

# DML Applications

---

Fanisi Mbozi, Gabrielle Péloquin-Skulski and Clemente Sánchez

Recent Developments in Political Methodology | September 29th, 2025

# Advantages and Limitations of DML

- DML has the **advantage** of relaxing parametric assumptions.
  - Don't need to assume constant treatment effect
  - Don't need to assume linearity

# Advantages and Limitations of DML

- DML has the **advantage** of relaxing parametric assumptions.
  - Don't need to assume constant treatment effect
  - Don't need to assume linearity
- DML has the **disadvantage** of assumptions on the **outcome model** and the **propensity score model**.

# Advantages and Limitations of DML

- DML has the **advantage** of relaxing parametric assumptions.
  - Don't need to assume constant treatment effect
  - Don't need to assume linearity
- DML has the **disadvantage** of assumptions on the **outcome model** and the **propensity score model**.
- **Experimental design**: treatment is random, so the propensity score is known and outcome model can *afford* to be misspecified.

# Advantages and Limitations of DML

- DML has the **advantage** of relaxing parametric assumptions.
  - Don't need to assume constant treatment effect
  - Don't need to assume linearity
- DML has the **disadvantage** of assumptions on the **outcome model** and the **propensity score model**.
- **Experimental design**: treatment is random, so the propensity score is known and outcome model can *afford* to be misspecified.
- **Observational design**: both the outcome and propensity score must be correctly specified, because rate of convergence is slow.

# Causal Model Choice

## Partially Linear Regression

- Continuous treatment
- No heterogeneous treatment effects
- No functional form assumption for controls
- Command: DoubleMLPLR

## Interactive Regression Model

- Dichotomous treatment
- Allows heterogeneous treatment effects
- No functional form assumption for controls
- Command: DoubleMLIRM

# Causal Model Choice

## Partially Linear Regression

- Continuous treatment
- No heterogeneous treatment effects
- No functional form assumption for controls
- Command: DoubleMLPLR

## Interactive Regression Model

- Dichotomous treatment
- Allows heterogeneous treatment effects
- No functional form assumption for controls
- Command: DoubleMLIRM

- **Clustered Data:** DML assumes i.i.d.; assign whole clusters to folds and use cluster-robust SEs (Command: DoubleMLClusterData).

# Experimental design

---



## Platas and Raffler (2021): Closing the Gap

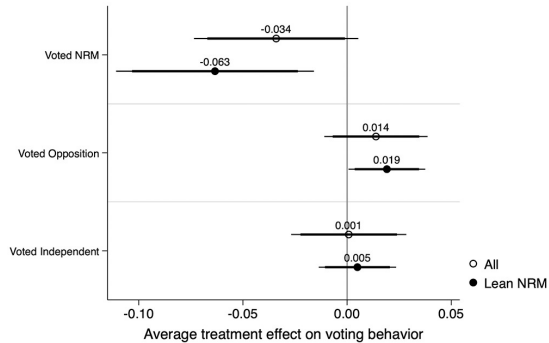
Experiment on the effects of **providing information in dominant party regimes**.

- Information asymmetry in dominant party regimes (in favor of the regime).
- Can provision of information about all parties (especially opposition ones), **reduce the asymmetry** in terms of voter knowledge, opposition likability or vote intentions.
- 11 constituencies in Uganda. Within each constituency randomly assign some villages to a screening of parliamentary candidate debates.

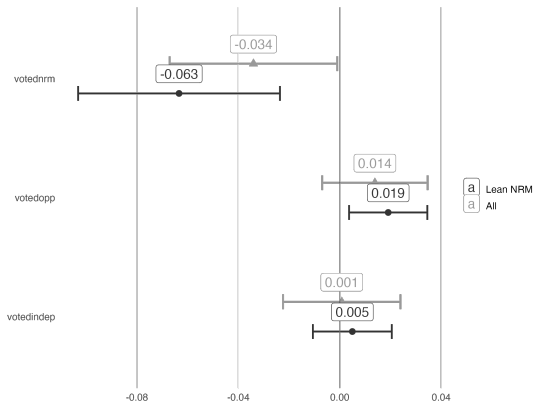
# Replicating original results

Main relationship:

- $\text{Vote intention} \sim \text{Received Informational Treatment (0/1)}$
- $\text{Vote (Incumbent/Opposition/Indep.)} \sim \text{Treatment} + \text{Treatment} \times \text{Covariates}$ .

