Quant II

Lab 5: Conditioning strategies

Giacomo Lemoli

February 23, 2023

Today's plan

- Identify effects under CIA assumption

Today's plan

- Identify effects under CIA assumption
- OLS

Today's plan

- Identify effects under CIA assumption
- OLS
- OLS properties

Today's plan

- Identify effects under CIA assumption
- OLS
- OLS properties
  - FWL theorem (refresher)

Today's plan

- Identify effects under CIA assumption
- OLS
- OLS properties
  - FWL theorem (refresher)
  - Weighting and effective sample

Today's plan

- Identify effects under CIA assumption
- OLS
- OLS properties
  - FWL theorem (refresher)
  - Weighting and effective sample
- Sensitivity

Today's plan

- Identify effects under CIA assumption
- OLS
- OLS properties
    - FWL theorem (refresher)
    - Weighting and effective sample
- Sensitivity
- Other conditioning strategies: matching and weighting

Today's plan

- Identify effects under CIA assumption
- OLS
- OLS properties
  - FWL theorem (refresher)
  - Weighting and effective sample
- Sensitivity
- Other conditioning strategies: matching and weighting
  - Matching and weighting

Today's plan

- Identify effects under CIA assumption
- OLS
- OLS properties
  - FWL theorem (refresher)
  - Weighting and effective sample
- Sensitivity
- Other conditioning strategies: matching and weighting
  - Matching and weighting
  - Doubly-robust estimators

Identification under CIA

- Goal standard: randomized experiment. Treatment assignment is known to (made by) the researcher

Identification under CIA

- Goal standard: randomized experiment. Treatment assignment is known to (made by) the researcher
- Observational settings: treatment assignment not observed. Assumptions needed

Identification under CIA

- Goal standard: randomized experiment. Treatment assignment is known to (made by) the researcher
- Observational settings: treatment assignment not observed. Assumptions needed
- CIA/Strong ignorability: treatment assignment is independent of potential outcomes after conditioning on a set of variables

Identification under CIA

- Goal standard: randomized experiment. Treatment assignment is known to (made by) the researcher
- Observational settings: treatment assignment not observed. Assumptions needed
- CIA/Strong ignorability: treatment assignment is independent of potential outcomes after conditioning on a set of variables
- If the same characteristics give equal probability of being treated, actual differences are due to a random draw

Identification under CIA

- Goal standard: randomized experiment. Treatment assignment is known to (made by) the researcher
- Observational settings: treatment assignment not observed. Assumptions needed
- CIA/Strong ignorability: treatment assignment is independent of potential outcomes after conditioning on a set of variables
- If the same characteristics give equal probability of being treated, actual differences are due to a random draw
  - E.g. different treatment probabilities across strata

Identification under CIA

- Goal standard: randomized experiment. Treatment assignment is known to (made by) the researcher
- Observational settings: treatment assignment not observed. Assumptions needed
- CIA/Strong ignorability: treatment assignment is independent of potential outcomes after conditioning on a set of variables
- If the same characteristics give equal probability of being treated, actual differences are due to a random draw
  - E.g. different treatment probabilities across strata
- Informally: "selection on observables"

Identification under CIA

- Goal standard: randomized experiment. Treatment assignment is known to (made by) the researcher
- Observational settings: treatment assignment not observed. Assumptions needed
- CIA/Strong ignorability: treatment assignment is independent of potential outcomes after conditioning on a set of variables
- If the same characteristics give equal probability of being treated, actual differences are due to a random draw
    - E.g. different treatment probabilities across strata
- Informally: "selection on observables"
- Considered strong in most contexts

Identification under CIA

Recall the CIA assumption.

$$D_i \perp (Y_{1i}, Y_{0i}) | X_i, 0 < P(D_i = 1 | X_i) < 1$$

This assumption guarantees that group/covariate stratum-specific ATEs are identified in the data:

$$\tau(x) = E[Y_{1i} - Y_{0i} | X_i = x] = E[Y_{1i} | D_i = 1, X_i = x] - E[Y_{0i} | D_i = 0, X_i = x]$$

And then τ_{PATE} is identified by averaging over the covariates distribution (cf. Imbens and Wooldridge 2009, p.26-27)

$$\tau_{PATE} = E[\tau(x)] = \int_X \tau(x) dF(x)$$

Same holds for τ_{PATT}, changing the distribution over which to average:

$$\tau_{PATT} = E[\tau(x) | D_i = 1] = \int_{X|D=1} \tau(x) dF(x | D_i = 1)$$

CIA in empirical research

When can CIA be plausibly invoked?

Geographic factors influence treatment assignment. E.g. comparative development literature.

CIA in empirical research

When can CIA be plausibly invoked?

Geographic factors influence treatment assignment. E.g. comparative development literature.

ON THE ORIGINS OF GENDER ROLES: WOMEN AND THE PLOUGH*

Alberto Alesina
Paola Giuliano
Nathan Nunn

The study examines the historical origins of existing cross-cultural differences in beliefs and values regarding the appropriate role of women in society. We test the hypothesis that traditional agricultural practices influenced the historical gender division of labor and the evolution of gender norms. We find that, consistent with existing hypotheses, the descendants of societies that traditionally practiced plough agriculture today have less equal gender norms, measured using reported gender-role attitudes and female participation in the workplace, politics, and entrepreneurial activities. Our results hold looking across countries, across districts within countries, and across ethnicities within districts. To test for the importance of cultural persistence, we examine the children of immigrants living in Europe and the United States. We find that even among these individuals, all born and raised in the same country, those with a heritage of traditional plough use exhibit less equal beliefs about gender roles today. *JEL* Codes: D03, J16, N30.

CIA in empirical research

Treatment assignment function is observed (but not deterministic. Why?). E.g. media effects literature.

CIA in empirical research

Treatment assignment function is observed (but not deterministic. Why?). E.g. media effects literature.

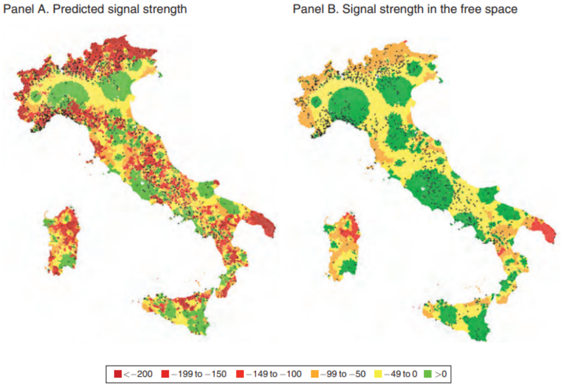


FIGURE 4. MEDIASET SIGNAL STRENGTH AND HYPOTHETICAL SIGNAL STRENGTH IN THE FREE SPACE IN 1985

Notes: Panel A shows Mediaset actual signal strength, in decibels (dB), across municipalities in 1985. Panel B shows the hypothetical signal strength in the absence of geomorphological obstacles. The black dots represent the location of transmitters.

