data is likely to be very imbalanced and your estimates might be biased

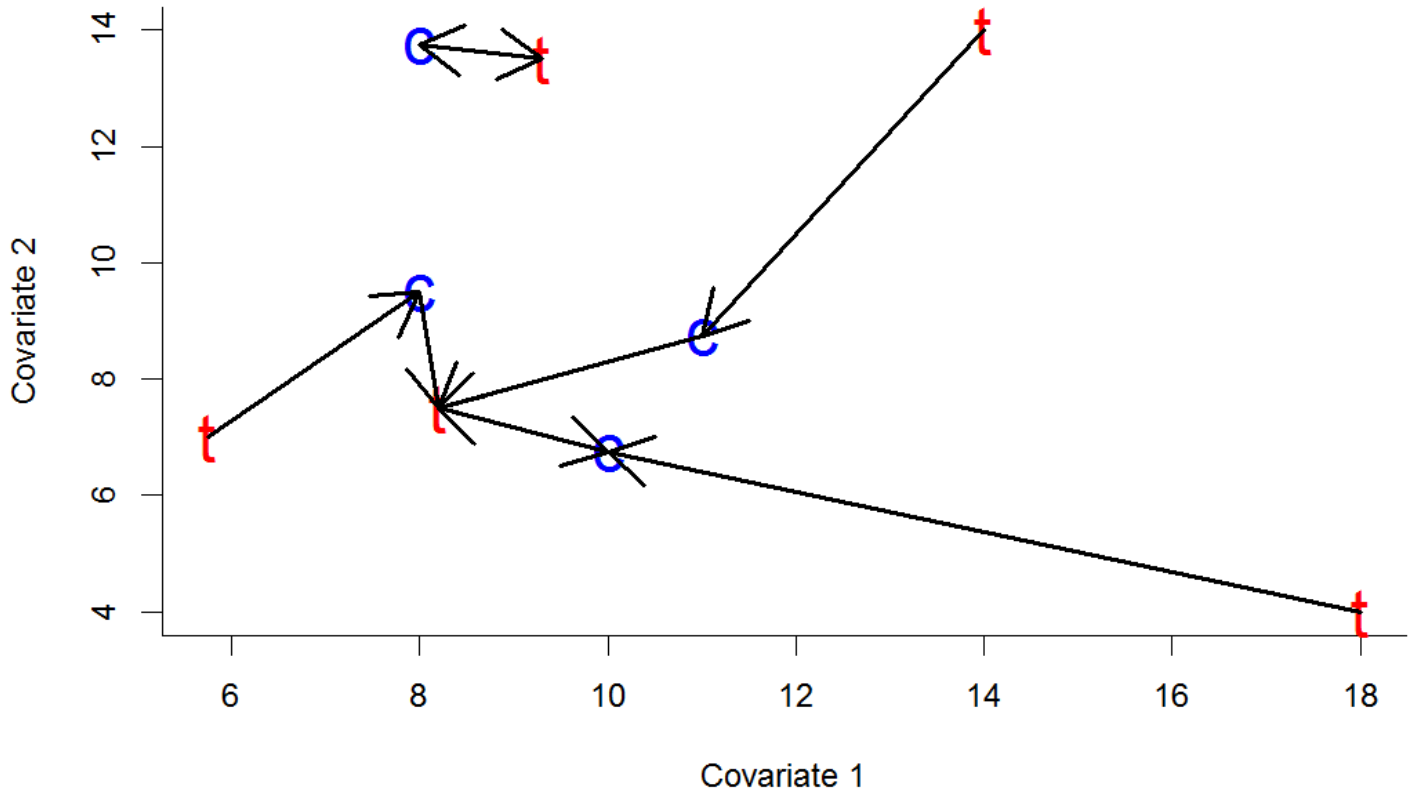
If you prune everything, you'll have perfect balance but you'll have zero observations left to calculate your effects

King, Lucas, and Nielsen (2014) provide a way to understand and visualize all the