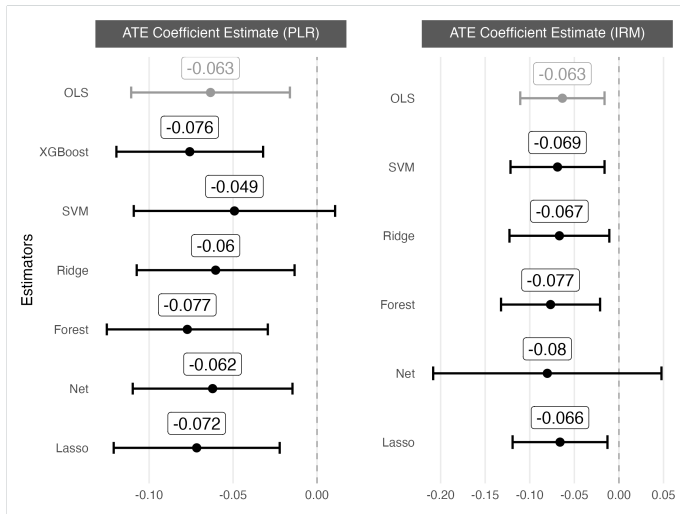


(a) Original



(b) Replication

# DML application for experiment



- The result we replicated have a **binary treatment** and **outcome**. The **classification** versions of the machine learning models will likely perform better here.

## Robustness and Application Notes

- The result we replicated have a **binary treatment** and **outcome**. The **classification** versions of the machine learning models will likely perform better here.
- For the DoubleMLIRM function, treatment is assumed to be **binary**, so the learner used to model the propensity score HAS to be a **classification learner**, not a regression one (e.g. use *classif.cv\_glmnet* instead of *regr.cv\_glmnet*).

## Robustness and Application Notes

- The result we replicated have a **binary treatment** and **outcome**. The **classification** versions of the machine learning models will likely perform better here.
- For the DoubleMLIRM function, treatment is assumed to be **binary**, so the learner used to model the propensity score HAS to be a **classification learner**, not a regression one (e.g. use `classif.cv_glmnet` instead of `regr.cv_glmnet`).
- Just make a mental note that such differences in data type matter for exploring pairs of learners when using IRM vs. PLR.

# Observational design

---

Motivating the Machine: Which Brokers do Parties Pay?



Motivating the Machine: Which Brokers do Parties Pay?

- How do elites **compensate brokers** for their help during elections?

### Motivating the Machine: Which Brokers do Parties Pay?

- How do elites **compensate brokers** for their help during elections?
- While some might act due to ideological commitments, the literature suggests that most of them require **material incentives**.

### Motivating the Machine: Which Brokers do Parties Pay?

- How do elites **compensate brokers** for their help during elections?
- While some might act due to ideological commitments, the literature suggests that most of them require **material incentives**.
- Little evidence of compensation by the literature, and what little there is focuses exclusively in the campaign season.

### Motivating the Machine: Which Brokers do Parties Pay?

- How do elites **compensate brokers** for their help during elections?
- While some might act due to ideological commitments, the literature suggests that most of them require **material incentives**.
- Little evidence of compensation by the literature, and what little there is focuses exclusively in the campaign season.
- *Immediately after the election elites reward brokers that have **delivered votes**, but later on they reward brokers with **connections**.*

## Motivating the Machine: Which Brokers do Parties Pay?

- How do elites **compensate brokers** for their help during elections?
- While some might act due to ideological commitments, the literature suggests that most of them require **material incentives**.
- Little evidence of compensation by the literature, and what little there is focuses exclusively in the campaign season.
- *Immediately after the election elites reward brokers that have **delivered votes**, but later on they reward brokers with **connections**.*

## Design

- Survey of 1,000 party brokers in Ghana.

## Motivating the Machine: Which Brokers do Parties Pay?

- How do elites **compensate brokers** for their help during elections?
- While some might act due to ideological commitments, the literature suggests that most of them require **material incentives**.
- Little evidence of compensation by the literature, and what little there is focuses exclusively in the campaign season.
- *Immediately after the election elites reward brokers that have **delivered votes**, but later on they reward brokers with **connections**.*

## Design

- Survey of 1,000 party brokers in Ghana.
- Condition on observables.