Regression and CIA

- Linear regression is the most obvious tool to estimate causal effects under CIA

Regression and CIA

- Linear regression is the most obvious tool to estimate causal effects under CIA
- Estimates linear approximations of the conditional means of the potential outcomes

Regression and CIA

- Linear regression is the most obvious tool to estimate causal effects under CIA
- Estimates linear approximations of the conditional means of the potential outcomes
- Important property of OLS: residualize by covariates is equivalent to control for them

Frisch-Waugh-Lovell theorem: a refresher

- Linear model with K covariates. In matrix form: $y = X'\beta + \varepsilon$
- FWL gives a formula for the OLS estimate of the k^{th} coefficient.

$$\hat{\beta}_k = (X_k' M_{[X_{-k}]} X_k)^{-1} X_k' M_{[X_{-k}]} y$$

Equivalent to the following:

- Regress the individual variable X_k on all the other covariates and take the residuals
- Regress the outcome variable y on all the covariates, except X_k, and take the residuals
- Regress the residuals of y on the residuals for X
- Note that to get $\hat{\beta}_k$ it is enough to regress the non-residualized y on residualized X_k (why?), but the SE won't be right
- Useful because typically we are interested in just one regressor (e.g. a treatment indicator), so we can reduce the dimensionality of the model

FWL in practice

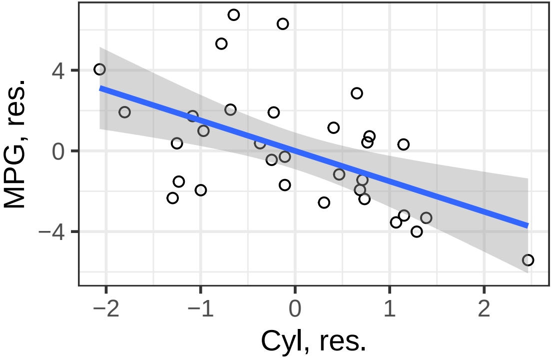
```r
library(tidyverse)
# Import a dataset
data("mtcars")

# Multivariate regression
fit <- lm(mpg ~ cyl + drat + wt, mtcars)

# FWL
resy <- lm(mpg ~ drat + wt, mtcars) %>% residuals()
resx <- lm(cyl ~ drat + wt, mtcars) %>% residuals()
fit2 <- lm(resy ~ resx)

# Compare results
out <- c(coefficients(fit)["cyl"], coefficients(fit2)["resx"])
names(out) <- c("Multivariate", "Univariate Residualized")
out
```

```
##            Multivariate Univariate Residualized
##               -1.509577               -1.509577
```

FWL in practice

Residual-residual plots show the relationship between two variables while controlling for the others

```
as.data.frame(cbind(resy, resx)) %>% rename(mpg_res = resy, cyl_res = resx) %>%
  ggplot(aes(x=cyl_res, y=mpg_res)) + geom_point(size=1.5, colour="black", shape=21) +
  geom_smooth(method="lm") +
  labs(x = "Cyl, res.", y = "MPG, res.") + theme_bw()
```



Heterogeneous treatment effects

Estimate the ATT under CIA:

- Fix a value of $X = x$

Heterogeneous treatment effects

Estimate the ATT under CIA:

- Fix a value of $X = x$
- Estimate the effect of D on Y for units with $X = x$. This is a causal estimate

Heterogeneous treatment effects

Estimate the ATT under CIA:

- Fix a value of $X = x$
- Estimate the effect of D on Y for units with $X = x$. This is a causal estimate
- Repeat for all values of X

Heterogeneous treatment effects

Estimate the ATT under CIA:

- Fix a value of $X = x$
- Estimate the effect of D on Y for units with $X = x$. This is a causal estimate
- Repeat for all values of X
- Aggregate all these causal estimates: a weighted average

Heterogeneous treatment effects

Estimate the ATT under CIA:

- Fix a value of $X = x$
- Estimate the effect of D on Y for units with $X = x$. This is a causal estimate
- Repeat for all values of X
- Aggregate all these causal estimates: a weighted average
  - Weights are the shares of units with $X = x$

Heterogeneous treatment effects

An unbiased estimator is

$$\hat{\tau}_{ATT} = \frac{\sum_X \hat{\tau}_x \hat{P}(D_i = 1|X_i = x)\hat{P}(X_i = x)}{\sum_X \hat{P}(D_i = 1|X_i = x)\hat{P}(X_i = x)}$$

Instead, OLS estimates

$$\hat{\tau}_{OLS} = \frac{\sum_X \hat{\tau}_X \hat{P}(D_i = 1|X_i = x)(1 - \hat{P}(D_i = 1|X_i = x))\hat{P}(X_i = x)}{\sum_X \hat{P}(D_i = 1|X_i = x)(1 - \hat{P}(D_i = 1|X_i = x))\hat{P}(X_i = x)}$$

Difference: instead of weighting more the group that represents more units, it weights more the group where the treatment status has higher variance.

What does it mean?

Effective sample

- From Angrist and Krueger (1999), Angrist and Pischke (2009), Aronow and Samii (2016), the following result holds:

$$\hat{\beta} \xrightarrow{p} \frac{E[w_i \tau_i]}{E[w_i]}, \text{ where } w_i = (D_i - E[D_i|X_i])^2$$

where

$$E[w_i|X_i] = E[(D_i - E[D_i|X_i])^2|X_i)] = Var[D_i|X_i]$$

- Conditional variance weighting equivalent to run the regression on an *effective* sample different from the one we think we are working with
- To characterize the effective sample we can estimate the w_is

Effective sample

$$E[w_i|X_i] = E[(D_i - E[D_i|X_i])^2|X_i] = Var[D_i|X_i]$$

- If we assume linearity of the treatment assignment in X_i, the weight is equal to the square of the residual from regressing the treatment indicator on X_i

Effective sample

$$E[w_i|X_i] = E[(D_i - E[D_i|X_i])^2|X_i] = Var[D_i|X_i]$$

- If we assume linearity of the treatment assignment in X_i, the weight is equal to the square of the residual from regressing the treatment indicator on X_i
- Higher conditional variance of treatment $\implies$ more variance not explained by the covariates $\implies$ higher error term

Effective sample

$$E[w_i|X_i] = E[(D_i - E[D_i|X_i])^2|X_i] = Var[D_i|X_i]$$

- If we assume linearity of the treatment assignment in X_i, the weight is equal to the square of the residual from regressing the treatment indicator on X_i
- Higher conditional variance of treatment $\implies$ more variance not explained by the covariates $\implies$ higher error term
- To estimate the regression weights:

Effective sample

$$E[w_i|X_i] = E[(D_i - E[D_i|X_i])^2|X_i] = Var[D_i|X_i]$$

- If we assume linearity of the treatment assignment in X_i, the weight is equal to the square of the residual from regressing the treatment indicator on X_i
- Higher conditional variance of treatment $\implies$ more variance not explained by the covariates $\implies$ higher error term
- To estimate the regression weights:
  - Run the regression $D_i = X_i\gamma + e_i$

Effective sample

$$E[w_i|X_i] = E[(D_i - E[D_i|X_i])^2|X_i] = Var[D_i|X_i]$$

- If we assume linearity of the treatment assignment in X_i, the weight is equal to the square of the residual from regressing the treatment indicator on X_i
- Higher conditional variance of treatment $\implies$ more variance not explained by the covariates $\implies$ higher error term
- To estimate the regression weights:
  - Run the regression $D_i = X_i\gamma + e_i$
  - Take residual $\hat{e}_i = D_i - X_i\hat{\gamma}$ and square it

Effective sample

- With the weights estimates, one can characterize the effective sample

Effective sample

- With the weights estimates, one can characterize the effective sample
- Covariate means in nominal sample: $\bar{Z}_i = \frac{1}{n}\sum_{i=1}^{n} Z_i$

Effective sample

- With the weights estimates, one can characterize the effective sample
- Covariate means in nominal sample: $\bar{Z}_i = \frac{1}{n}\sum_{i=1}^{n} Z_i$
- Covariate means in effective sample: $\hat{\mu}(Z_i) = \frac{\sum_{i=1}^{n} \hat{w}_i Z_i}{\sum_{i=1}^{n} \hat{w}_i}$

Application: weather and global warming beliefs

- Egan and Mullin (2012): how people form their attitudes based on personal experiences

Application: weather and global warming beliefs

- Egan and Mullin (2012): how people form their attitudes based on personal experiences
- Use local weather variation to estimate the effect of experiencing weather changes on beliefs about global warming

Application: weather and global warming beliefs

- Egan and Mullin (2012): how people form their attitudes based on personal experiences
- Use local weather variation to estimate the effect of experiencing weather changes on beliefs about global warming
- We want to characterize the effective sample

Application: weather and global warming beliefs

- Egan and Mullin (2012): how people form their attitudes based on personal experiences
- Use local weather variation to estimate the effect of experiencing weather changes on beliefs about global warming
- We want to characterize the effective sample
- Ask where weather is most variable (conditional on covariates)

Application: weather and global warming beliefs

```r
# Import the data
library(haven)
d <- read_dta("gwdataset.dta")

# Import state IDs
zips <- read_dta("zipcodetostate.dta")
zips <- zips %>% select(c(statenum, statefromzipfile)) %>% unique()
zips <- zips %>% filter(!(statenum == 8 & statefromzipfile == "NY"))

# Import population data
pops <- read.csv("population_ests_2013.csv")

# Format
pops$state <- tolower(pops$NAME)
d$getwarmord <- as.double(d$getwarmord)
```

Application: weather and global warming beliefs

```r
# Estimate primary model of interest:
d$doi <- factor(d$doi)
d$statenum <- factor(d$statenum)
d$wbnid_num <- factor(d$wbnid_num)
Y <- "getwarmord"
D <- "ddt_week"
X <- names(d)[c(15,17,42:72)]
reg_formula <- paste0(Y, "~", D, "+", paste0(X, collapse = "+"))
reg_out <- lm(as.formula(reg_formula), d)

# Or
out <- lm(getwarmord~ddt_week+educ_hsless+educ_coll+educ_postgrad+
          educ_dk+party_rep+party_leanrep+party_leandem+
          party_dem+male+raceeth_black+raceeth_hisp+
          raceeth_notwbh+raceeth_dkref+age_1824+age_2534+
          age_3544+age_5564+age_65plus+age_dk+ideo_vcons+
          ideo_conservative+ideo_liberal+ideo_vlib+ideo_dk+
          attend_1+attend_2+attend_3+attend_5+attend_6+
          attend_9+as.factor(doi)+as.factor(statenum)+
          as.factor(wbnid_num),d)
```

Base Model

```
summary(reg_out)$coefficients[1:10,]
```

```
##                 Estimate  Std. Error   t value   Pr(>|t|)
## (Intercept)   1.945740062 0.771478843 2.5220913 0.01169077
## ddt_week      0.004857915 0.002475887 1.9620908 0.04979656
## wbnid_num3103 0.843451519 0.922666490 0.9141456 0.36067588
## wbnid_num3154 1.575071541 0.973391215 1.6181280 0.10568587
## wbnid_num3159 1.903629413 1.021302199 1.8639237 0.06237963
## wbnid_num3804 1.406498119 0.794035963 1.7713280 0.07655528
## wbnid_num3810 1.330878449 0.806312016 1.6505750 0.09887602
## wbnid_num3811 1.082204367 0.798796489 1.3547936 0.17553267
## wbnid_num3812 1.219327925 0.803974284 1.5166255 0.12941222
## wbnid_num3813 0.986084952 0.829563706 1.1886790 0.23461152
```

Estimate the weights

```r
# Regress treatment indicator on the vector of covariates
D_formula <- paste0(D, "~", paste0(X, collapse = "+"))
outD <- lm(as.formula(D_formula), d)

# Extract the residuals and take their square
eD2 <- residuals(outD)^2
```

Effective sample statistics

```
# Take some relevant variables
compare_samples<- d[, c("wave", "ddt_week", "ddt_twoweeks",
  "ddt_threeweeks", "party_rep", "attend_1", "ideo_conservative",
  "age_1824", "educ_hsless")]

# Compute statistics with and without weights
compare_samples <- t(apply(compare_samples,2,function(x)
  c(mean(x),sd(x),weighted.mean(x,eD2),
    sqrt(weighted.mean((x-weighted.mean(x,eD2))^2,eD2)))))
colnames(compare_samples) <- c("Nominal Mean", "Nominal SD",
      "Effective Mean", "Effective SD")
```

Effective Sample Statistics

```
compare_samples
```

```
##                     Nominal Mean Nominal SD Effective Mean Effective SD
## wave                  3.09693726  1.4252527     3.20788200    1.5609143
## ddt_week              3.83548593  5.9047249     5.11579140   10.8980228
## ddt_twoweeks          3.85505617  5.4572382     5.00137435    9.2262827
## ddt_threeweeks        3.96719696  4.7689594     5.10859485    8.4348180
## party_rep             0.29527208  0.4561989     0.28978321    0.4536617
## attend_1              0.11433244  0.3182383     0.12343459    0.3289354
## ideo_conservative     0.31132917  0.4630715     0.29325249    0.4552532
## age_1824              0.07195956  0.2584402     0.06881146    0.2531333
## educ_hsless           0.34151056  0.4742516     0.31219962    0.4633908
```