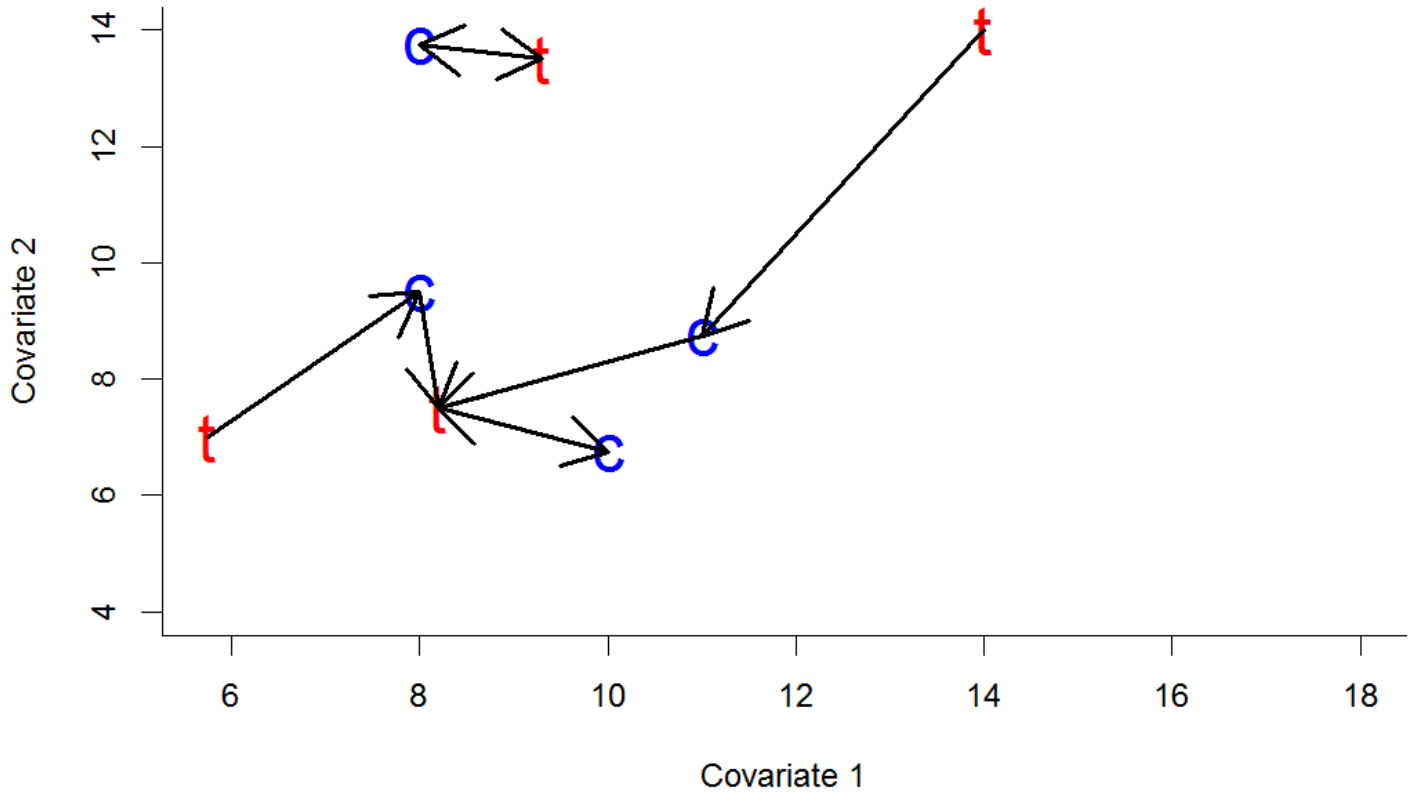
An example


Covariate 2
14
12
10
8
6
4
c
t
t
c
c
t
t
c
t
6
8
10
12
14
16
18
Covariate 1

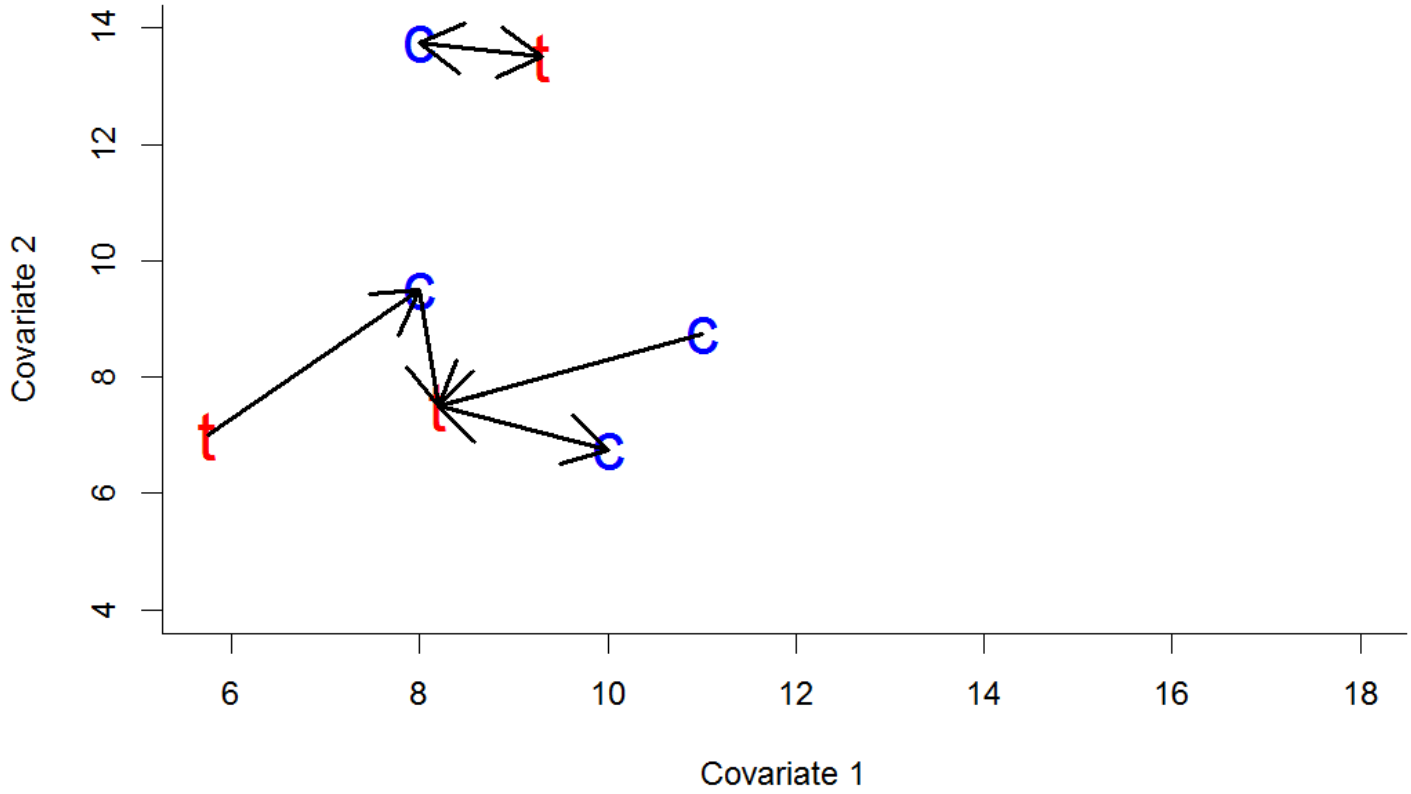
An example



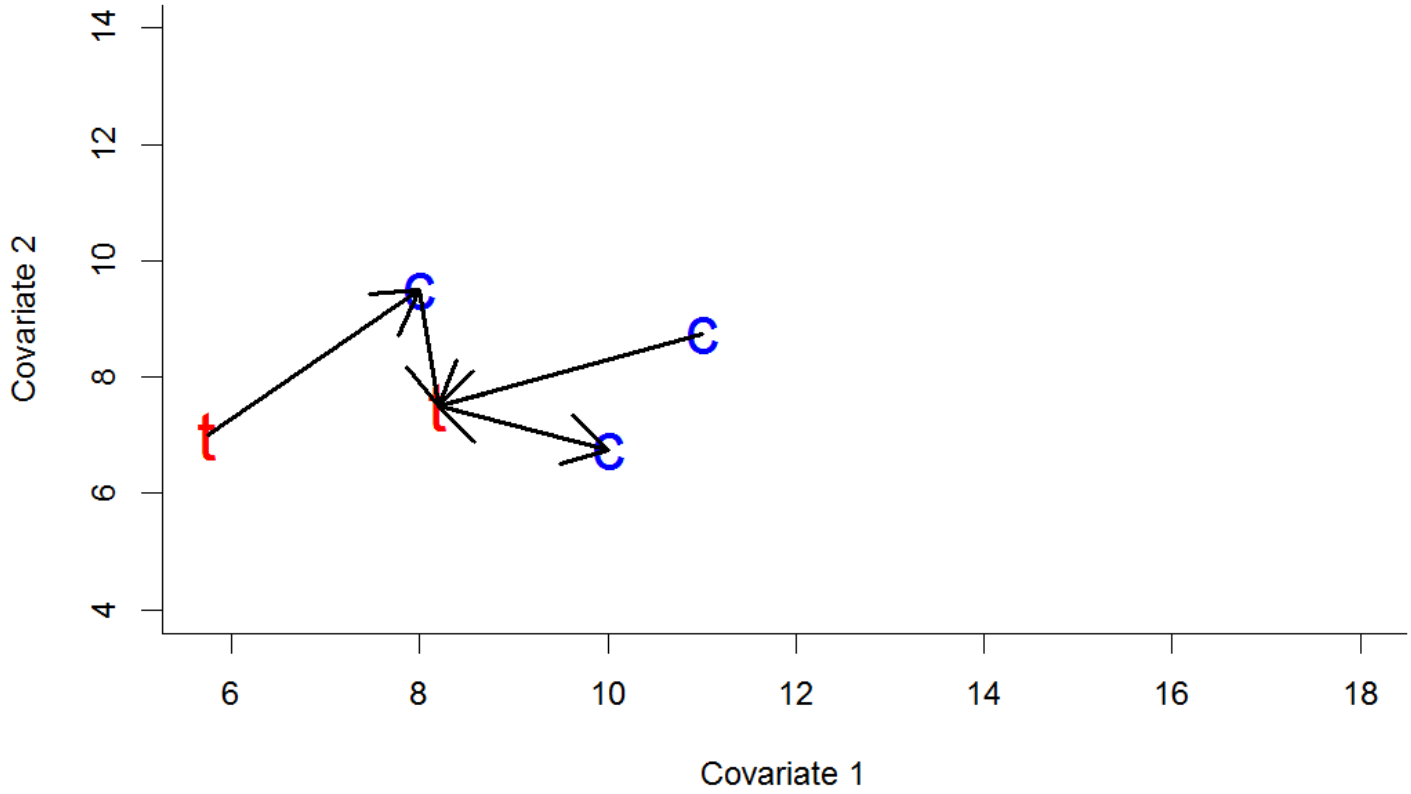
