Quant 2, Lab 4

Pitfalls of control strategies continued: Effective Samples, Specification Error/Double ML

Sylvan Zheng

Effective Samples Intuition

- Consider a unit whose D is perfectly explained by X

Effective Samples Intuition

- Consider a unit whose D is perfectly explained by X
- Then, does this unit help identify the effect of $D|X$ on Y?

Effective Samples Intuition

- Consider a unit whose D is perfectly explained by X
- Then, does this unit help identify the effect of $D|X$ on Y?

Effective Samples Intuition

- Consider a unit whose D is perfectly explained by X
- Then, does this unit help identify the effect of $D|X$ on Y?

```
set.seed(12)
N <- 1000
X <- rnorm(N)
D <- X + rbinom(N, size = 1, prob = 0.10) * rnorm(N)
Y <- D + X + rnorm(N)
df <- data.frame(X = X, D = D, Y = Y)
```

Effective Samples Intuition

- Consider a unit whose D is perfectly explained by X
- Then, does this unit help identify the effect of D|X on Y?

```
set.seed(12)
N <- 1000
X <- rnorm(N)
D <- X + rbinom(N, size = 1, prob = 0.10) * rnorm(N)
Y <- D + X + rnorm(N)
df <- data.frame(X = X, D = D, Y = Y)
```

- For most of the sample, $X = D$

```
table(df$X == df$D)
```

```
##
## FALSE  TRUE
##   113   887
```

Effective Samples: Intuition

```r
models <- list(
    feols(Y ~ D + X, data = df),
    feols(Y ~ D + X, data = df %>% filter(df$X != df$D))
)
```

Dependent Variable:	Y	
Model:	(1)	(2)
Variables		
Constant	-0.0192	0.0570
	(0.0336)	(0.0961)
D	1.019***	1.015***
	(0.0936)	(0.0901)
X	0.9519***	0.9995***
	(0.1022)	(0.1384)
Fit statistics		
Observations	1,000	113

IID standard-errors in parentheses

Effective Sample Weights

- Observations where D is already "explained" by X are basically useless

Effective Sample Weights

- Observations where D is already "explained" by X are basically useless
- Effective Sample Weights formalize this intuition

Effective Sample Weights

- Observations where D is already "explained" by X are basically useless
- Effective Sample Weights formalize this intuition
- The sample weight for unit i is $w_i = (D_i - E[D_i|X_i])^2$

Effective Sample Weights

- Observations where D is already "explained" by X are basically useless
- Effective Sample Weights formalize this intuition
- The sample weight for unit i is $w_i = (D_i - E[D_i|X_i])^2$
  - If we assume linearity of the treatment assignment in X_i, then easy to construct

Effective Sample Weights

- Observations where D is already "explained" by X are basically useless
- Effective Sample Weights formalize this intuition
- The sample weight for unit i is $w_i = (D_i - E[D_i|X_i])^2$
  - If we assume linearity of the treatment assignment in X_i, then easy to construct
  - Run the regression $D_i = X_i\gamma + e_i$

Effective Sample Weights

- Observations where D is already "explained" by X are basically useless
- Effective Sample Weights formalize this intuition
- The sample weight for unit i is $w_i = (D_i - E[D_i|X_i])^2$
  - If we assume linearity of the treatment assignment in X_i, then easy to construct
  - Run the regression $D_i = X_i\gamma + e_i$
  - Take residual $\hat{e}_i = D_i - X_i\hat{\gamma}$ and square it

What can you do with the sample weights?

- Use them to characterize your “effective” sample

What can you do with the sample weights?

- Use them to characterize your "effective" sample
- Maybe some substantive interpretation

What can you do with the sample weights?

- Use them to characterize your “effective” sample
- Maybe some substantive interpretation
- Eg, original sample is “representative”. Is effective sample “representative?”

What can you do with the sample weights?

- Use them to characterize your "effective" sample
- Maybe some substantive interpretation
- Eg, original sample is "representative". Is effective sample "representative?"
- Interpretation on the research setting. Controlling for confounders almost mechanically makes the effective sample non representative.

