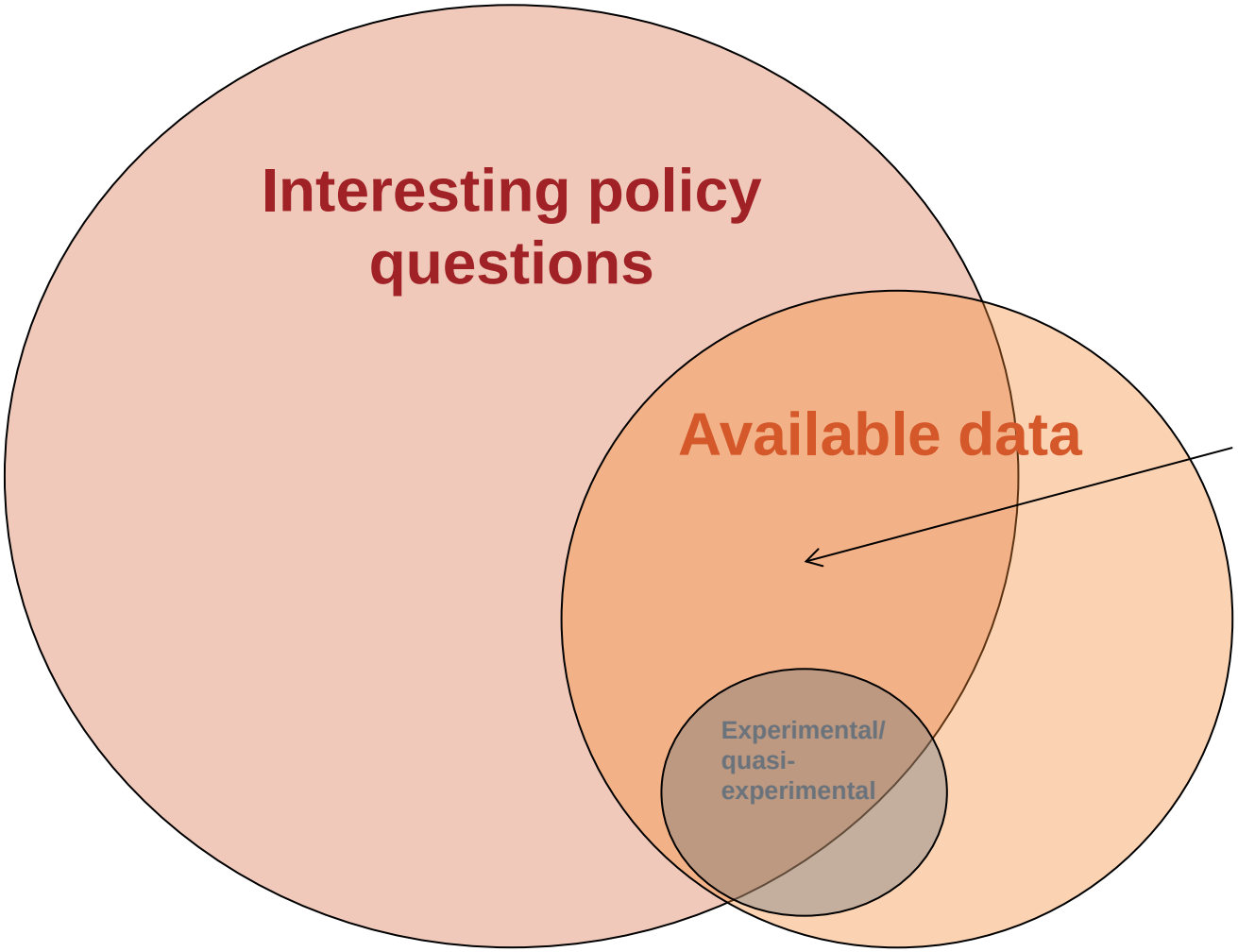
Causal inference with observational data in R: A walk-through from some recent research

Jim Savage
Grattan Institute Associate

Khakieconomist.com

Causal inference with observational data



Motivation:

Lost of interesting questions have data we can use to inform decisions, though experiments/ natural experiments may not exist. We still want to make causal inference.

All inference will end up being interpreted in a causal way

Outline—inference in R

- An example from *Graduate Winners*

Data:

- Choosing the appropriate model
- Choosing independent variables—bad control?
- Cleaning the data (missing data? Scaling variables) – we can talk about this after the presentation.

Inference:

- Gold standard: lots of data and good instrument
- No instrument? Am I estimating on the right data?
- Propensity score matching
- Proximity score matching (new!)

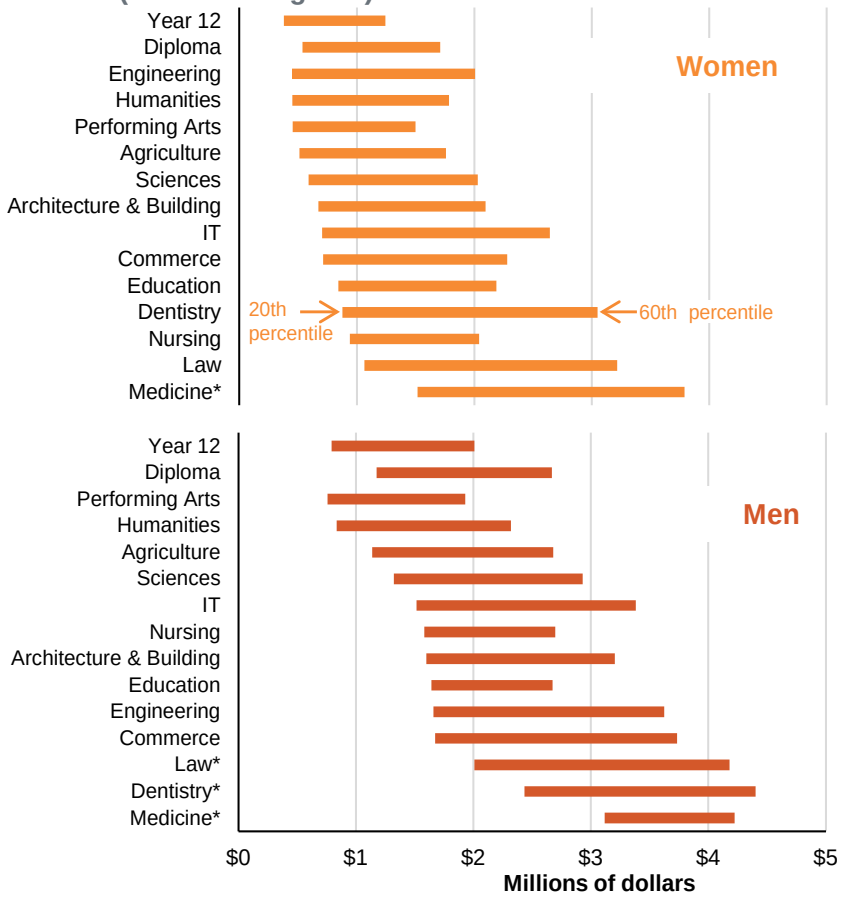
Graduate Winners

- Graduates earn a lot more
- Tuition is heavily subsidised by government

- This is entirely justifiable *if* graduates provide large externalities *and* would not study otherwise

- There are many valid claims on the public purse; is university tuition one of them?

Figure 1 – Spread of gross lifetime earnings: 20th – 60th percentile (Bachelor degrees)



Problem with estimating causal effect

- Differences between graduates and non-graduates may be because of selection/omitted variables bias
  - Graduates may be quite different to non-graduates
  - May have pushy parents, or high-quality peers

- Not allowed to randomly educate some people more and others less.

Graduate Winners

Public benefits:

- Graduates pay a lot more tax (as they earn more)
- How about non-financial spill-overs?

Our approach:

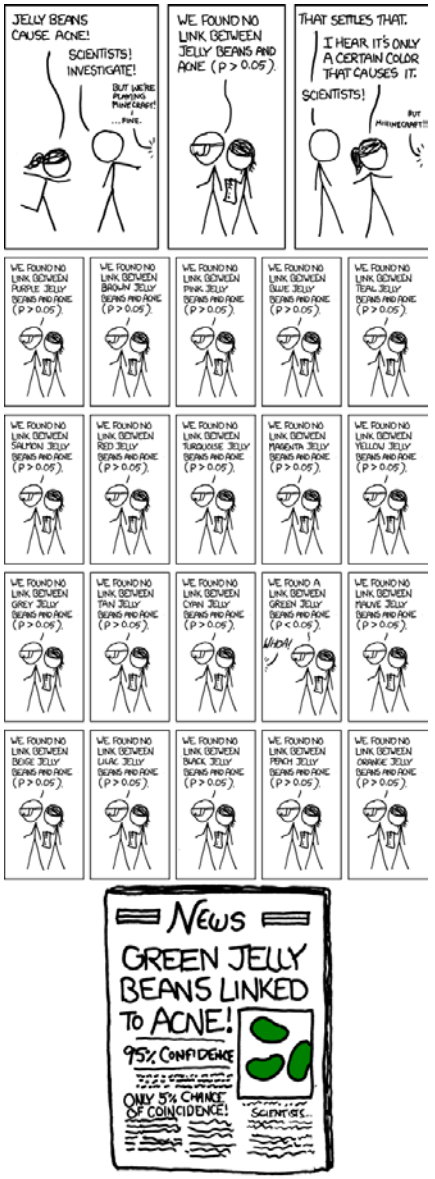
Data: HILDA, General Social Survey, Australian Survey of Social Attitudes (2005, 2007).