Effective sample maps

```r
# Construct the "effective sample weights" for each state
wts_by_state <- tapply(eD2, d$statenum, sum)
wts_by_state <- wts_by_state/sum(wts_by_state)*100
wts_by_state <- data.frame(eff = wts_by_state,
                           statenum = as.numeric(names(wts_by_state)))

# Merge to the state name variable
data_for_map <- merge(wts_by_state, zips, by="statenum")

# Construct the "nominal sample weights" for each state
wts_by_state <- tapply(rep(1,6726),d$statenum,sum)
wts_by_state <- wts_by_state/sum(wts_by_state)*100
wts_by_state <- data.frame(nom = wts_by_state,
                           statenum = as.numeric(names(wts_by_state)))

# Add to the other data
data_for_map <- merge(data_for_map, wts_by_state, by="statenum")
```

Effective sample maps

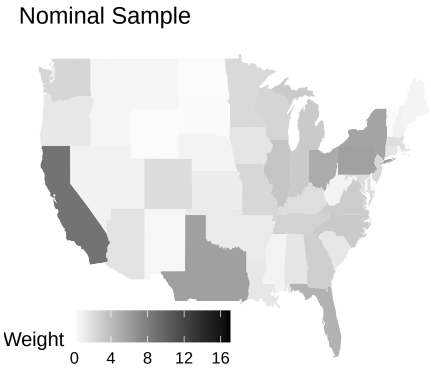
```
# Get correct state names
require(maps,quietly=TRUE)
data(state.fips)

# Add them to the dataset
data_for_map <- left_join(data_for_map, state.fips,
                          by = c("statefromzipfile" = "abb"))

# More data prep
data_for_map$state <- sapply(as.character(data_for_map$polyname),
                             function(x)strsplit(x,":")[[1]][1])
data_for_map <- data_for_map %>% group_by(statefromzipfile) %>%
  summarise_all(first) %>% ungroup() %>% select(-polyname)

# Diff between nominal and effective weights
data_for_map$diff <- data_for_map$eff - data_for_map$nom

# Merge with population data
data_for_map <- left_join(data_for_map, pops, by="state")

# Actual "weight" of each state in the US
data_for_map$pop_pct <- data_for_map$POPESTIMATE2013/sum(
  data_for_map$POPESTIMATE2013)*100

# Different representativity of the two samples
data_for_map <- mutate(data_for_map,
                       pop_diff_eff = eff - pop_pct,
                       pop_diff_nom = nom - pop_pct)
data_for_map <- mutate(data_for_map,
                       pop_diff = pop_diff_eff - pop_diff_nom)

require(ggplot2,quietly=TRUE)
state_map <- map_data("state")
```

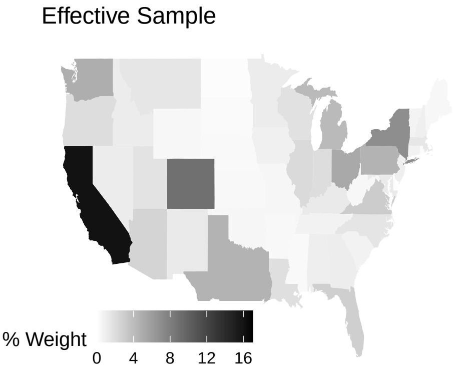
More setup

```r
# Plot the weights in each sample
plot_eff <- ggplot(data_for_map, aes(map_id = state)) +
  geom_map(aes(fill=eff), map = state_map) +
  expand_limits(x= state_map$long, y = state_map$lat) +
  scale_fill_continuous("% Weight", limits=c(0,17), low="white", high="black") +
  labs(title = "Effective Sample") +
  theme(legend.position=c(.2,.1),legend.direction = "horizontal",
        axis.line = element_blank(), axis.text = element_blank(),
        axis.ticks = element_blank(), axis.title = element_blank(),
        panel.background = element_blank(),
        plot.background = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank())


plot_nom <- ggplot(data_for_map, aes(map_id = state)) +
  geom_map(aes(fill=nom), map = state_map) +
  expand_limits(x=state_map$long, y=state_map$lat) +
  scale_fill_continuous("% Weight", limits=c(0,17), low="white", high="black") +
  labs(title="Nominal Sample") +
  theme(legend.position=c(.2,.1),legend.direction = "horizontal",
        axis.line = element_blank(), axis.text = element_blank(),
        axis.ticks = element_blank(), axis.title = element_blank(),
        panel.background = element_blank(),
        plot.background = element_blank(),
        panel.border = element_blank(), panel.grid = element_blank())
```

Maps

```
require(gridExtra,quietly=TRUE)
grid.arrange(plot_nom,plot_eff,ncol=2)
```



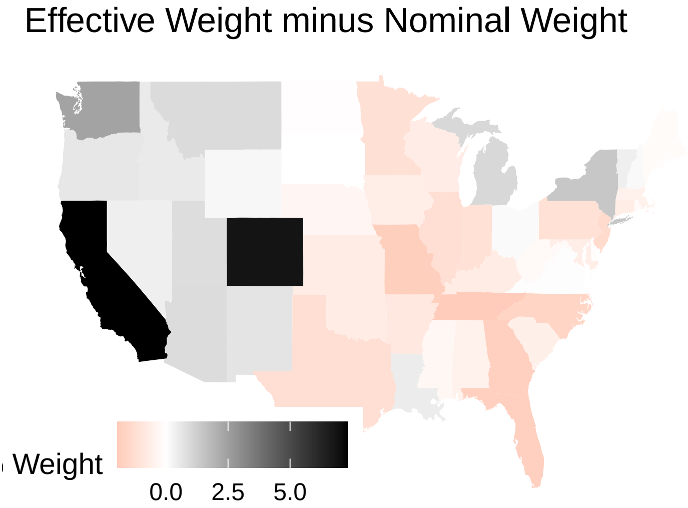


Setup comparison plot

```
plot_diff <- ggplot(data_for_map,aes(map_id=state)) +
  geom_map(aes(fill=diff), map = state_map) +
  expand_limits(x = state_map$long, y = state_map$lat) +
  scale_fill_gradient2("% Weight", low = "red", mid = "white", high = "blac
  labs(title = "Effective Weight minus Nominal Weight") +
  theme(legend.position=c(.2,.1),legend.direction = "horizontal",
        axis.line = element_blank(), axis.text = element_blank(),
        axis.ticks = element_blank(), axis.title = element_blank(),
        panel.background = element_blank(),
        plot.background = element_blank(),
        panel.border = element_blank(), panel.grid = element_blank())
```

Difference in weights

```
plot_diff
```



Sensitivity

- As all assumptions, CIA is untestable

Sensitivity

- As all assumptions, CIA is untestable
- But we can study how results could be affected by hypothetical departures from it, i.e. how *sensitive* they are