Example: Egan/Mullin 2012

Turning Personal Experience into Political Attitudes: The Effect of Local Weather on Americans' Perceptions about Global Warming

Patrick J. Egan New York University
Megan Mullin Temple University

How do people translate their personal experiences into political attitudes? It has been difficult to explore this question using observational data, because individuals are typically exposed to experiences in a selective fashion, and self-reports of exposure may be biased and unreliable. In this study, we identify one experience to which Americans are exposed nearly at random—their local weather—and show that weather patterns have a significant effect on people's beliefs about the evidence for global warming.

Example: Egan/Mullin 2012

- Outcome variable `getwarmord` (climate change attitudes)

Example: Egan/Mullin 2012

- Outcome variable `getwarmord` (climate change attitudes)
- Treatment variable `ddtweek` (change in change in temp)

Example: Egan/Mullin 2012

- Outcome variable `getwarmord` (climate change attitudes)
- Treatment variable `ddtweek` (change in change in temp)

Example: Egan/Mullin 2012

- Outcome variable `getwarmord` (climate change attitudes)
- Treatment variable `ddtweek` (change in change in temp)

```
out.d <- feols(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 | doi + state + wbnid_num, d)
# Extract the residuals and take their square
d$wts <- residuals(out.d)^2
```

Example: Egan/Mullin 2012

- Suppose we want to characterize the sample contribution by state

Example: Egan/Mullin 2012

- Suppose we want to characterize the sample contribution by state
- "Nominal" weight: just the (normalized) number of observations per state

Example: Egan/Mullin 2012

- Suppose we want to characterize the sample contribution by state
- "Nominal" weight: just the (normalized) number of observations per state

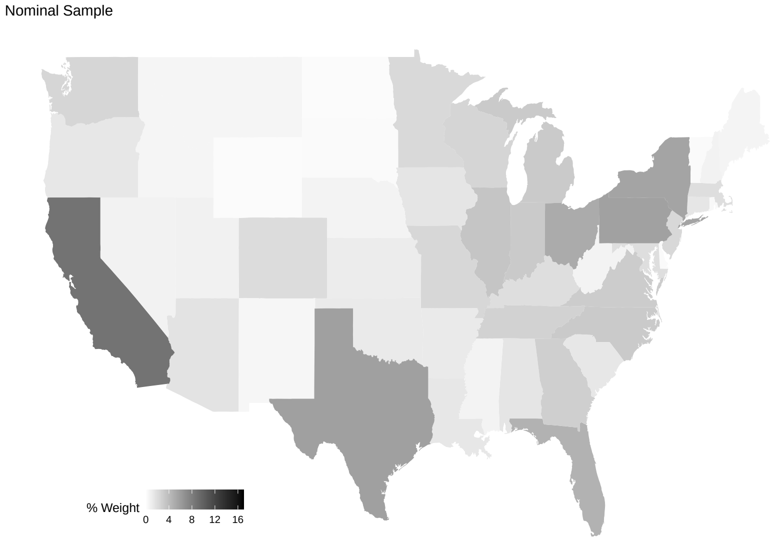
Example: Egan/Mullin 2012

- Suppose we want to characterize the sample contribution by state
- "Nominal" weight: just the (normalized) number of observations per state

```r
nom_map <- theme_state_map(d %>%
    group_by(state) %>%
    summarize(nom = n() * 100 / nrow(d)) %>%
    ggplot(aes(map_id = state)) +
    geom_map(aes(fill = nom), map = state_map) +
    labs(title = "Nominal Sample"))
```

Example: Egan/Mullin 2012

```
nom_map
```



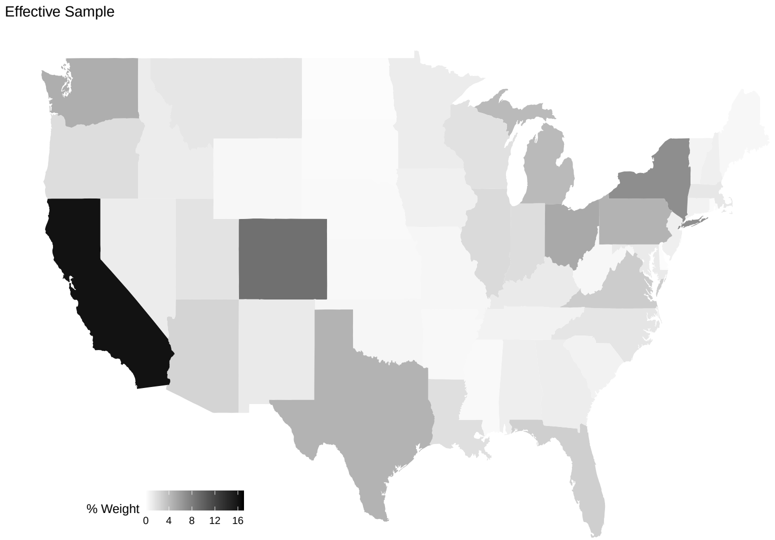
Example: Egan/Mullin 2012

- To characterize the "effective" contribution of each state, use the effective sample weight instead