Selected many variables (>70) that would be thought of as contributing to public good.

Several techniques to try to (get closer to) identify(ing) causal effect of university education on non-financial outcomes

Many dependent variables: risky

If you're not explicitly incorporating previous findings into priors, you can always check your results against the literature.



Executing many models

In R, it's quite easy to run many models, and capture their output in a file. If you have a data-frame of dependent variables, and one of independent variables:

```
for (i in names(dep.vars)){
        ind.vars1 <- ind.vars[complete.cases(dep.vars[,i]),]
        Model.code <- paste("lm(", i, "~., data = ind.vars1)")
        Output <- eval(parse(text = Model.code))
        Text.insert <- capture.output(summary(Output))
        cat(Text.insert, file = "filename.txt", sep = "\n",
        append = TRUE)
}
```

Choosing a model

Choose according to the dependent variable:

Continuous dependent variable → linear-style model
In R: lm, lmer (in lme4)

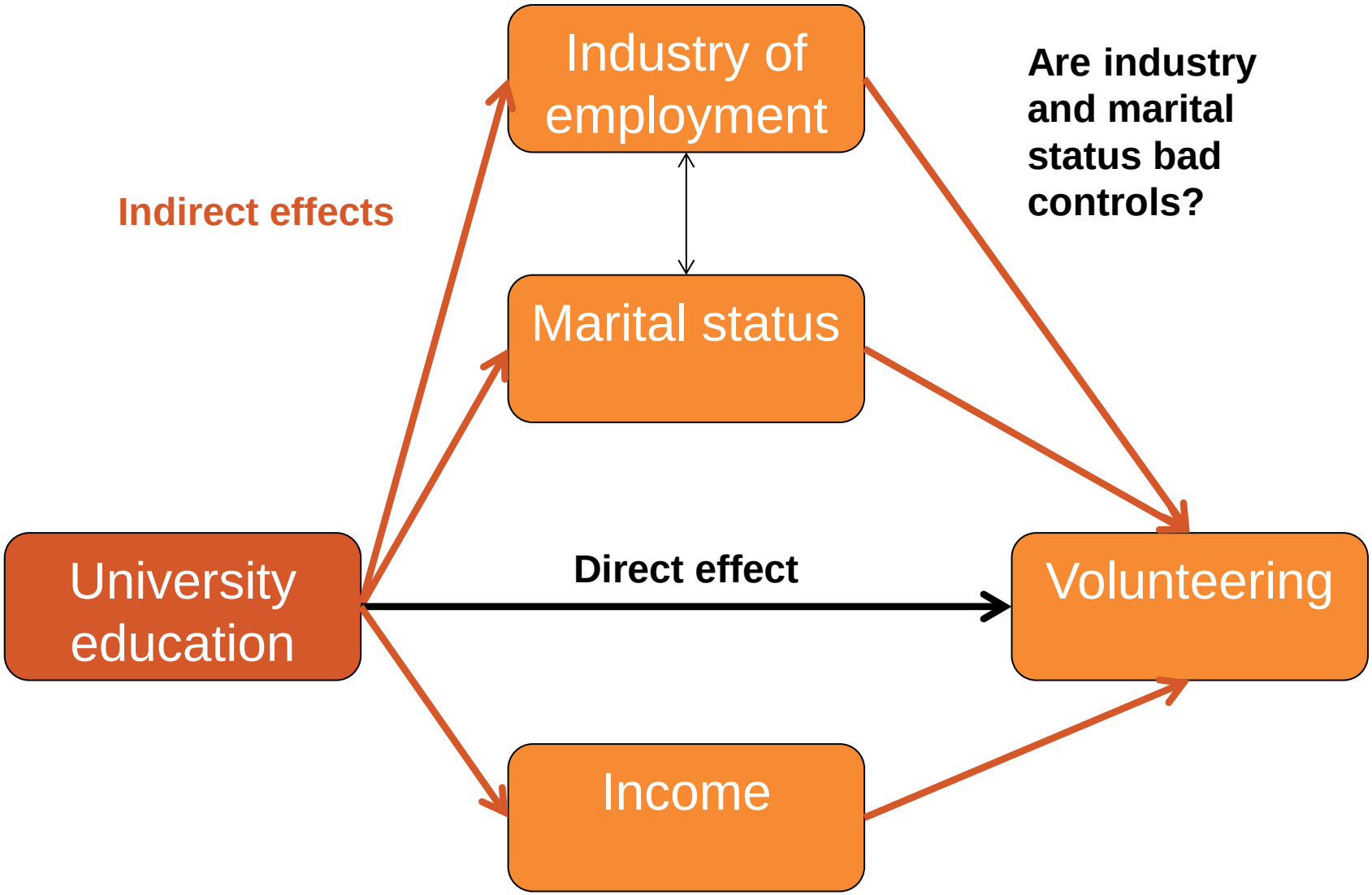
Binary dependent variable → probit/logit-style
In R: glm, glmer (in lme4), bayesglm (in arm)

Likert variable (terrible, bad, neutral, good, excellent) → Ordered logit/probit
In R: bayespolr (arm) was the only one that would work properly for me.

Factor data (red, blue, purple) → Multinomial Logit
In R: mlogit (mlogit)

Choosing independent variables

Remember: we want the causal effect of university



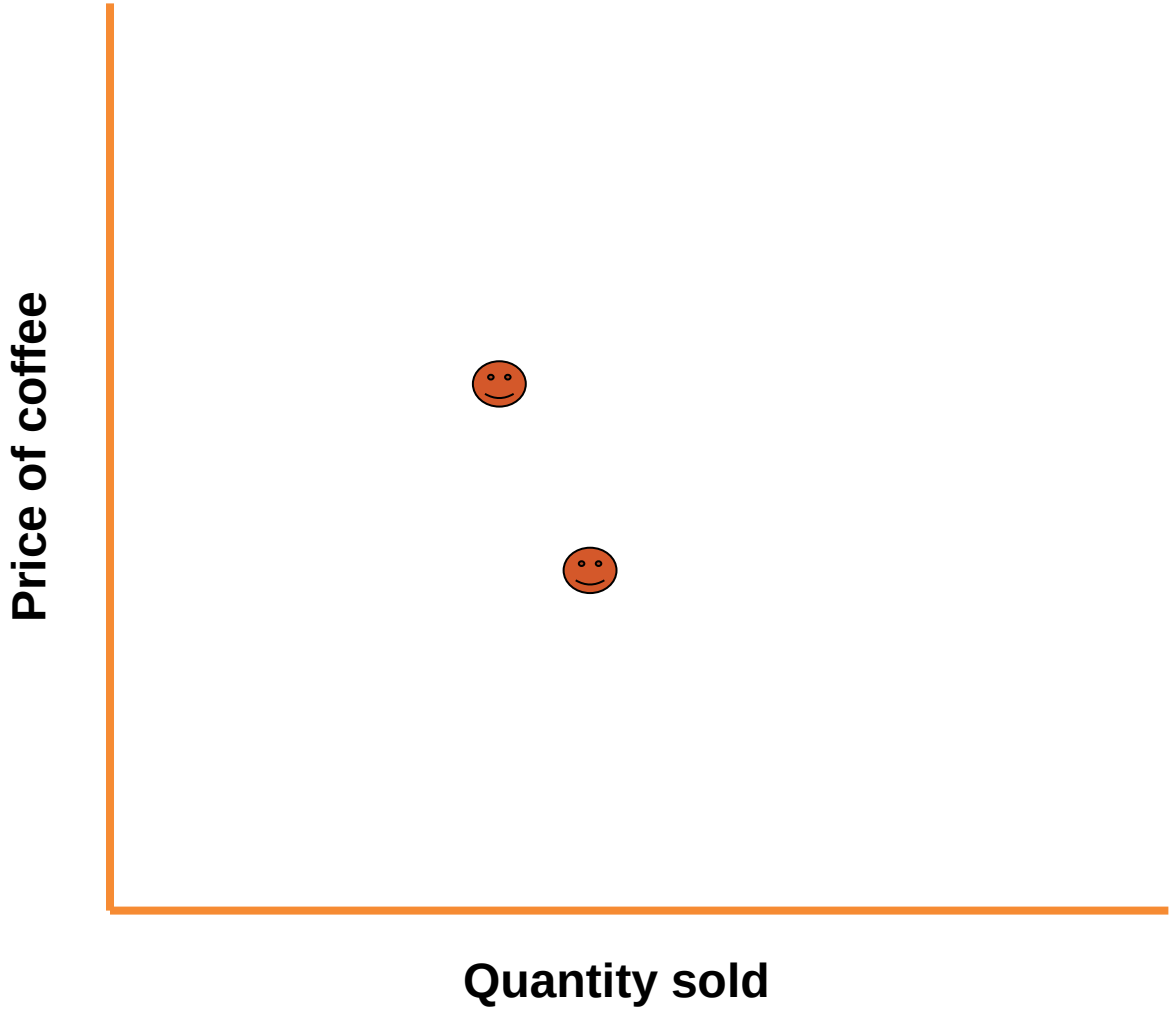
Bad controls: a difficult dilemma

If there are plausible effects of university on industry or marital status, and if these affect volunteering, then the coefficient on university won't be the causal effect.