Sensitivity

- As all assumptions, CIA is untestable
- But we can study how results could be affected by hypothetical departures from it, i.e. how *sensitive* they are
- Focus on the strategy proposed by Cinelli and Hazlett (2020), implemented through `sensemakr` (in R and Stata)

Sensitivity

- As all assumptions, CIA is untestable
- But we can study how results could be affected by hypothetical departures from it, i.e. how *sensitive* they are
- Focus on the strategy proposed by Cinelli and Hazlett (2020), implemented through `sensemakr` (in R and Stata)
- Turns out to be useful to test the "threats" to our assumption

Sensitivity in the OVB framework

Basic framework:

- A linear model $Y = \tau D + X'\beta + \gamma Z + \varepsilon_{full}$, where D is the treatment, X are observed controls and Z are unobserved controls

Sensitivity in the OVB framework

Basic framework:

- A linear model $Y = \tau D + X'\beta + \gamma Z + \varepsilon_{full}$, where D is the treatment, X are observed controls and Z are unobserved controls
- The researcher can only estimate $Y = \tau_{res} D + X'\beta_{res} + \varepsilon_{res}$

Sensitivity in the OVB framework

Basic framework:

- A linear model $Y = \tau D + X'\beta + \gamma Z + \varepsilon_{full}$, where D is the treatment, X are observed controls and Z are unobserved controls
- The researcher can only estimate $Y = \tau_{res} D + X'\beta_{res} + \varepsilon_{res}$
- We know that $\hat{\tau}_{res} = \hat{\tau} + \hat{\gamma}\hat{\delta} = \hat{\tau} + Bias$, where $\hat{\delta} = \frac{Cov(\tilde{D},\tilde{Z})}{V(\tilde{D})}$

Sensitivity in the OVB framework

Basic framework:

- A linear model $Y = \tau D + X'\beta + \gamma Z + \varepsilon_{full}$, where D is the treatment, X are observed controls and Z are unobserved controls
- The researcher can only estimate $Y = \tau_{res} D + X'\beta_{res} + \varepsilon_{res}$
- We know that $\hat{\tau}_{res} = \hat{\tau} + \hat{\gamma}\hat{\delta} = \hat{\tau} + Bias$, where $\hat{\delta} = \frac{Cov(\tilde{D}, \tilde{Z})}{V(\tilde{D})}$
- In essence, what we do is to give values to the *Bias* and study how much the estimate of $\hat{\tau}_{res}$ change

Sensitivity in the OVB framework

Basic framework:

- A linear model $Y = \tau D + X'\beta + \gamma Z + \varepsilon_{full}$, where D is the treatment, X are observed controls and Z are unobserved controls
- The researcher can only estimate $Y = \tau_{res} D + X'\beta_{res} + \varepsilon_{res}$
- We know that $\hat{\tau}_{res} = \hat{\tau} + \hat{\gamma}\hat{\delta} = \hat{\tau} + Bias$, where $\hat{\delta} = \frac{Cov(\tilde{D},\tilde{Z})}{V(\tilde{D})}$
- In essence, what we do is to give values to the *Bias* and study how much the estimate of $\hat{\tau}_{res}$ change
- Now, note that the Bias has two components:

Sensitivity in the OVB framework

Basic framework:

- A linear model $Y = \tau D + X'\beta + \gamma Z + \varepsilon_{full}$, where D is the treatment, X are observed controls and Z are unobserved controls
- The researcher can only estimate $Y = \tau_{res} D + X'\beta_{res} + \varepsilon_{res}$
- We know that $\hat{\tau}_{res} = \hat{\tau} + \hat{\gamma}\hat{\delta} = \hat{\tau} + Bias$, where $\hat{\delta} = \frac{Cov(\tilde{D},\tilde{Z})}{V(\tilde{D})}$
- In essence, what we do is to give values to the *Bias* and study how much the estimate of $\hat{\tau}_{res}$ change
- Now, note that the Bias has two components:
  - $\hat{\gamma}$: the *impact* of Z on the outcome

Sensitivity in the OVB framework

Basic framework:

- A linear model $Y = \tau D + X'\beta + \gamma Z + \varepsilon_{full}$, where D is the treatment, X are observed controls and Z are unobserved controls
- The researcher can only estimate $Y = \tau_{res} D + X'\beta_{res} + \varepsilon_{res}$
- We know that $\hat{\tau}_{res} = \hat{\tau} + \hat{\gamma}\hat{\delta} = \hat{\tau} + Bias$, where $\hat{\delta} = \frac{Cov(\tilde{D},\tilde{Z})}{V(\tilde{D})}$
- In essence, what we do is to give values to the *Bias* and study how much the estimate of $\hat{\tau}_{res}$ change
- Now, note that the Bias has two components:
  - $\hat{\gamma}$: the *impact* of Z on the outcome
  - $\hat{\delta}$: the *imbalance* of Z across treated/control groups

Sensitivity in the OVB framework

- It is convenient to rewrite OVB in terms of partial R-squared/correlations

Sensitivity in the OVB framework

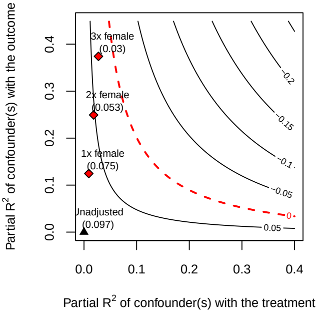
- It is convenient to rewrite OVB in terms of partial R-squared/correlations
- Allows for non-linearities in the effects of confounders and in assessing the sensitivity of the standard errors

Sensitivity in the OVB framework

- It is convenient to rewrite OVB in terms of partial R-squared/correlations
- Allows for non-linearities in the effects of confounders and in assessing the sensitivity of the standard errors
- See the technical details in the journal article and examples in the sensemakr website

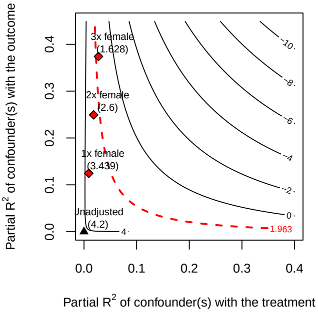
Cinelli and Hazlett (2020)

```
library(sensemakr)
data("darfur")

# Run regression model
model <- lm(peacefactor ~ directlyharmed + age + farmer_dar + herder_dar + pastvoted + hhsize_darfur + female +
              village, data = darfur)

# Run sensitivity analysis
sensitivity <- sensemakr(model, treatment = "directlyharmed", benchmark_covariates = "female", kd = 1:3)

# Results description
# sensitivity

# Plot
plot(sensitivity)
```



Cinelli and Hazlett (2020)

```
# Plot
plot(sensitivity, sensitivity.of = "t-value")
```