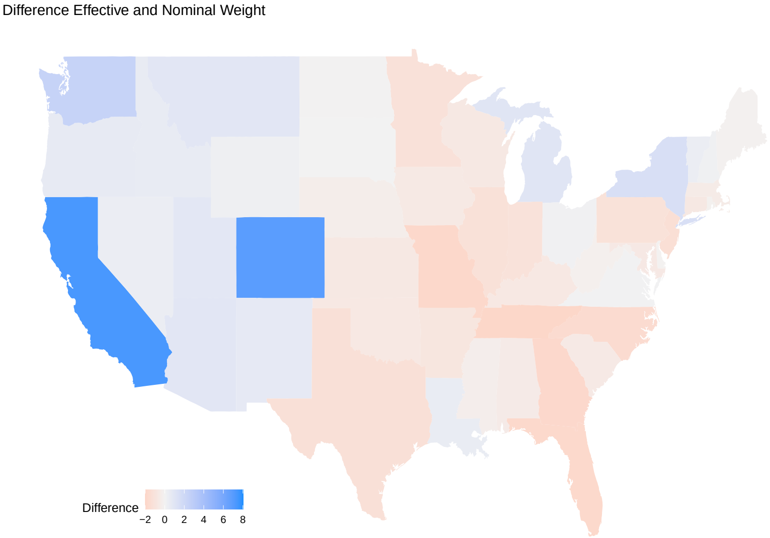
```
eff_map <- theme_state_map(d %>%
    group_by(state) %>%
    summarize(eff = sum(wts) * 100 / sum(d$wts)) %>%
    ggplot(aes(map_id = state)) +
    geom_map(aes(fill = eff), map = state_map) +
    labs(title = "Effective Sample"))
```

Example: Egan/Mullin 2012

```
eff_map
```



Example: Egan/Mullin 2012



Frisch-Waugh-Lovell theorem

- 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.

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

Frisch-Waugh-Lovell theorem

- 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

Frisch-Waugh-Lovell theorem

- 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

Frisch-Waugh-Lovell theorem

- 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

Frisch-Waugh-Lovell theorem

- 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
  - "The part of Y unexplained by X_{not_k}" $\sim$ "The part of X_k unexplained by X_{not_k}"

Frisch-Waugh-Lovell theorem

- 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
  - "The part of Y unexplained by X_{not_k}" $\sim$ "The part of X_k unexplained by X_{not_k}"
  - 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

FWL in R

```
set.seed(123)
N <- 1000
X <- rnorm(N, mean = 0, sd = 1)
# Generate binary treatment D, making D and X correlated
D <- rbinom(N, size = 1, prob = plogis(X))
Y <- 2 * D + 0.5 * X + rnorm(N, mean = 0, sd = 1)
model_ols <- lm(Y ~ D + X)
```

FWL in R

```r
set.seed(123)
N <- 1000
X <- rnorm(N, mean = 0, sd = 1)
# Generate binary treatment D, making D and X correlated
D <- rbinom(N, size = 1, prob = plogis(X))
Y <- 2 * D + 0.5 * X + rnorm(N, mean = 0, sd = 1)
model_ols <- lm(Y ~ D + X)
```

```r
coeftable(model_ols)[, 1:2] %>% kable()
```

	Estimate	Std. Error
(Intercept)	-0.0292034	0.0450014
D	2.0576326	0.0682031
X	0.4307956	0.0343768

FWL in R

```r
resid_Y <- residuals(lm(Y ~ X))
resid_D <- residuals(lm(D ~ X))
model_fwl <- lm(resid_Y ~ resid_D)
```

FWL in R

```r
resid_Y <- residuals(lm(Y ~ X))
resid_D <- residuals(lm(D ~ X))
model_fwl <- lm(resid_Y ~ resid_D)
```