Without exogenous treatment, a choice between omitted variables bias with certainty, or wrong estimate with near certainty.

Instrumental Variables can help us get around this.

Gold standard: big data set with IV

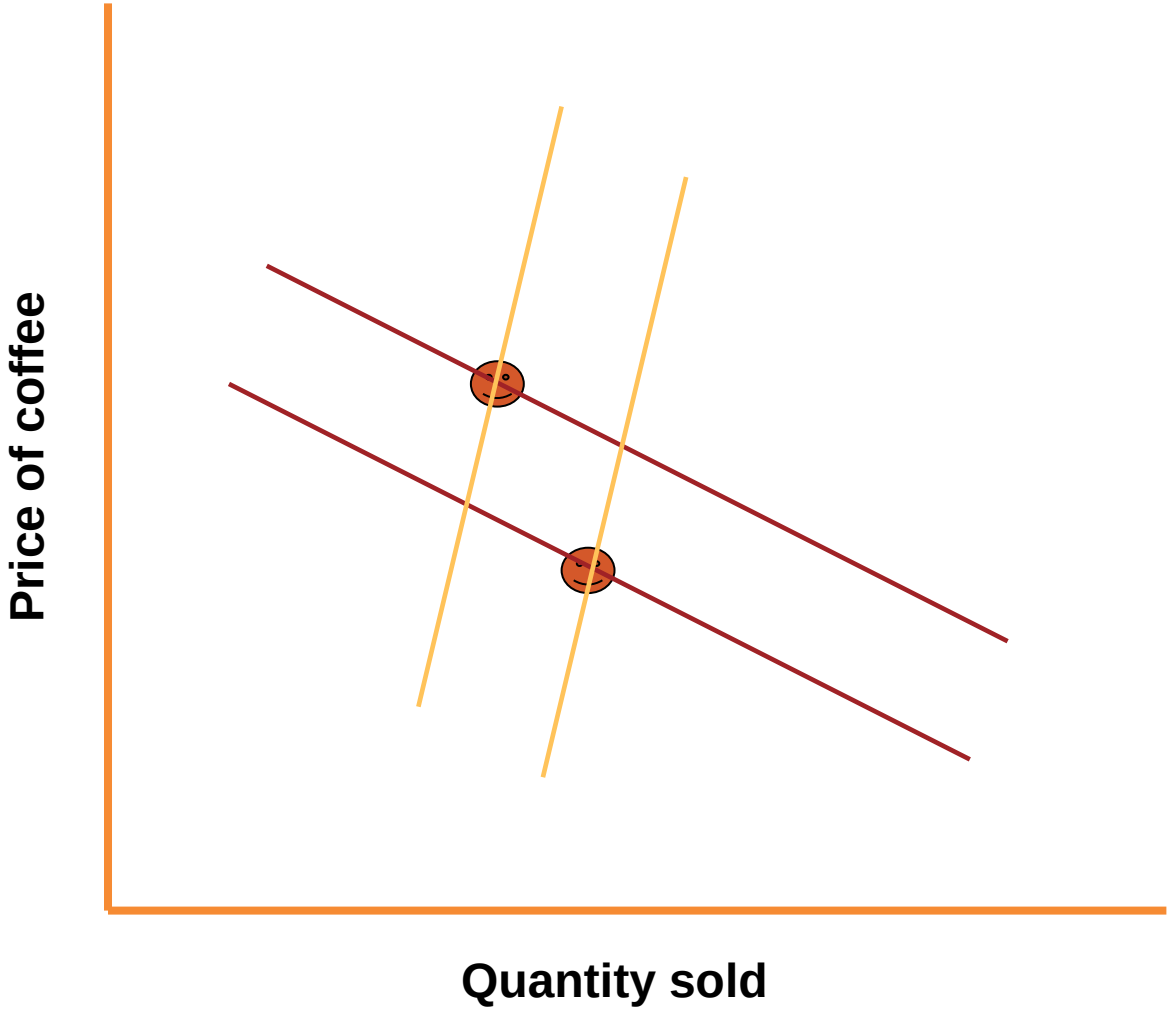


We want to know how quantity responds to price

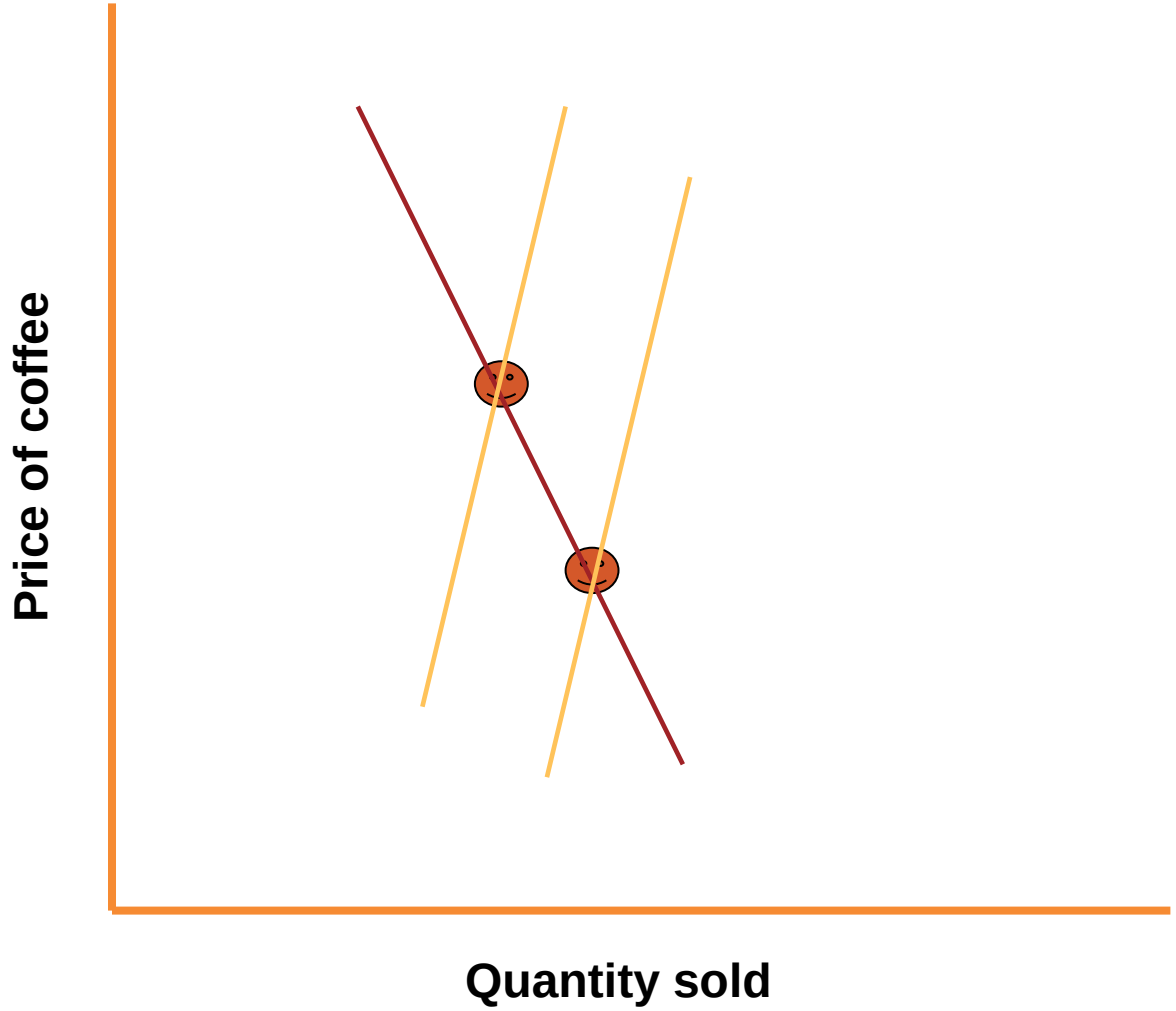
All we see is price and quantity together.

(Assume coffee cannot be stored)

Gold standard: big data set with IV



Gold standard: big data set with IV



The difference may have a large impact on business decisions.

Instrumental variables

We want to find a variable that:

1. Explains changes in the supply of coffee
2. Does not explain changes in the price of coffee other than through its effect on supply (ie. No effect on demand)

Eg. Plague of locusts in Ethiopia, rainfall in Brazil. These would constitute natural experiments—randomly assign treatment (lower/higher prices) in some periods but not others.