Regression problems

The linear approximation can create problems if the CEF is non-linear

Remember the notion of regression adjustment

$$E[Y_{0i}|X_i] = \alpha_0 + \beta_0(X_i - \bar{X})$$
$$E[Y_{1i}|X_i] = \alpha_1 + \beta_1(X_i - \bar{X})$$

The estimate of τ_{PATE} is $\hat{\tau}_{reg} = \hat{\alpha}_1 - \hat{\alpha}_0$, or the coefficient of D in the regression $Y_i = \alpha + \tau_{reg} D_i + \gamma(X_i - \bar{X}) + \delta(D_i * (X_i - \bar{X})) + \varepsilon_i$

Regression problems

With some algebra, we can obtain another decomposition (see also Imbens and Rubin (Ch. 12) and Imbens and Woldridge (2009))

$$\hat{\tau}_{reg} = \underbrace{\bar{Y}_1 - \bar{Y}_0}_{\text{Difference in means}} \underbrace{- (\frac{N_0}{N_1 + N_0}\hat{\beta}_1 + \frac{N_1}{N_1 + N_0}\hat{\beta}_0)(\bar{X}_1 - \bar{X}_0)}_{\text{Regression adjustment}}$$

Regression problems

With some algebra, we can obtain another decomposition (see also Imbens and Rubin (Ch. 12) and Imbens and Woldridge (2009))

$$\hat{\tau}_{reg} = \underbrace{\bar{Y}_1 - \bar{Y}_0}_{\text{Difference in means}} \underbrace{- (\frac{N_0}{N_1 + N_0}\hat{\beta}_1 + \frac{N_1}{N_1 + N_0}\hat{\beta}_0)(\bar{X}_1 - \bar{X}_0)}_{\text{Regression adjustment}}$$

Another way to see it:

$$\hat{\tau}_{reg} = \frac{N_1}{N_1 + N_0}\hat{\tau}_1 + \frac{N_0}{N_1 + N_0}\hat{\tau}_0$$

$$\hat{\tau}_1 = \bar{Y}_1 - \bar{Y}_0 - (\bar{X}_1 - \bar{X}_0)\hat{\beta}_0$$

$$\hat{\tau}_0 = \bar{Y}_1 - \bar{Y}_0 - (\bar{X}_1 - \bar{X}_0)\hat{\beta}_1$$

Regression problems

With some algebra, we can obtain another decomposition (see also Imbens and Rubin (Ch. 12) and Imbens and Woldridge (2009))

$$\hat{\tau}_{reg} = \underbrace{\bar{Y}_1 - \bar{Y}_0}_{\text{Difference in means}} \underbrace{-\left(\frac{N_0}{N_1 + N_0}\hat{\beta}_1 + \frac{N_1}{N_1 + N_0}\hat{\beta}_0\right)(\bar{X}_1 - \bar{X}_0)}_{\text{Regression adjustment}}$$

Another way to see it:

$$\hat{\tau}_{reg} = \frac{N_1}{N_1 + N_0}\hat{\tau}_1 + \frac{N_0}{N_1 + N_0}\hat{\tau}_0$$

$$\hat{\tau}_1 = \bar{Y}_1 - \bar{Y}_0 - (\bar{X}_1 - \bar{X}_0)\hat{\beta}_0$$

$$\hat{\tau}_0 = \bar{Y}_1 - \bar{Y}_0 - (\bar{X}_1 - \bar{X}_0)\hat{\beta}_1$$

Unobserved counterfactuals predicted with coefficients from the other (observed) group. Extrapolation if $\bar{X}_1 \neq \bar{X}_0$.

Extrapolation

Propensity score

- Balancing score: a function of the covariates that is a sufficient statistic for the covariates values in the treatment distribution

$$D_i \perp X_i | b(X_i)$$

- Several possible balancing scores, e.g. the covariates themselves - If treatment is independent of the POs conditional on X_i, it is also independent conditional on $b(X_i)$ (proof in Imbens and Rubin, p.267) - We are interested in scores that reduce the dimensionality of the conditioning set X_i

Propensity score

- The propensity score is defined as

$$e(X_i) = P(D_i = 1|X_i) = E(D_i|X_i)$$

- The propensity score is a balancing score (Imbens and Rubin, p.266)

$$D_i \perp X_i | e(X_i)$$

Weighting

Weighting units by their propensity score is theoretically appealing, because it gives an unbiased estimate for the PATE under CIA.

Weighting

Weighting units by their propensity score is theoretically appealing, because it gives an unbiased estimate for the PATE under CIA.

$$E\left[\frac{D_i Y_i}{e(X_i)} - \frac{(1-D_i)Y_i}{1-e(X_i)}\right] = E\left[\frac{D_i Y_{1i}}{e(X_i)} - \frac{(1-D_i)Y_{0i}}{1-e(X_i)}\right] =$$

$$= E\left[E\left[\frac{D_i Y_{1i}}{e(X_i)} - \frac{(1-D_i)Y_{0i}}{1-e(X_i)}\right]|X_i\right] = E\left[\frac{E[D_i|X_i]E[Y_{1i}|X_i]}{e(X_i)} - \frac{E[1-D_i|X_i]E[Y_{0i}|X_i]}{1-e(X_i)}\right] =$$

$$= E\left[\frac{e(X_i)E[Y_{1i}|X_i]}{e(X_i)} - \frac{(1-e(X_i))E[Y_{0i}|X_i]}{1-e(X_i)}\right] =$$

$$= E[E[Y_{1i} - Y_{0i}|X_i]] = E[Y_{1i} - Y_{0i}] = \tau_{PATE}$$

Weighting

A natural weighting estimator is the IPW estimator.

$$\hat{\tau} = \frac{1}{N}\sum_{i=1}^{N}\frac{D_i Y_i}{\hat{e}(X_i)} - \frac{1}{N}\sum_{i=1}^{N}\frac{(1-D_i)Y_i}{1-\hat{e}(X_i)}$$

We usually use a "normalized" version of it.

$$\hat{\tau}_{ipw} = \frac{\sum_{i=1}^{N}\frac{D_i Y_i}{\hat{e}(X_i)}}{\sum_{i=1}^{N}\frac{D_i}{\hat{e}(X_i)}} - \frac{\sum_{i=1}^{N}\frac{(1-D_i)Y_i}{(1-\hat{e}(X_i))}}{\sum_{i=1}^{N}\frac{(1-D_i)}{(1-\hat{e}(X_i))}}$$

PS shifts modeling issues from estimating $E[Y_i|X_i]$ to estimating $e(X_i)$.