	Estimate	Std. Error
(Intercept)	0.000000	0.0310951
resid_D	2.057633	0.0681689

Specification Error

- Linear specification for controls is a substantive assumption

Specification Error

- Linear specification for controls is a substantive assumption
- Introduce additional bias to linear regression estimation

Specification Error

- Linear specification for controls is a substantive assumption
- Introduce additional bias to linear regression estimation
- Simulation:

Specification Error

- Linear specification for controls is a substantive assumption
- Introduce additional bias to linear regression estimation
- Simulation:
  - True effect is 0.2

Specification Error

- Linear specification for controls is a substantive assumption
- Introduce additional bias to linear regression estimation
- Simulation:
  - True effect is 0.2
  - Nonlinearity in both $y \sim x$ and $d \sim x$

Specification Error

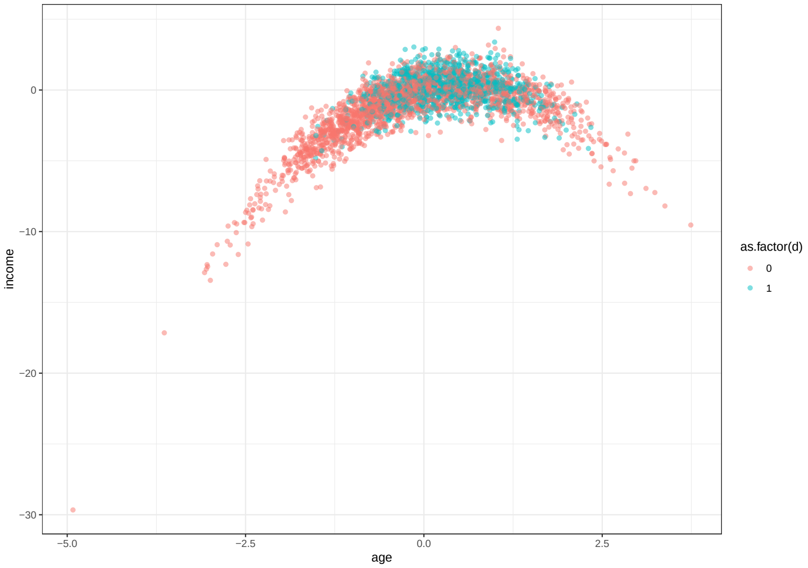
- Linear specification for controls is a substantive assumption
- Introduce additional bias to linear regression estimation
- Simulation:
  - True effect is 0.2
  - Nonlinearity in both $y \sim x$ and $d \sim x$

Specification Error

- Linear specification for controls is a substantive assumption
- Introduce additional bias to linear regression estimation
- Simulation:
  - True effect is 0.2
  - Nonlinearity in both y ~ x and d ~ x

```
set.seed(6)
N <- 3000
effect <- 0.2
age <- rnorm(N, 0, 1)
age2 <- -(age)^2 + age
d <- rbinom(N, size = 1, prob = plogis(age2))
income <- -(age)^2 + age + rnorm(N) + d * effect
```

Specification Error



Specification Error

- Using a linear specification for control leads to bias in estimate

```
etable(feols(income ~ age + d, data = dat), tex = T, fitstat = c
```

Dependent Variable:	income
Model:	(1)
Variables	
Constant	-1.294***
	(0.0403)
age	0.9361***
	(0.0328)
d	1.038***
	(0.0687)
Fit statistics	
Observations	3,000

IID standard-errors in parentheses
*Signif. Codes: ***: 0.01, **: 0.05, *: 0.1*

Specification Error

- But, a problem quickly arises.

Specification Error

- But, a problem quickly arises.
- How do we know what specification to use?

Specification Error

- But, a problem quickly arises.
- How do we know what specification to use?
- "Classical" Machine Learning: flexible algorithms to estimate nonlinear relationships

Specification Error

- But, a problem quickly arises.
- How do we know what specification to use?
- "Classical" Machine Learning: flexible algorithms to estimate nonlinear relationships
  - XGBoost, Random Forest, Lasso/ElasticNet… (Quant 3 for more on this)