An example



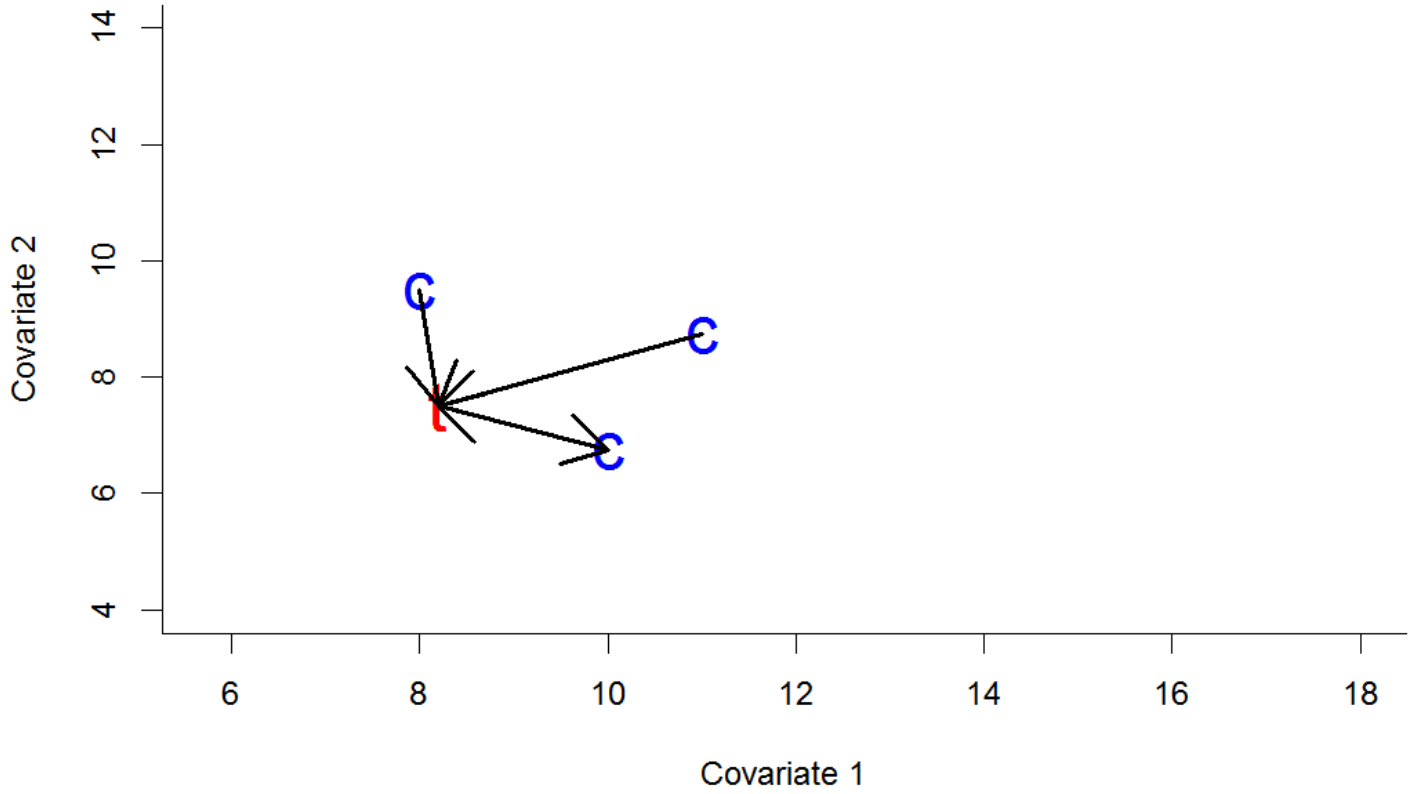
An example



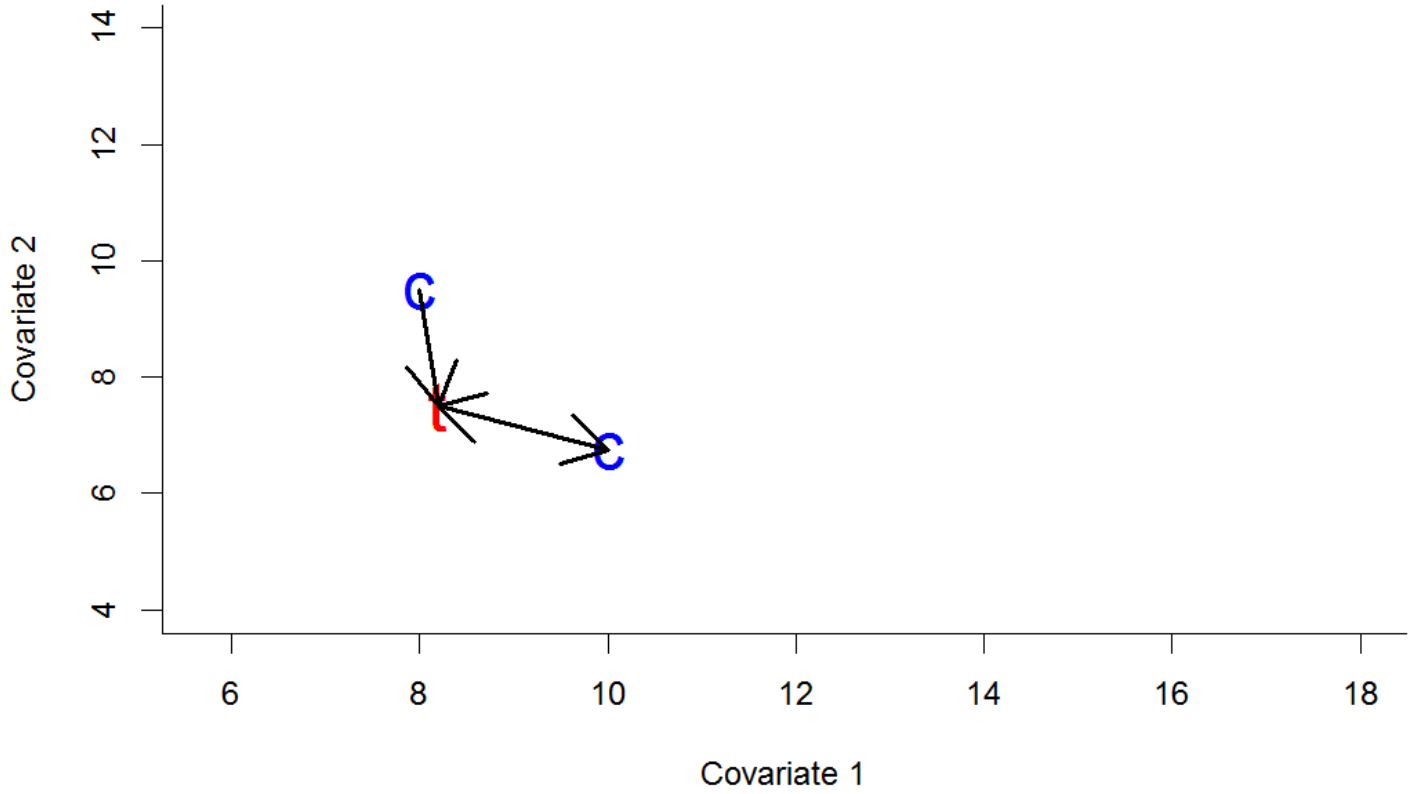
An example



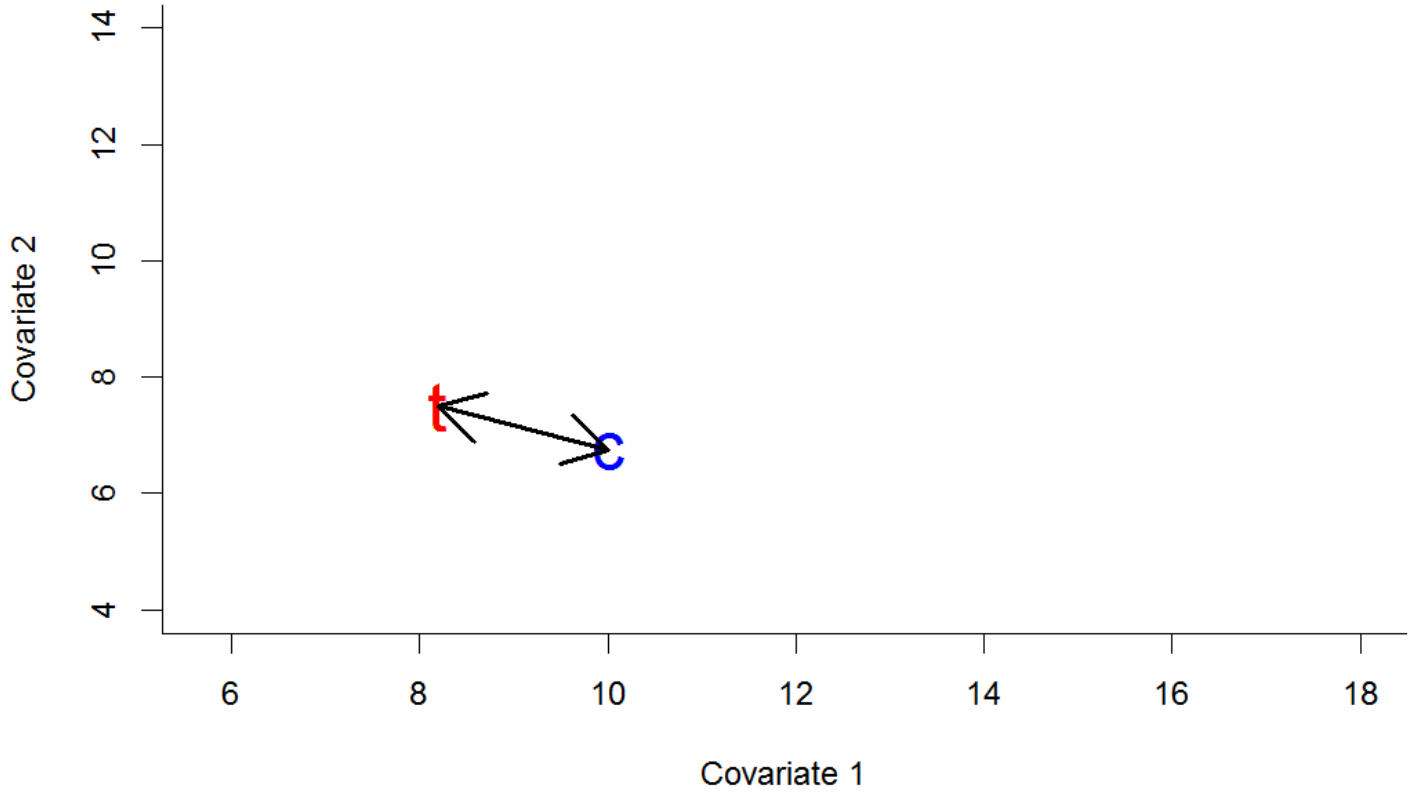
An example



An example



An example