Weighting

- Other weighting estimators do not model the probability of treatment but target covariate balance directly

Weighting

- Other weighting estimators do not model the probability of treatment but target covariate balance directly
- E.g. Covariate Balancing Propensity Score, Entropy Balancing

Weighting

- Other weighting estimators do not model the probability of treatment but target covariate balance directly
- E.g. Covariate Balancing Propensity Score, Entropy Balancing
- In R: all available in the package `WeightIt`

Weighting

- Other weighting estimators do not model the probability of treatment but target covariate balance directly
- E.g. Covariate Balancing Propensity Score, Entropy Balancing
- In R: all available in the package `WeightIt`
  - Other single packages: `CBPS`, `PSweight`, `Causalweight`

Weighting

- Other weighting estimators do not model the probability of treatment but target covariate balance directly
- E.g. Covariate Balancing Propensity Score, Entropy Balancing
- In R: all available in the package `WeightIt`
  - Other single packages: `CBPS`, `PSweight`, `Causalweight`
- Methods allowed for IPW: glm (default), non-parametric methods, SuperLearner, BART

Weighting

- Other weighting estimators do not model the probability of treatment but target covariate balance directly
- E.g. Covariate Balancing Propensity Score, Entropy Balancing
- In R: all available in the package `WeightIt`
  - Other single packages: `CBPS`, `PSweight`, `Causalweight`
- Methods allowed for IPW: glm (default), non-parametric methods, SuperLearner, BART
- In Stata: `teffects` performs IPW. `psweight` performs IPW and CBPS.

Weighting

Application to climate change opinions

```r
# Use WeightIt package
library(WeightIt)

# Binary indicator for treatment
d$treat <- ifelse(d$ddt_week>quantile(d$ddt_week)[4],1,0)

# Propensity score weighting
wd <- weightit(treat ~ educ_coll + party_leandem + as.factor(statenum) + as.factor(wbnid_num),
               data=d, estimand="ATT", method = "ps")

# Description
wd
```

```
## A weightit object
##  - method: "ps" (propensity score weighting)
##  - number of obs.: 6726
##  - sampling weights: none
##  - treatment: 2-category
##  - estimand: ATT (focal: 1)
##  - covariates: educ_coll, party_leandem, as.factor(statenum), as.factor(wbnid_num)
```

Weighting

```
# Describe the weights
summary(wd)
```

```
##                  Summary of weights
##
## - Weight ranges:
##
##         Min                              Max
## treated   1     ||                       1.00
## control   0 |---------------------------| 7.41
##
## - Units with 5 most extreme weights by group:
##
##              11      10      4    3    2
##  treated      1       1      1    1    1
##            3689    2250   1860 3820 3817
##  control 6.4225 6.5651 7.1274 7.41 7.41
##
## - Weight statistics:
##
##         Coef of Var   MAD Entropy # Zeros
## treated       0.000 0.000  -0.000       0
## control       1.402 0.947   0.735    1434
##
## - Effective Sample Sizes:
##
##            Control Treated
## Unweighted 5048.      1678
## Weighted   1702.62    1678
```

Weighting

```
# Weighted difference in means
library(estimatr)
lm_robust(getwarmord ~ treat, weights = wd$weights, data=d) %>% summary()
```

```
##
## Call:
## lm_robust(formula = getwarmord ~ treat, data = d, weights = wd$weights)
##
## Weighted, Standard error type:  HC2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|) CI Lower CI Upper   DF
## (Intercept)  2.52968    0.01883 134.336   0.0000 2.492770   2.5666 6724
## treat        0.05434    0.02652   2.049   0.0405 0.002351   0.1063 6724
##
## Multiple R-squared:  0.001205 ,  Adjusted R-squared:  0.001056
## F-statistic: 4.198 on 1 and 6724 DF,  p-value: 0.0405
```

```
# Bootstrap SE
# library("boot"); set.seed(1)
# est.fun <- function(data, index) {
#   W.out <- weightit(treat ~ educ_coll + party_leandem + as.factor(statenum) + as.factor(wbnid_num),
#                     data=data[index,], estimand="ATT", method = "ps")
#   fit <- glm(getwarmord ~ treat, data = data[index,], weights = W.out$weights)
#   return(coef(fit)["treat"])
# }
# boot.out <- boot(est.fun, data = d, R = 999)
# boot.ci(boot.out, type = "basic")
```

Combining weighting and regression

- Doubly-robust estimators: combine estimation of treatment assignment (PS) and CEF (regression)
- Consistent if any of the two is misspecified (not both)

Combining weighting and regression

- Doubly-robust estimators: combine estimation of treatment assignment (PS) and CEF (regression)
- Consistent if any of the two is misspecified (not both)
- Standard one: AIPW. Augments IPW with predicted outcomes from regressions on treated and control groups

$$\hat{\tau}_{aipw} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{D_i Y_i}{\hat{e}(X_i)} - \frac{\hat{Y}_{1i}(D_i - \hat{e}(X_i))}{\hat{e}(X_i)}\right) -$$

$$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{(1-D_i)Y_i}{1-\hat{e}(X_i)} + \frac{\hat{Y}_{0i}(D_i - \hat{e}(X_i))}{1-\hat{e}(X_i)}\right) =$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(\hat{Y}_{1i} + \frac{D_i(Y_i - \hat{Y}_{1i})}{\hat{e}(X_i)}\right) -$$

$$\frac{1}{N}\sum_{i=1}^{N}\left(\hat{Y}_{0i} + \frac{(1-D_i)(Y_i - \hat{Y}_{0i})}{1-\hat{e}(X_i)}\right)$$

Matching

- Non-parametric methods for causal effects under CIA

Matching

- Non-parametric methods for causal effects under CIA
- Approximates an experiment with block-randomization

Matching

- Non-parametric methods for causal effects under CIA
- Approximates an experiment with block-randomization
- Let's focus on the ATT as an estimand, because it requires weaker assumptions than CIA to be identified

Matching

- Non-parametric methods for causal effects under CIA
- Approximates an experiment with block-randomization
- Let's focus on the ATT as an estimand, because it requires weaker assumptions than CIA to be identified
- CMI: $E[Y_{0i}|D_i = 1, X_i] = E[Y_{0i}|D_i = 0, X_i], P(D_i = 1|X_i) < 1$

Matching

- Non-parametric methods for causal effects under CIA
- Approximates an experiment with block-randomization
- Let's focus on the ATT as an estimand, because it requires weaker assumptions than CIA to be identified
- CMI: $E[Y_{0i}|D_i = 1, X_i] = E[Y_{0i}|D_i = 0, X_i], P(D_i = 1|X_i) < 1$
- Have to care about counterfactuals for treated units and not for control units

Matching

- Exact matching:

1. For each treated unit, find control units with same values of X_i
2. Compute the difference in means within these strata and compute a weighted average using the distribution of X of the treated group

- In practice, this is done by targeting balance wrt the covariate distribution of the treated group

Matching

- Several matching algorithms available

Matching

- Several matching algorithms available
- Select close units by minimizing some function of the covariates