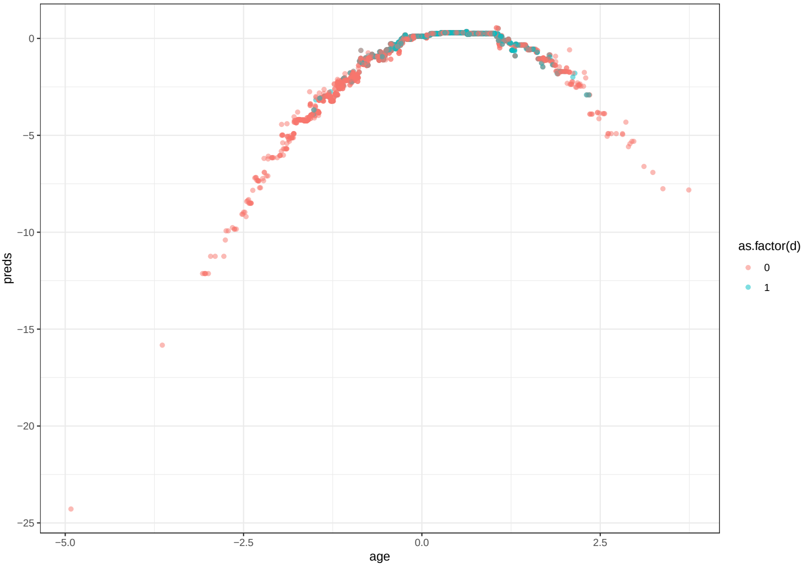
Classical Machine Learning

```
pacman::p_load(mlr3, xgboost, mlr3learners)

# Use XGBoost algorithm
learner <- lrn("regr.xgboost", eta = 0.1, nrounds = 35)
# Set up a task to predict 'income'
task <- as_task_regr(
    select(dat, income, age),
    target = "income"
)
# Fit to the data
learner$train(task)
```

Classical Machine Learning

```
dat$preds <- learner$predict_newdata(dat)$response
```



Double Machine Learning

- Ok, so we can predict Y given X in a nonlinear way.

Double Machine Learning

- Ok, so we can predict Y given X in a nonlinear way.
- How do we use this to retrieve a good estimate of `Y ~ D | X`?

Double Machine Learning

- Ok, so we can predict Y given X in a nonlinear way.
- How do we use this to retrieve a good estimate of `Y ~ D | X`?
- Basic idea is to use ML to model both `y ~ x` and `d ~ x` and use the residuals to retrieve a consistent estimate for θ

Double Machine Learning

- Ok, so we can predict Y given X in a nonlinear way.
- How do we use this to retrieve a good estimate of `Y ~ D | X`?
- Basic idea is to use ML to model both `y ~ x` and `d ~ x` and use the residuals to retrieve a consistent estimate for θ
- Sounds familiar? (FWL)

DML By Hand

```r
# Model y as a function of age
y.x <- lrn("regr.xgboost", eta = 0.1, nrounds = 35)
y.x.task <- as_task_regr(
    select(dat, income, age),
    target = "income"
)
y.x$train(y.x.task)
```

DML By Hand

```r
# Model y as a function of age
y.x <- lrn("regr.xgboost", eta = 0.1, nrounds = 35)
y.x.task <- as_task_regr(
    select(dat, income, age),
    target = "income"
)
y.x$train(y.x.task)
```

```r
# Model D as a function of age
d.x <- lrn("regr.xgboost", eta = 0.1, nrounds = 35)
d.x.task <- as_task_regr(
    select(dat, d, age),
    target = "d"
)
d.x$train(d.x.task)
```

DML By Hand

```r
# Model y as a function of age
y.x <- lrn("regr.xgboost", eta = 0.1, nrounds = 35)
y.x.task <- as_task_regr(
    select(dat, income, age),
    target = "income"
)
y.x$train(y.x.task)
```

```r
# Model D as a function of age
d.x <- lrn("regr.xgboost", eta = 0.1, nrounds = 35)
d.x.task <- as_task_regr(
    select(dat, d, age),
    target = "d"
)
d.x$train(d.x.task)
```

```r
# Calculate residuals
d.x.resid <- dat$d - d.x$predict_newdata(dat)$response
y.x.resid <- dat$income - y.x$predict_newdata(dat)$response
```

DML By Hand

```r
lm(y.x.resid ~ d.x.resid)
```

```
##
## Call:
## lm(formula = y.x.resid ~ d.x.resid)
##
## Coefficients:
## (Intercept)    d.x.resid
##    -0.03956      0.24633
```