Method of Two-Stage Least Squares

1) Price = X'B + A* rainfall in Brazil + e
2) Quantity = X'C + d * fitted price + v

Standard errors from this method are wrong—use bottled functions

Instrumental variables

In R: Several implementations of IV

ivreg (in AER); plm (in plm) – you may have to write your own

There are still issues

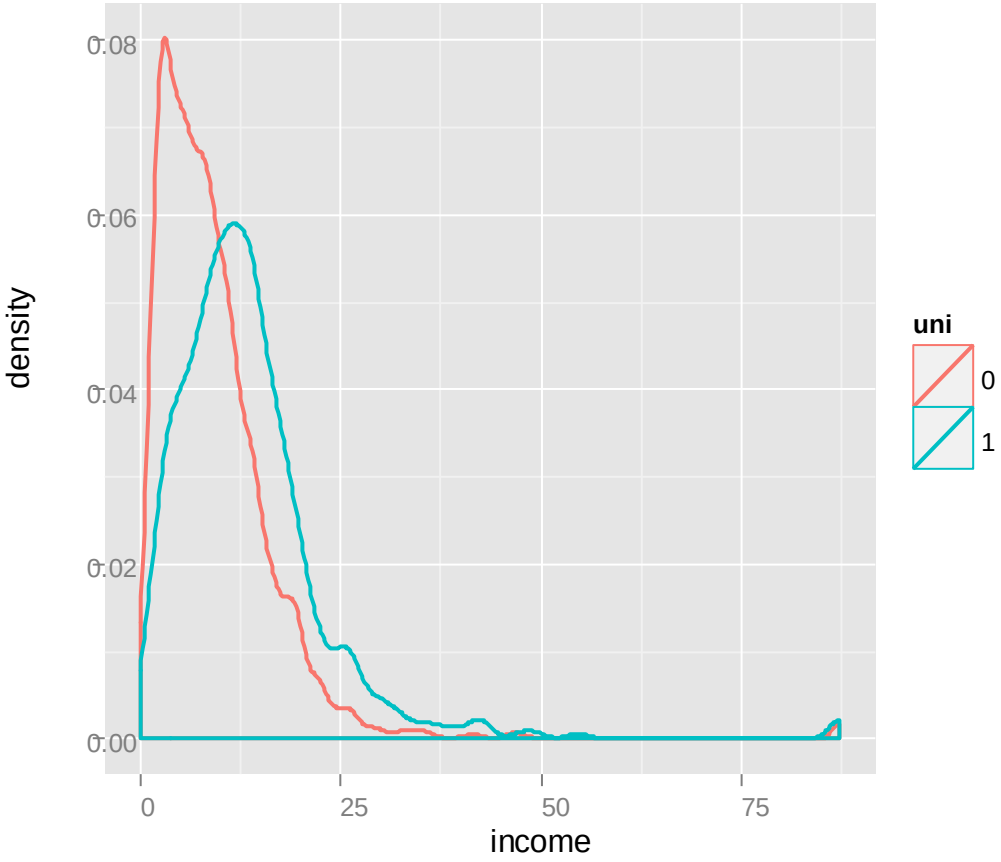
- **Good instruments are hard to find/rare in data**
- **Small sample issues (bias towards OLS estimates)**
- **Not magical**

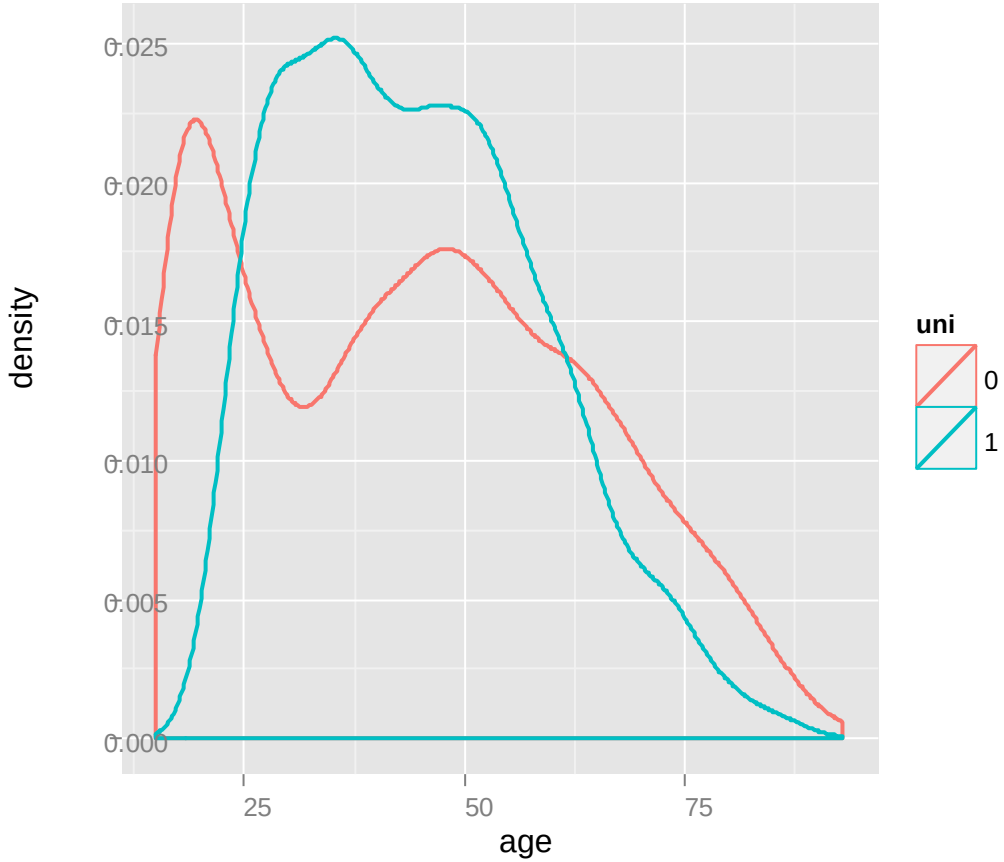
So what if I don’t have a good IV?

- **IVs are difficult to find/justify**
- **Don’t give up! Second-best solutions exist.**

Synthetic control groups

Are we estimating the model on the right data?





From HILDA 2010

Synthetic control groups

Idea:

- **Construct a control group that is as similar as possible in the observed covariates to the treatment group, only untreated**
- **Restrict the sample to just the treated and the comparable control**
- **Standard regression analysis on this restricted group**

- **In the context of Graduate Winners: who are the survey respondents without a degree but otherwise most like university graduates?**

Synthetic control groups

The standard approach: propensity score matching (Ruben and Rosenbaum 1983; Deheji and Wahba 2002)

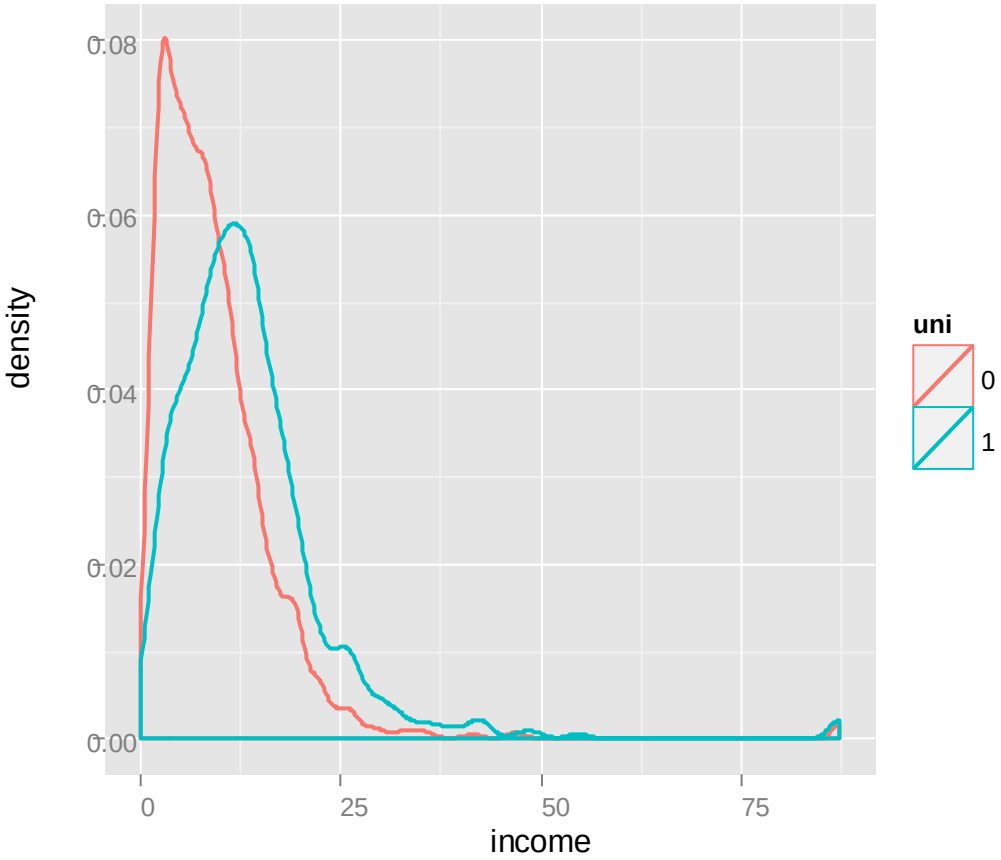
1. **Run a model to predict treatment**
2. **For each treated observation, take the untreated observation with the closest fitted value**
3. **Discard the rest of the sample**