Calculating the frontier

The first thing we'll want to do is get a matched dataset for every point along the frontier.

Do this by using the `makeFrontier()` function:

```
install.packages("MatchingFrontier")
library(MatchingFrontier)
match.variables <- names(LL)[!names(LL) %in% c("treated","re78")]
our.frontier <- makeFrontier(dataset = LL,
                             treatment = "treated",
                             outcome = "re78",
                             match.on = match.variables)
```

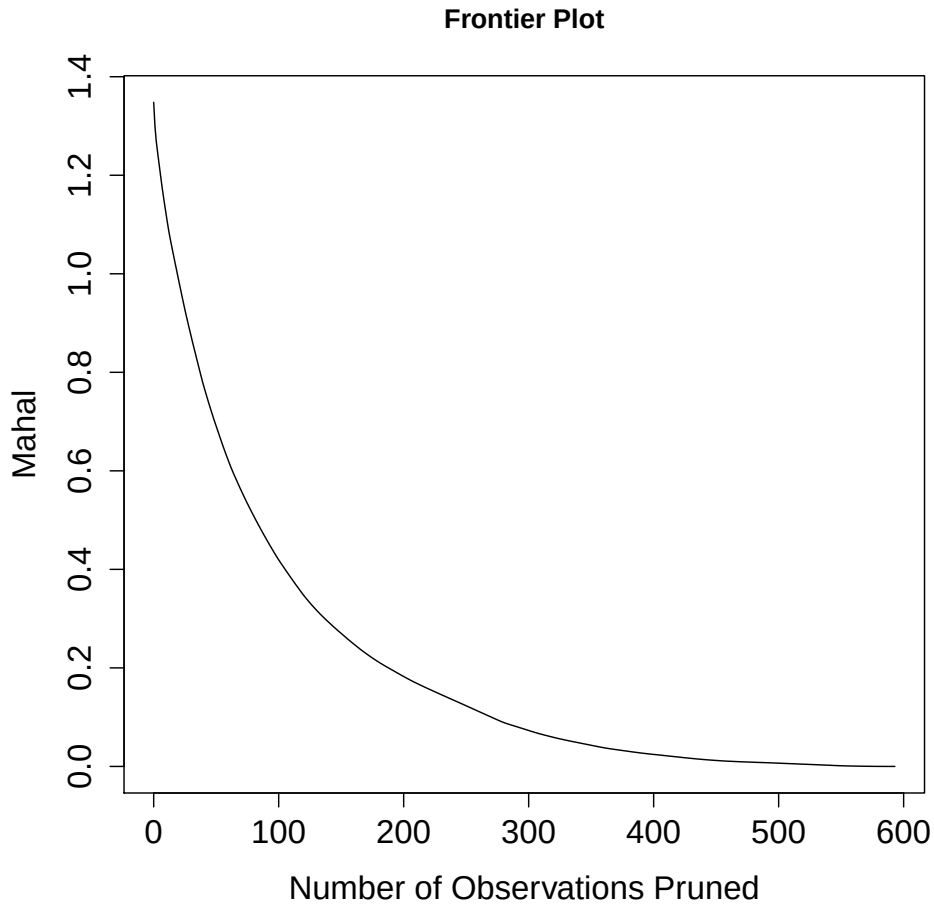
Get the causal effects

You've now got matched datasets for each possible sample size. Let's calculate the FSATT in each one:

```
myests <- frontierEst(myfrontier,
                      mydataset,
                      myform = formula('re78 ~ treated'),
                      treatment = mytreatment)
```