# Replicating original results

Main relationship:

- T5: Payment immediately after election  $\sim$  **Vote swing**
- T6: Payment after two years  $\sim$  **Number of connections**

# Replicating original results

Main relationship:

- T5: Payment immediately after election  $\sim$  **Vote swing**
- T6: Payment after two years  $\sim$  **Number of connections**

Additional:

- Matrix of covariates
- Constituency fixed effects
- SEs **clustered** at the polling station



# Replicating original results

## Main relationship:

- T5: Payment immediately after election  $\sim$  **Vote swing**
- T6: Payment after two years  $\sim$  **Number of connections**

## Additional:

- Matrix of covariates
- Constituency fixed effects
- SEs **clustered** at the polling station

Table 5. Major Patronage Payments Immediately after the Election

	(1)
NPP pres. vote swing at polling station 2012–16	.702** (.321)
NPP pres. vote swing at polling station 2012–16 (raw votes)	
Campaign activity in 2016 (.9)	.007 (.005)

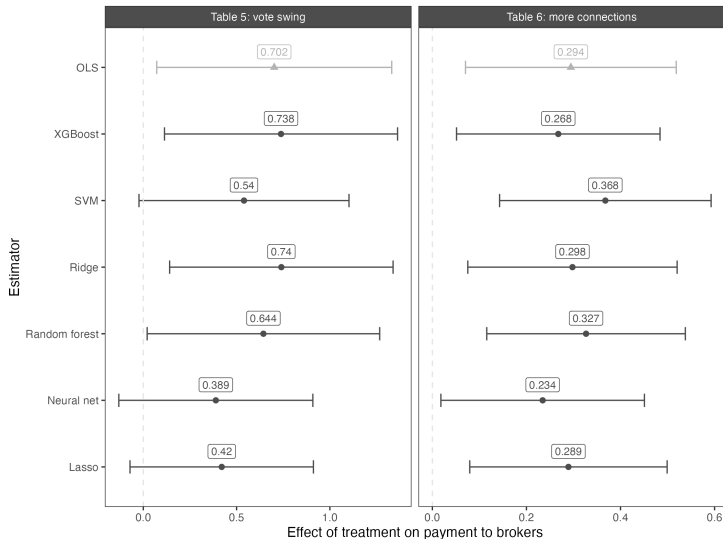
Table 6. Predictors of Major Patronage in the Nonelectoral F

	(1)
Connections Up (wave 1)	.294** (.114)
Connections Up, politicians (wave 1)	
Connections Up, bureaucrats (wave 1)	
Connections Up, constituency executives (wave 1)	

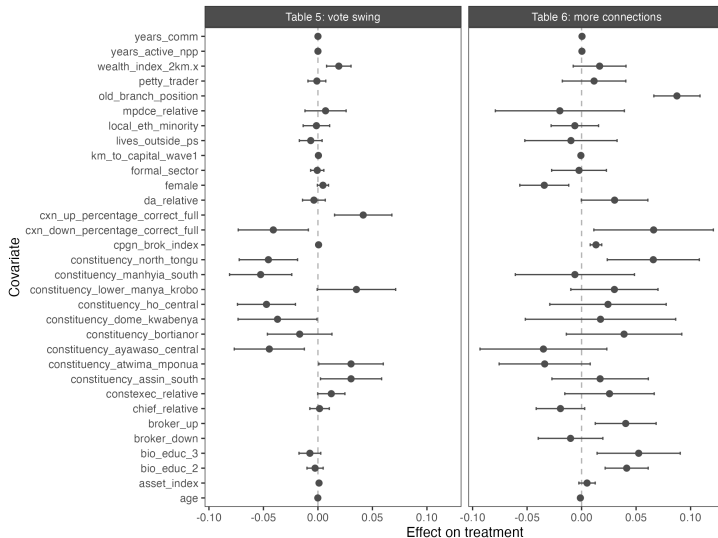
```
> etable(model_replication_t5, model_replication_t6)
Dependent Var.:      model_replication_t5      model_replication_t6
                    big_pat_inmed      big_pat_after2
Constant              -0.1439 (0.0907)      0.4557** (0.1059)
npp12to16_ps_swing.NEW 0.7017* (0.3213)      -0.0481 (0.3241)
cpgn_brok_index         0.0071 (0.0054)      0.0110 (0.0081)
cxn_up_percentage_correct_full 0.1042 (0.0953)      0.2944* (0.1143)
cxn_down_percentage_correct_full 0.1680* (0.0779)      -0.1010 (0.0869)
age                    0.0011 (0.0012)      -0.0038* (0.0017)
female                 0.0494 (0.0311)      0.2702** (0.0417)
chief_relative         0.0598 (0.0324)      0.0244 (0.0344)
constexec_relative     0.1299 (0.0709)      0.0691 (0.0580)
da_relative            -0.0593 (0.0334)      0.0033 (0.0476)
mpdce_relative         -0.0508 (0.0464)      -0.0918 (0.0739)
local_eth_minority     0.0036 (0.0276)      -0.0393 (0.0347)
lives_outside_ps      -0.0131 (0.0415)      0.0164 (0.0620)
petty_trader          0.0093 (0.0377)      0.0439 (0.0541)
formal_sector         0.0212 (0.0284)      -0.0686 (0.0394)
asset_index           0.0162 (0.0082)      -0.0200 (0.0109)
years_active_npp      0.0010 (0.0021)      -0.0029 (0.0020)
years_comm            0.0010 (0.0008)      0.0022 (0.0013)
kn_to_capital_wave1   3.65e-5 (0.0023)      -0.0010 (0.0023)
wealth_index_2km.x    -0.0347 (0.0342)      -0.0171 (0.0337)
bio_educ_2            -0.0004 (0.0227)      0.0039 (0.0348)
bio_educ_3            0.0379 (0.0491)      0.0306 (0.0617)
```