Synthetic control groups

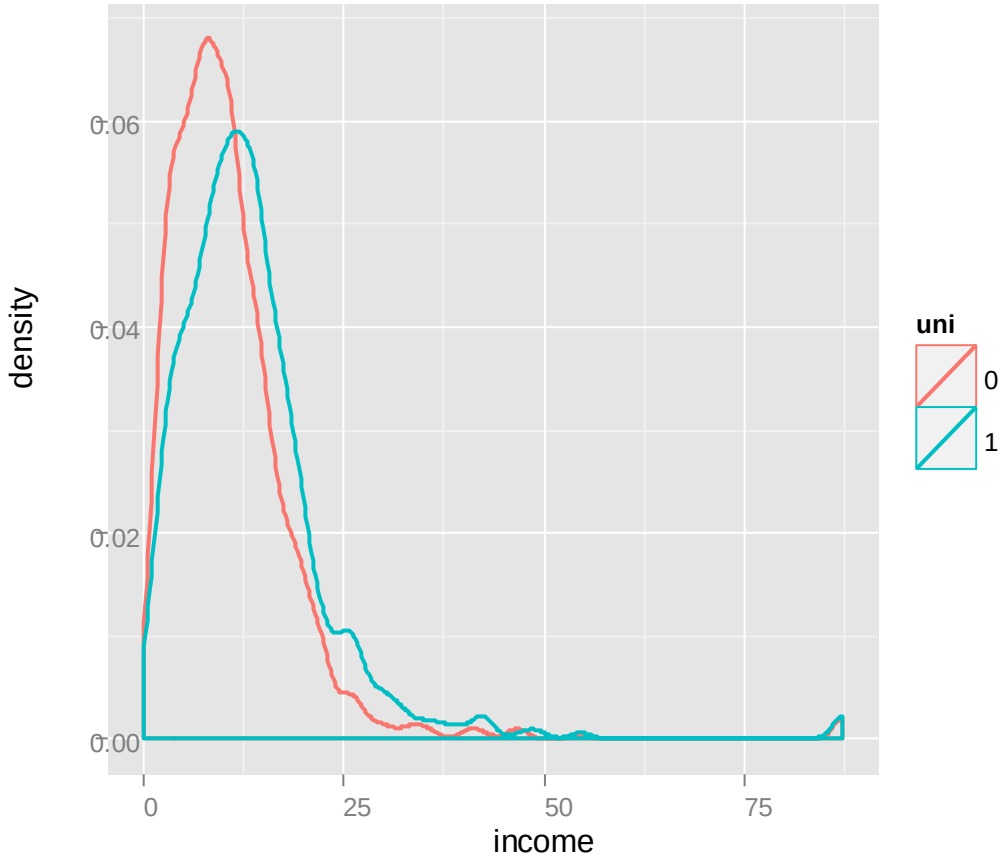
Easy in R (using “matching()”from ‘arm’):

```
#Propensity model
prop <- glm(uni ~ ., family = binomial(link = "probit"), data = df1)

# Get fitted values
prop.fitted <- predict(prop, type = "link")

# Get matches
matches <- matching(df1$uni, prop.fitted)

# Create new dataset using matches only
df1.matched <- df1[matches$matched, ]

# Run your model
control.match <- lm(depvar ~ ., data = df1.matched)
```