Look at the results

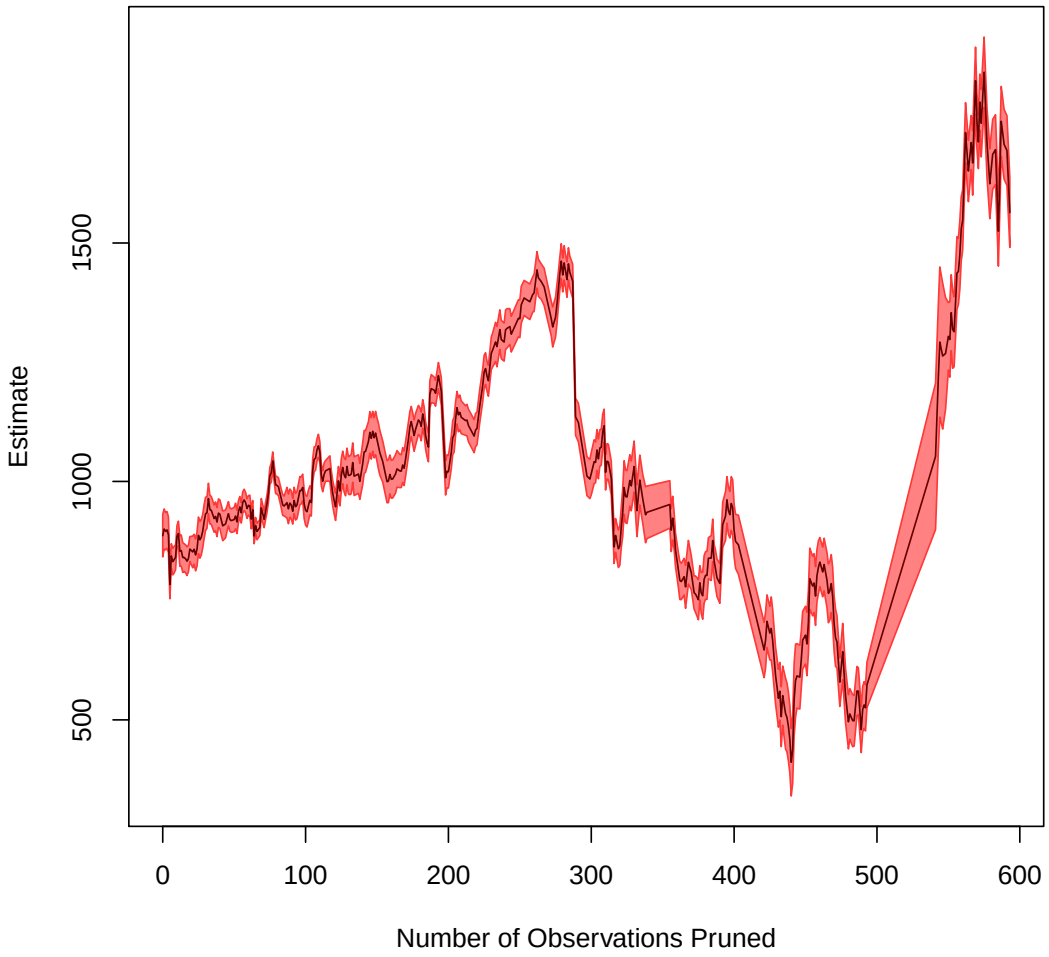
```
plotFrontier(our.frontier, type = "l")
```



Now calculate the treatment effect at each point on the frontier

```
my.form = as.formula(re78 ~ treated + age + education +
                       black + married + nodegree + re74)
our.estimates <-
  estimateEffects(our.frontier,
                  formula = "re78 ~ treated",
                  mod.dependence.formula = my.form,
                  continuous.vars = c("age","education",
                                      "re74"))
```

Treatment effect at each point on the frontier



Questions?