# DML application for observational study

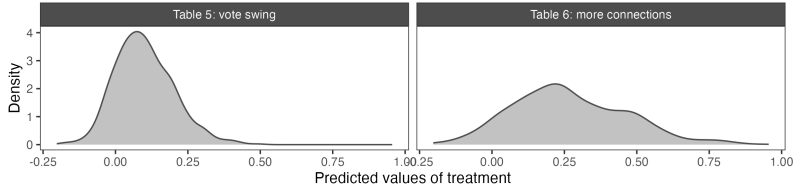
Five-fold cross-fitting and 20 repetitions



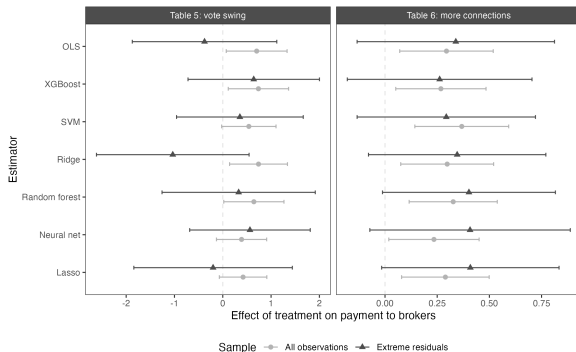
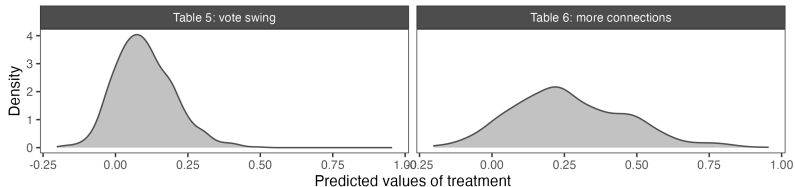
# Balance of covariates



# General propensity approach



# General propensity approach



# Conclusions

---

## Takeaways and Practical Advice

- **Learner choice matters** — compare multiple learners.

## Takeaways and Practical Advice

- **Learner choice matters** — compare multiple learners.
- **Tuning parameters matter** — more repetitions = more stable estimate; more folds = less overfitting.



## Takeaways and Practical Advice

- **Learner choice matters** — compare multiple learners.
- **Tuning parameters matter** — more repetitions = more stable estimate; more folds = less overfitting.
- **Divergence across learners** indicates sensitivity to functional form.

# Takeaways and Practical Advice

- **Learner choice matters** — compare multiple learners.
- **Tuning parameters matter** — more repetitions = more stable estimate; more folds = less overfitting.
- **Divergence across learners** indicates sensitivity to functional form.
- Differences with **OLS** suggest (1) **treatment effect heterogeneity**, (2) **nonlinearity**, or (3) **limited overlap**.

## Takeaways and Practical Advice

- **Learner choice matters** — compare multiple learners.
- **Tuning parameters matter** — more repetitions = more stable estimate; more folds = less overfitting.
- **Divergence across learners** indicates sensitivity to functional form.
- Differences with **OLS** suggest (1) **treatment effect heterogeneity**, (2) **nonlinearity**, or (3) **limited overlap**.
- **Random assignment** ensures the propensity score model is correct, eliminating possibility of **bias**.

# Takeaways and Practical Advice

- **Learner choice matters** — compare multiple learners.
- **Tuning parameters matter** — more repetitions = more stable estimate; more folds = less overfitting.
- **Divergence across learners** indicates sensitivity to functional form.
- Differences with **OLS** suggest (1) **treatment effect heterogeneity**, (2) **nonlinearity**, or (3) **limited overlap**.
- **Random assignment** ensures the propensity score model is correct, eliminating possibility of **bias**.
- **Binary vs. continuous** treatments rely on different estimators (**AIPW** vs. **PLR**).