Synthetic control groups

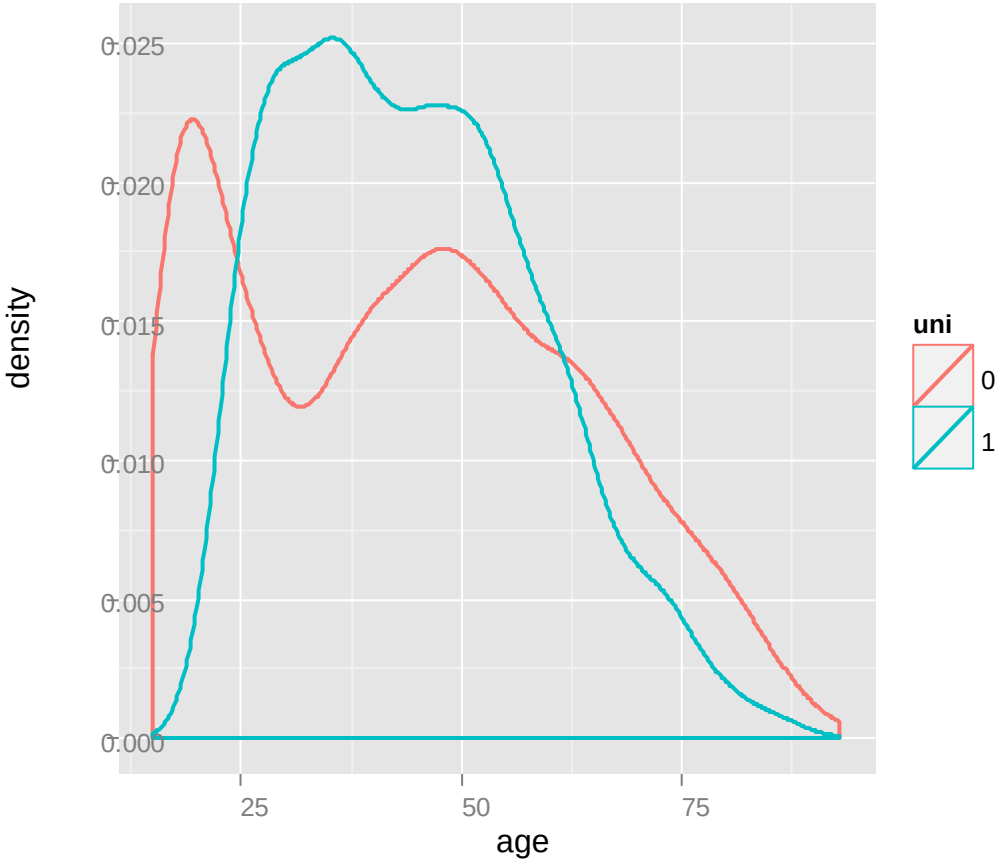


No matching

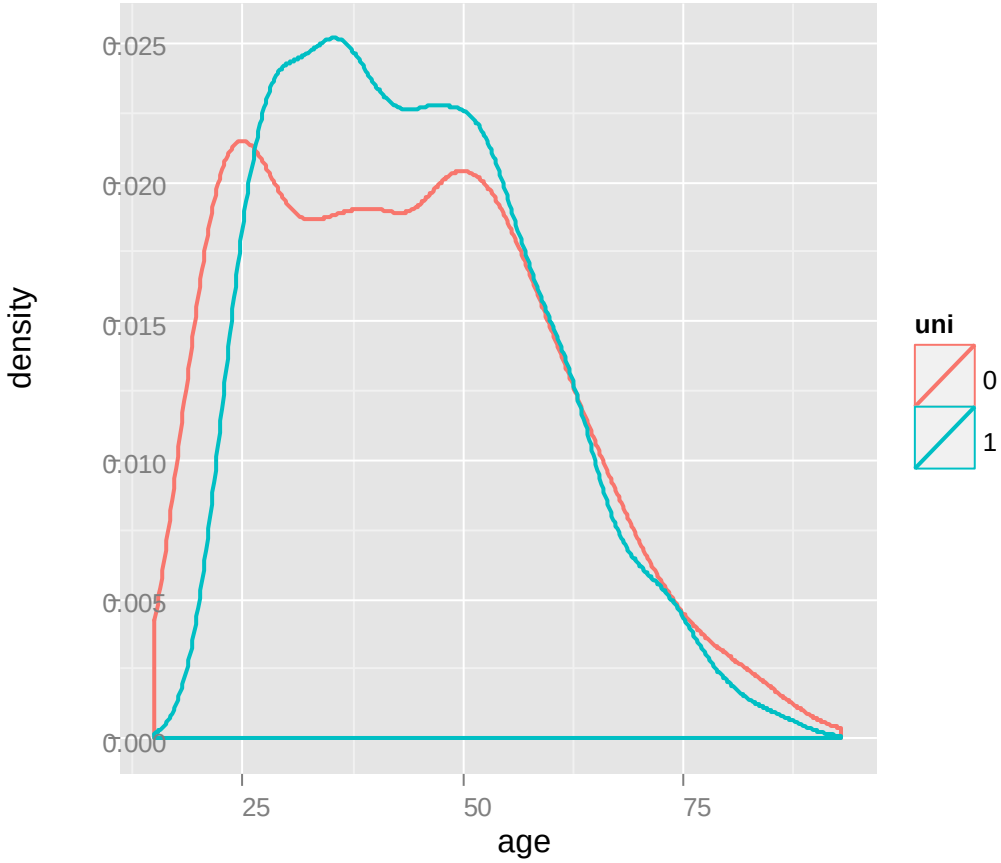


Propensity score matching

Synthetic control groups

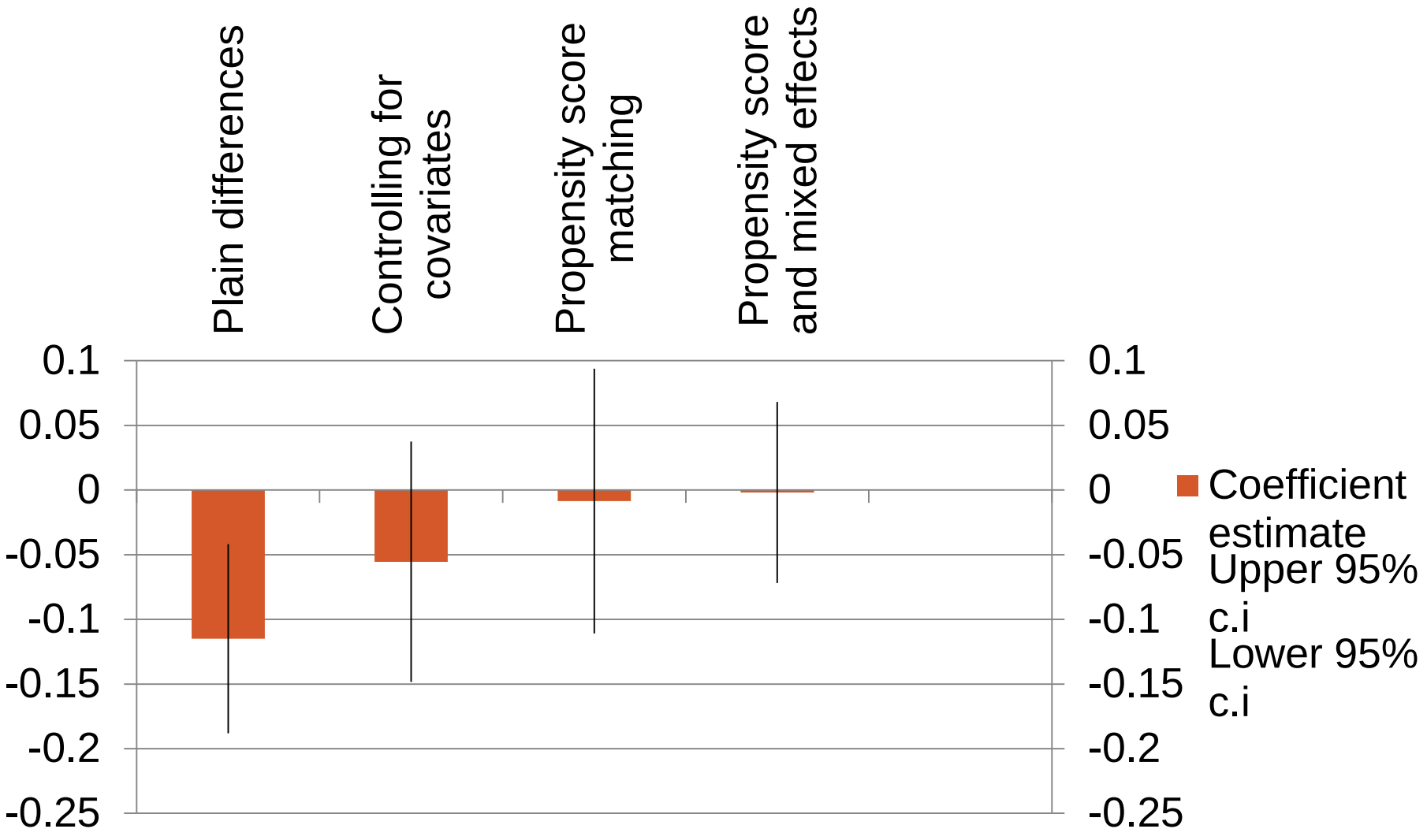


No matching



Propensity score matching

Does university improve life satisfaction?



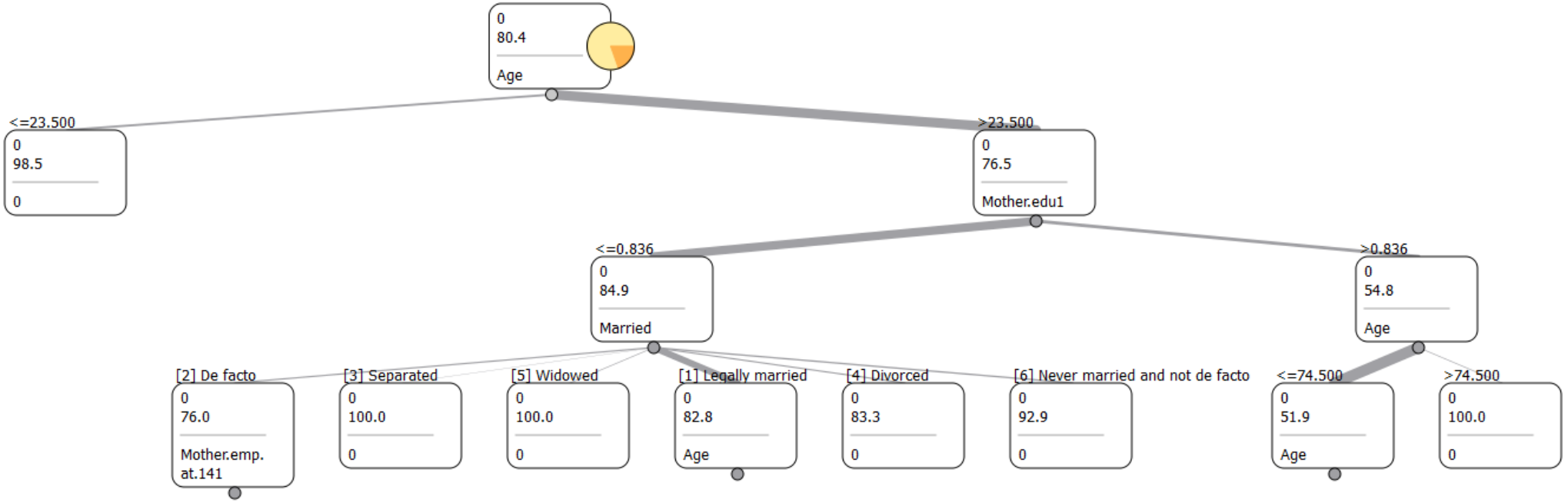
Drawbacks of propensity score matching

While propensity score matching can improve causal estimates, the estimates may not be robust to changes in the propensity score model (Smith and Todd, 2005)

My proposal: use a more robust model for propensity, and match only on variables that matter for predicting treatment.

…

CART and Random Forest



If two observations wind up in the same terminal node, their proximity goes up by 1

Proximity Score Matching

The idea:

1. **Run a Random Forest on the treatment, saving the proximity scores**
2. **For each treated observation, select the untreated observation with the highest proximity**
3. **Save the proximities**
4. **Run a regression, using the saved proximities as weights (that way, we value observations that are more like treated observations more than we do observations that are less similar).**
5. **Simulate 1:4 to get confidence intervals (especially important for small samples)**

Proximity Score Matching

Implementation:

```
In R:
# Get a logical vector for treatment:
uni <- df1$uni==1

# Run the Random Forest and save the proximity matrix
rf1 <- randomForest(uni ~ ., data = df1, ntree = 200, proximity =
TRUE)

# Get the proximity matrix and set the diagonal entries to 0
prox <- rf1$proximity - diag(nrow(rf1$proximity))

# Subset the proximity matrix for treated rows, untreated columns
prox.true <- prox[uni,!uni]

# What are the untreated observations with the closest proximity?
control.samples <- apply(prox.true, 1, which.max)
```

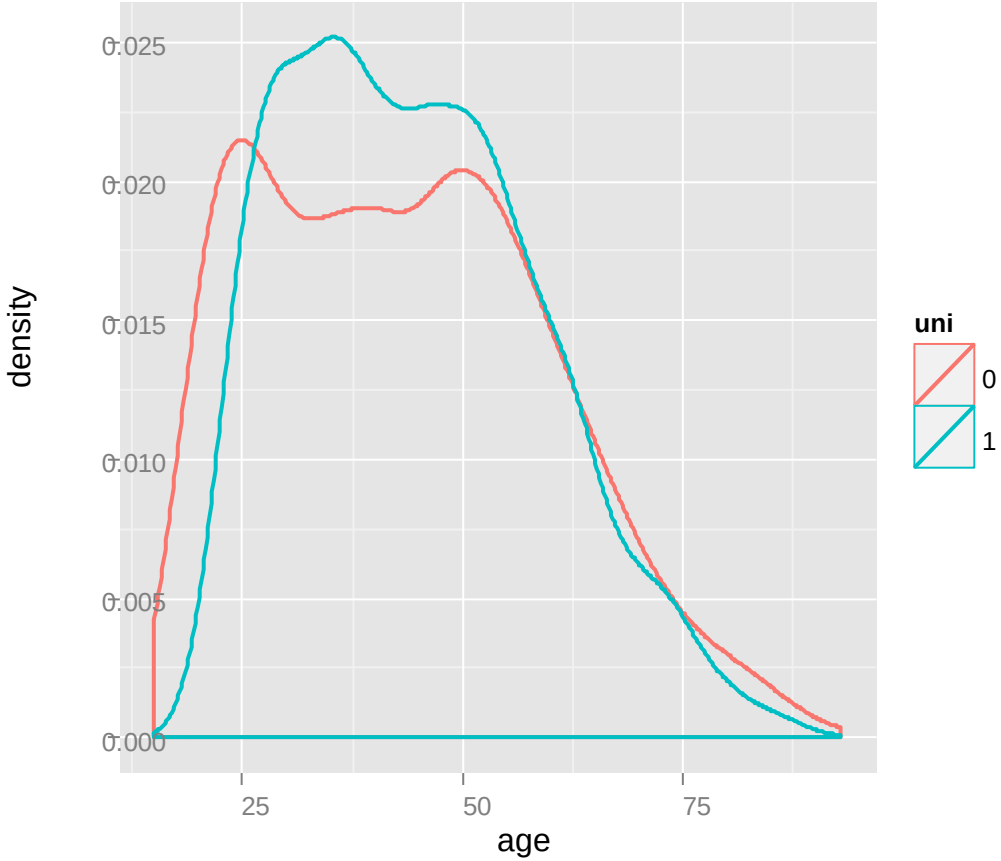
Proximity Score Matching

Implementation:

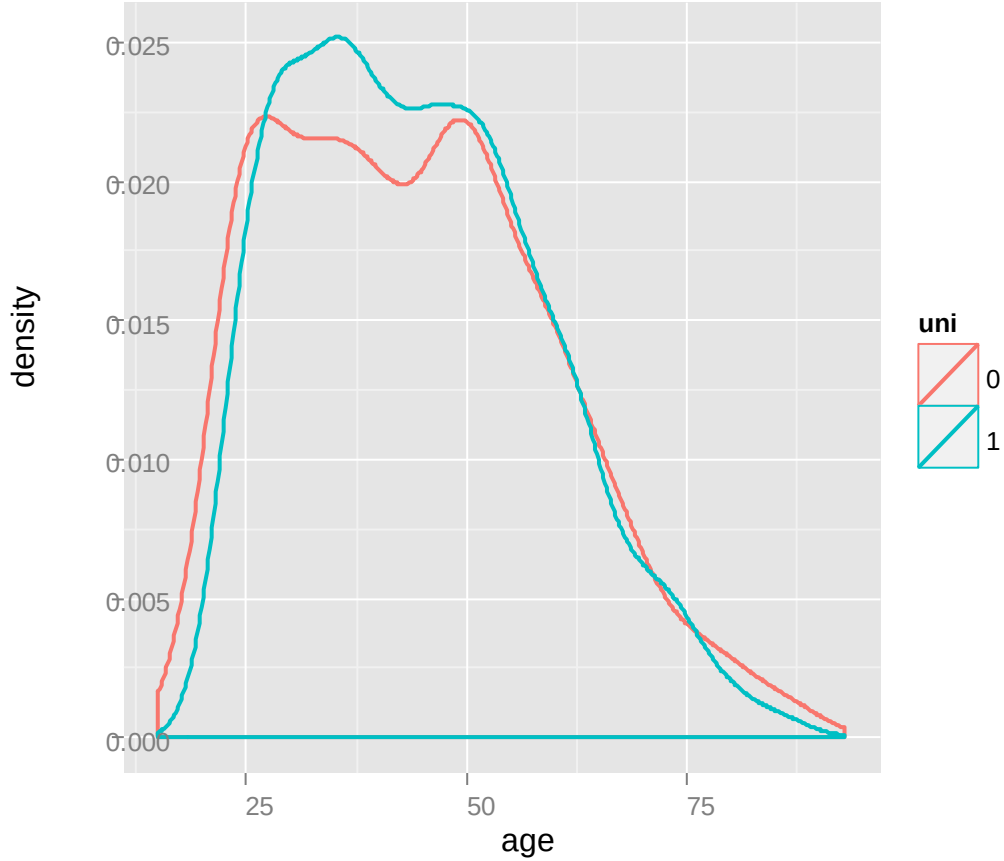
```
In R:
# Select the original dataset for treated observations
treatment <- df1[uni,]

# Get the observations with no university
df.no.uni <- df1[!uni,]

# We don’t want to draw an untreated observation more than once
control.samples.u <- unique(control.samples)

# Get the untreated observations that are most similar to the
treated observations
control.u <- df.no.uni[control.samples.u,]

# Mash treated and untreated observations together
whole.sample.u <- rbind(treatment, control.u)
```