Matching

- Several matching algorithms available
- Select close units by minimizing some function of the covariates
- Generally create a matched sample of units, on which one can estimate ATT/ATE

Matching

- Several matching algorithms available
- Select close units by minimizing some function of the covariates
- Generally create a matched sample of units, on which one can estimate ATT/ATE
- Researchers can try different matching algorithms until an acceptable level of balance is achieved

Matching

- Several matching algorithms available
- Select close units by minimizing some function of the covariates
- Generally create a matched sample of units, on which one can estimate ATT/ATE
- Researchers can try different matching algorithms until an acceptable level of balance is achieved
- Useful diagnostics: Kolmogorov-Smirnov tests for equality of distributions, quantile-quantile plots

Matching

- R package `MatchIt` performs exact matching, genetic, CEM, and NN

Matching

- R package `MatchIt` performs exact matching, genetic, CEM, and NN
- Different distance metrics allowed

Matching

- R package `MatchIt` performs exact matching, genetic, CEM, and NN
- Different distance metrics allowed
- Other single packages: `Matching`, `GenMatch`, `cem`

Matching

- R package `MatchIt` performs exact matching, genetic, CEM, and NN
- Different distance metrics allowed
- Other single packages: `Matching`, `GenMatch`, `cem`
- In Stata: `teffects` for NN and PS-matching. `cem` for CEM.

Matching

Improve balance with matching. Ruggeri, Dorussen, and Gizelis (2016)

```
# Import the data
library(haven)
data <- read_dta("matchingdata.dta")

# Keep non-missing values
data <- na.omit(data)

# Treatment distribution
library(janitor)
tabyl(data$PKO)
```

```
##  data$PKO     n    percent
##         0 13051 0.94799157
##         1   716 0.05200843
```

Matching

```r
# Use MatchIt package
library(MatchIt)

# Nearest-neighbor matching on the propensity score
nn_match <- matchit(PKO ~ avgttime + avgadjimr + popgpw2000_40 + avgmnt +  borddist + capdist + prec_new, metho

nn_match
```

```
## A matchit object
##  - method: 1:1 nearest neighbor matching without replacement
##  - distance: Propensity score
##              - estimated with logistic regression
##  - number of obs.: 13767 (original), 1432 (matched)
##  - target estimand: ATT
##  - covariates: avgttime, avgadjimr, popgpw2000_40, avgmnt, borddist, capdist, prec_new
```

```r
# Matched sample
data_nn <- match.data(nn_match)

# Matched data
tabyl(data_nn$PKO)
```

```
##  data_nn$PKO   n percent
##            0 716     0.5
##            1 716     0.5
```

Matching

```r
# Compare balance in raw and matched data
summary(nn_match)$sum.all[,1:3]
```

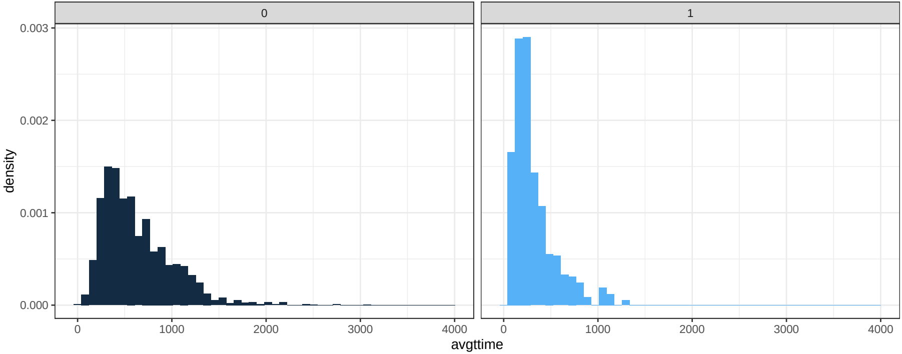
```
##                Means Treated Means Control Std. Mean Diff.
## distance        1.886561e-01  4.451170e-02       0.9039073
## avgttime        3.179041e+02  6.365755e+02      -1.4044072
## avgadjimr       1.402006e+02  1.309813e+02       0.3937550
## popgpw2000_40   5.282647e+05  1.842708e+05       0.6475519
## avgmnt          1.908064e-01  1.039625e-01       0.2943259
## borddist        1.417071e+02  1.988932e+02      -0.3827070
## capdist         5.797424e+02  8.136026e+02      -0.4359644
## prec_new        1.769303e+03  1.356781e+03       0.4541444
```

```r
summary(nn_match)$sum.matched[,1:3]
```

```
##                Means Treated Means Control Std. Mean Diff.
## distance        1.886561e-01  1.820085e-01      0.04168650
## avgttime        3.179041e+02  3.352170e+02     -0.07629930
## avgadjimr       1.402006e+02  1.454425e+02     -0.22388296
## popgpw2000_40   5.282647e+05  4.773519e+05      0.09584090
## avgmnt          1.908064e-01  1.731964e-01      0.05968288
## borddist        1.417071e+02  1.571460e+02     -0.10332192
## capdist         5.797424e+02  5.044745e+02      0.14031509
## prec_new        1.769303e+03  1.822873e+03     -0.05897540
```

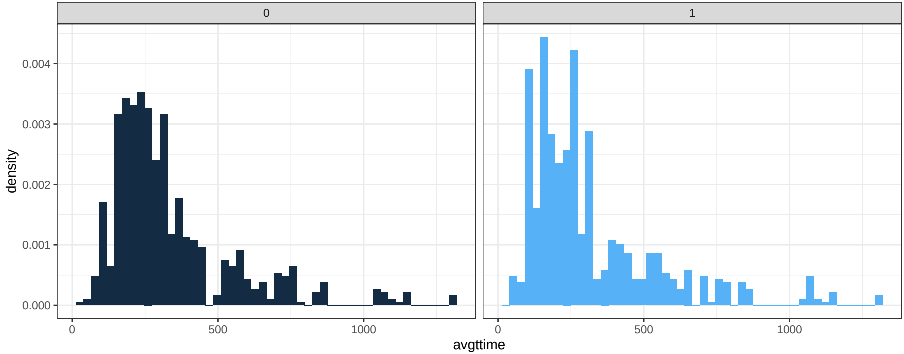
Matching

```r
library(ggplot2)
# Raw
ggplot(data, aes(x=avgttime, y = ..density.., fill=PKO)) +
  geom_histogram(bins=50) + facet_wrap(~PKO)+ theme_bw() +
  theme(legend.position="none")
```



Matching

```r
# Matched
ggplot(data_nn, aes(x=avgttime, y = ..density.., fill=PKO)) +
  geom_histogram(bins=50) + facet_wrap(~PKO)+ theme_bw() +
  theme(legend.position="none")
```




```r
# Larger set of diagnostic plots provided:
# plot(nn_match)
```