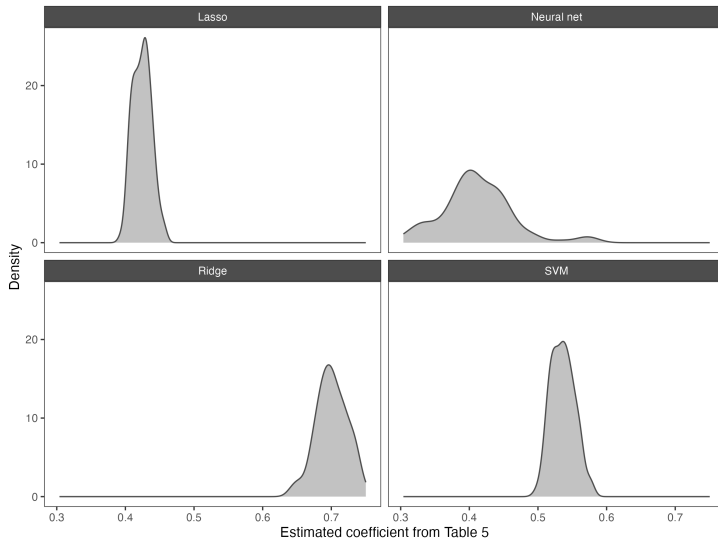
# Takeaways and Practical Advice

- **Learner choice matters** — compare multiple learners.
- **Tuning parameters matter** — more repetitions = more stable estimate; more folds = less overfitting.
- **Divergence across learners** indicates sensitivity to functional form.
- Differences with **OLS** suggest (1) **treatment effect heterogeneity**, (2) **nonlinearity**, or (3) **limited overlap**.
- **Random assignment** ensures the propensity score model is correct, eliminating possibility of **bias**.
- **Binary vs. continuous** treatments rely on different estimators (**AIPW** vs. **PLR**).
- Standard DML assumes **i.i.d.**; clustering requires adjustments.

# Appendix

---

# Repeating estimation 100 times (Brierley and Nathan, 2022)



# DML with continuous treatment and i.i.d. SEs

R Code

```
data_dml <- data |>
  # turn into data.table
  as.data.table() |>
  # create the DML data object
  DoubleMLData$new(y_col = outcome,
                   d_cols = treatment,
                   x_cols = c(covariates, dummies))
# define learner as random forest
learner <- lrn("regr.ranger")
# obtain two clones of the learner
ml_l_sim <- learner$clone()
ml_m_sim <- learner$clone()
dml_object <- data_dml |>
  # specify the learners
  DoubleMLPLR$new(ml_l = ml_l_sim, ml_m = ml_m_sim,
                 # use 5-fold cross-fitting and 20 rounds
                 n_folds=5, n_rep=20)
# fit the model
dml_object$fit()
# extract the coefficients and standard errors
coefficients <- tibble(estimate = dml_object$coef, se = dml_object$se)
```



# DML with continuous treatment and clustering

R Code

```
data_dml <- data |>
  # turn into data.table
  as.data.table() |>
  # create the DML data object (note the change in function)
  DoubleMLClusterData$new(y_col = outcome,
                           d_cols = treatment,
                           x_cols = c(covariates, dummies),
                           # key change
                           cluster_cols = clusters)

# define learner as random forest
learner <- lrn("regr.ranger")
# obtain two clones of the learner
ml_l_sim <- learner$clone()
ml_m_sim <- learner$clone()
dml_object <- data_dml |>
  # specify the learners
  DoubleMLPLR$new(ml_l = ml_l_sim, ml_m = ml_m_sim,
                  # use 5-fold cross-fitting and 20 rounds
                  n_folds=5, n_rep=20)

# fit the model
dml_object$fit()
# extract the coefficients and standard errors
coefficients <- tibble(estimate = dml_object$coef, se = dml_object$se)
```

# DML Interactive Regression Model and Clustering

R Code

```
data_dml <- data |>
  # turn into data.table
  as.data.table() |>
  # create the DML data object (note the change in function)
  DoubleMLClusterData$new(y_col = outcome,
                          d_cols = treatment,
                          x_cols = c(covariates, dummies),
                          # key change
                          cluster_cols = clusters)

# define learner as random forest
learner <- lrn("classif.ranger")
# obtain two clones of the learner
ml_g = learner$clone() # outcome model
ml_m = learner$clone() # treatment model-must be classification learner
dml_object <- data_dml |>
  # specify the learners
  DoubleMLIRM$new(dml_cl_obj, ml_g, ml_m, n_folds=5)

# fit the model
dml_object$fit()
# extract the coefficients and standard errors
coefficients <- tibble(estimate = dml_object$coef, se = dml_object$se)
```