Proximity Score Matching

Comparison with propensity score matching



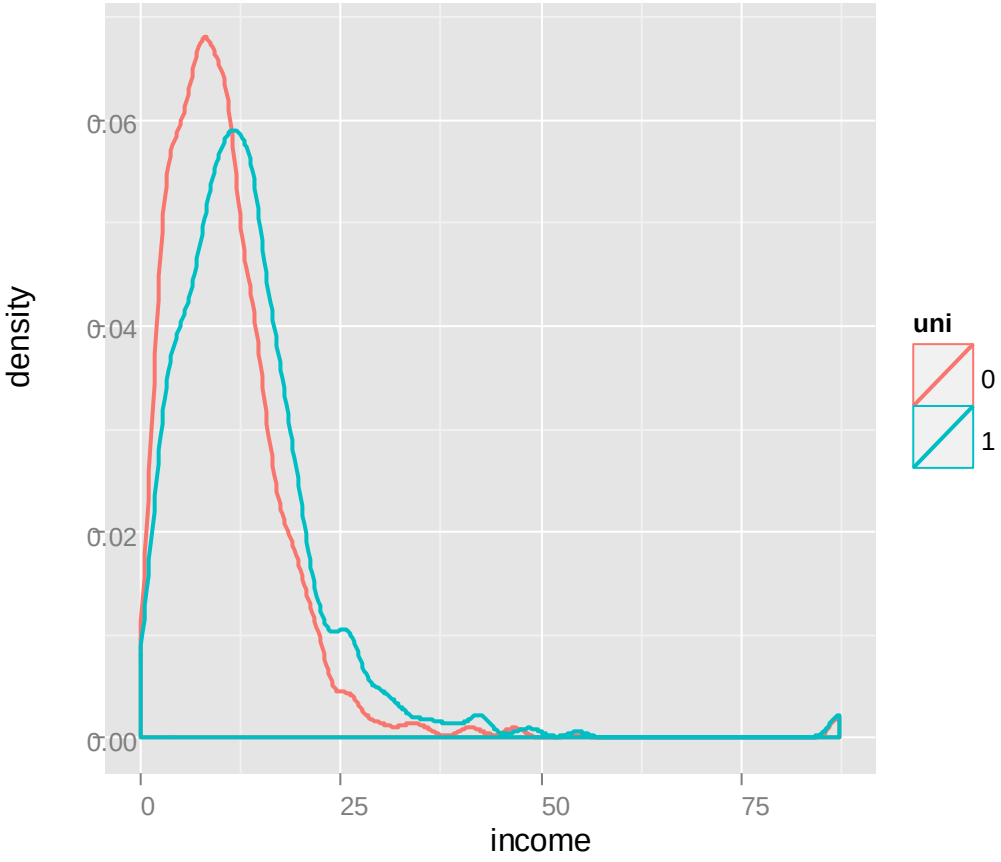
Propensity score matching



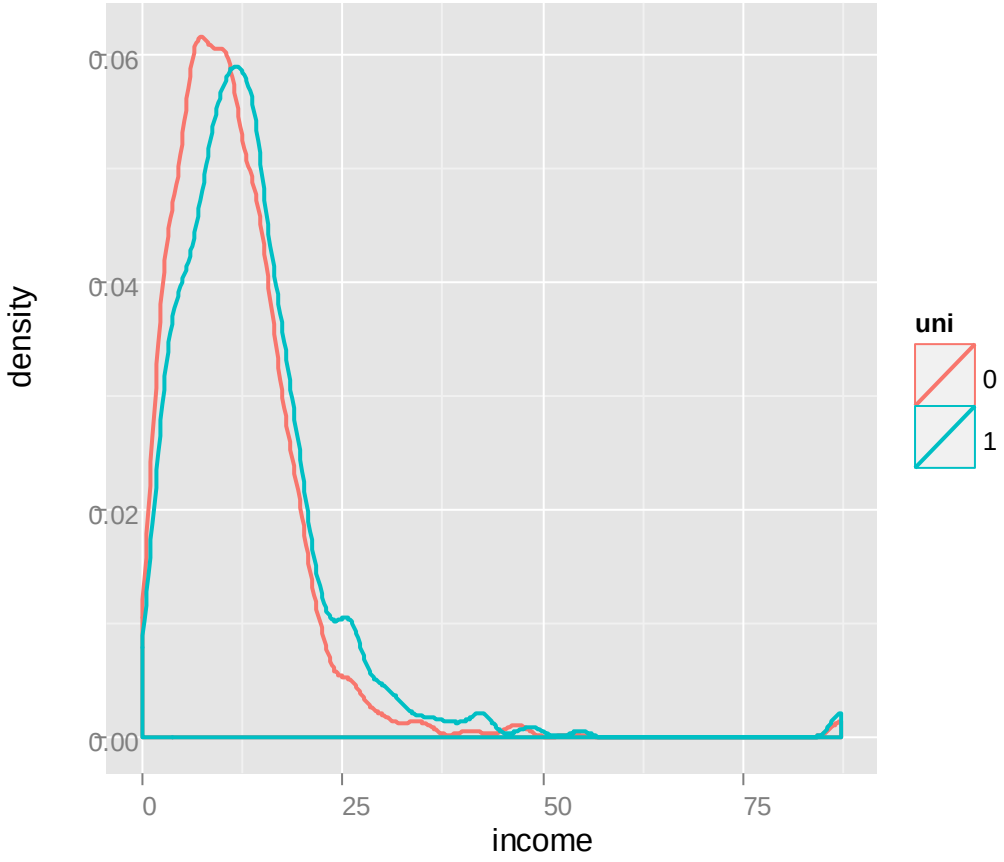
Proximity score matching

Proximity Score Matching

Comparison with propensity score matching

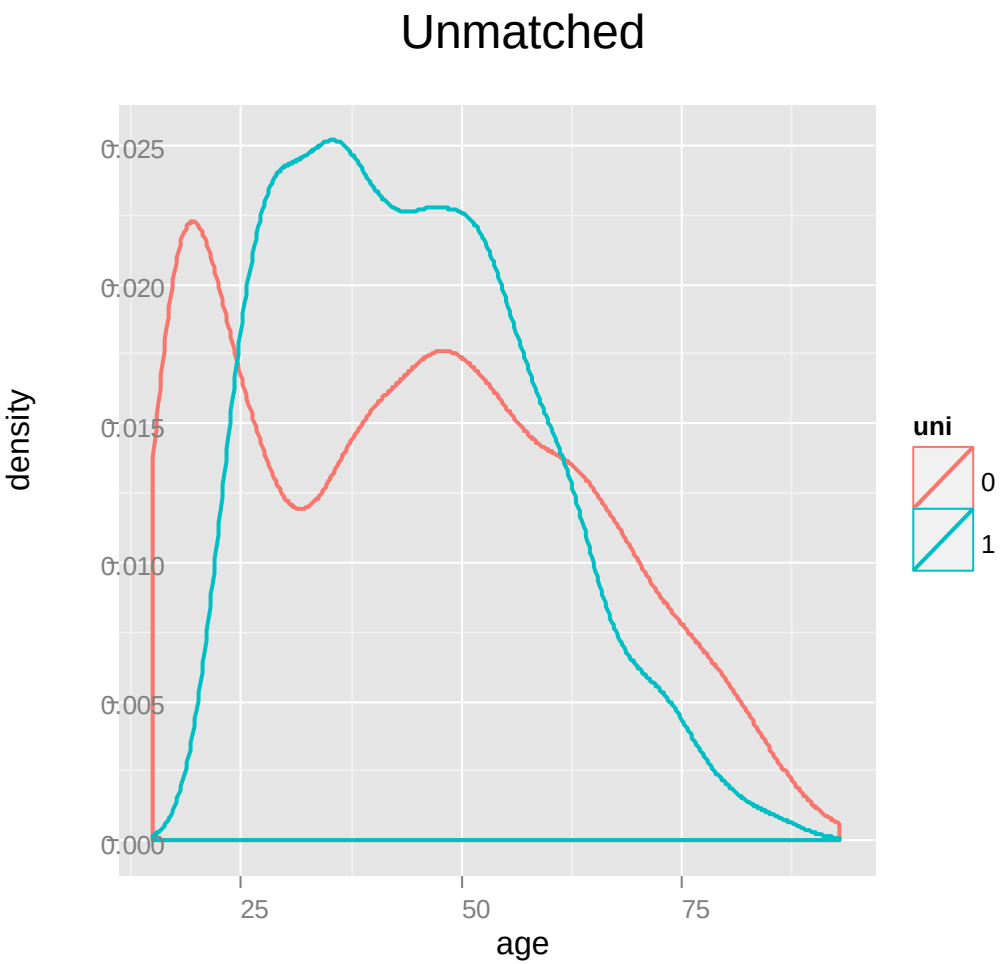


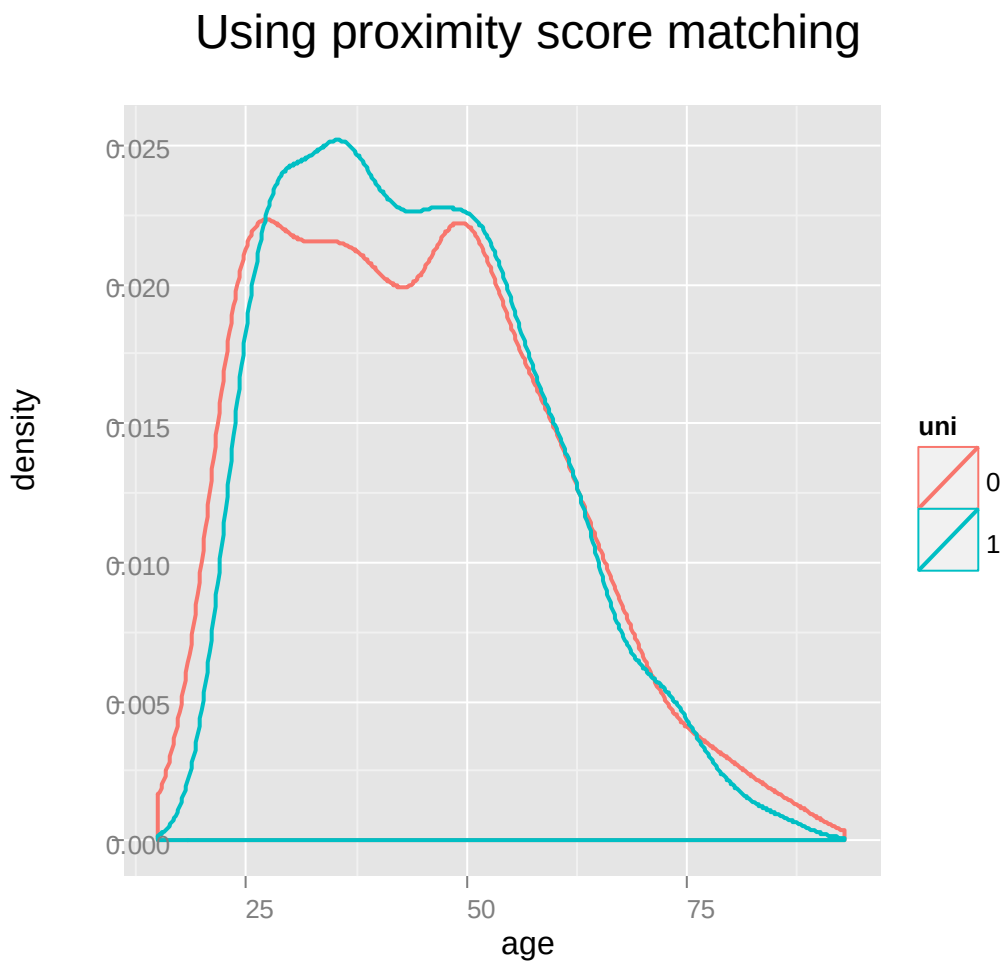
Propensity score matching



Proximity score matching

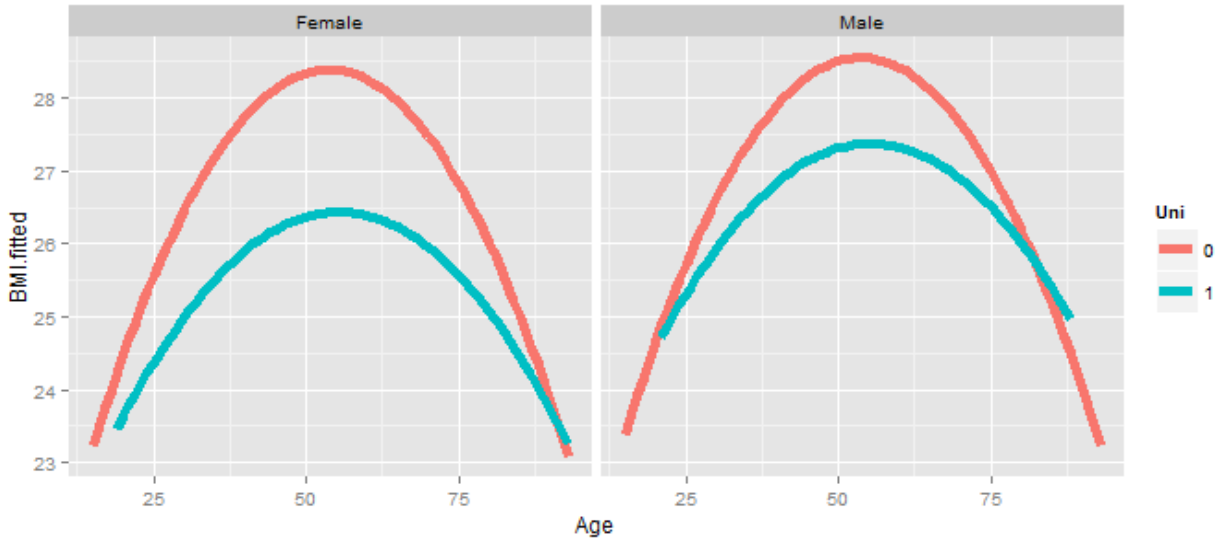
The results are striking





Does university make you skinny?

With no matching



With proximity score matching