Introduction to DeclareDesign

Tara Slough

May 8, 2016

What is DeclareDesign?

- Framework, R package for formally characterizing research designs
  - Problem: Degree of detail about research design varies greatly in published work, pre-analysis plans etc.
  - Motivation: Create framework capable of characterizing all research designs (any/all methods)
- Tools for analysis of designs
  - Big question: What could we have learned from a design?
  - Ancillary benefits: Create "dummy" datasets with which you can practice different analyses

What could we have learned from a design?

- A research design at the front-end consists of:
  - Design choices by researcher/nature/dataset maker (depends on method)
  - Set of beliefs/assumptions about how the world works
- We can learn about how a design functions through simulation
  - Logic of Monte Carlo analysis
  - Draw data, analyze, save estimates, repeat many times
  - Diagnoses based on estimates

Six Components of a Research Design

1. **Population**: Set of units about which inferences are sought and their characteristics
2. **Potential outcomes**: Outcomes each unit might exhibit depending on how causal process changes the world
3. **Sampling strategy**: Strategy used to select units to include in study
4. **Assignment**: Manner in which units are assigned to reveal one potential outcome or another
5. **Estimand**: Quantities that we want to learn about in the world, in terms of potential outcomes
6. **Estimator**: Procedure for generating estimates of the quantities we want to learn about

Population

- Start from theory:
  - Where should the theory apply?
  - Where shouldn't it apply? (scope conditions)
- What is the population that we want to make an inference about?
- Often practical limitations about the population that we can study
  - Regardless of method employed
  - But stay idealistic for the moment

Sampling Strategy

- How are we selecting units to analyze?
  - How do we choose the case (context)?
  - What does this selection method mean about inference to the population above?
- Types of samples:
  - Population (“big data,” very specific populations)
  - Convenience (lab experiments, some surveys etc.)
  - Some sort of random sample (some surveys etc.)
- If we can’t make a population inference from a sample, do we:
  - Redefine population?
  - Only worry about sample diagnostics?
- One source of concerns about external validity
  - Unclear (to me) that this is an “experiments” issue. . .

Potential Outcomes

- Codifies our asssumptions about the relationships between different treatment conditions, baseline covariates, and outcomes
  - Should be rooted in theory
- Functional form of relationships must be specified
  - Should be informed by theory, but often theories are not specific on this point
  - Less difficult in case of binary treatments
  - Another source of concerns about external validity (model-based inference)
- Some difficulties of experimental analysis should be viewed as potential outcomes!
  - (Non)-compliance
  - Spillovers
  - Attrition

Assignment

How is treatment/independent variable of interest assigned?

- Experiments:
  - Input the randomizations.
  - Defaults allow for simple, complete, blocked, clustered, and blocked and clustered randomization, among others
- Quasi-experiments/natural experiments:
  - Treatment assignment requires more assumptions about the assignment process
- Other observational work:
  - Stronger assumptions about the assignment of treatment, assignment could be modeled on covariates

Estimand, Estimator

- What do we want to know?
  - We've talked about the ATE, various marginal effects, conditional marginal effects
  - Other effects of interest: ITT, LATE, CATE
  - Estimands not specified frequently (enough) in existing literature
  - Snarky comment: Stars don't mean much if we don't know what the coefficient is measuring
- Estimator:
  - Too often we utilize estimators without identifying estimand
  - Many estimators consistent with each estimand – though some work better than others. . .

The Guts: Declare Design

```
population <- declare_population()
sampling <- declare_sampling()
assignment <- declare_assignment()
potential_outcomes <- declare_potential_outcomes()
estimand <- declare_estimand()
estimator <- declare_estimator(estimand = estimand)

my_design <- declare_design(
          population         = population,
          sampling           = sampling,
          potential_outcomes = potential_outcomes,
          assignment         = assignment,
          estimator          = estimator)
```

Audience for `DeclareDesign`

Three-ish audiences:

1. **Ninjas:** Advanced R users that (might) specify user-input functions for any design
2. **Advanced:** Use built-in functionalities to characterize a wide range of designs
3. **Novices:** Use template functions (some here, many forthcoming) to characterize and examine a variety of designs with a few simple arguments.

Using DeclareDesign

```
install.packages("devtools") # run once only!
library(devtools)
install_github("DeclareDesign/DeclareDesign") # run once o
library(devtools)


source("templates file.R")
```

Two templates in this file:

- Generalized *m*-arm
- 2 × 2 factorial

See .pdf for detail about all arguments to these functions.

How Can we Use DeclareDesign to Learn Designs?

- Suppose you want to extend Chong, De la O, Karlan, Wantchekon (2014) to a different context.
- Three treatment arms
  1. Pure control (no flyer)
  2. Placebo (flyer about the federal transfer)
  3. Treatment (flyer about federal transfer with results of corruption audit)
- DV: Turnout
- 600 municipalities are candidates for evaluation
- You can only afford to implement treatment and do data collection in 450 municipalities
- Hypothesized treatment effect comes from Chong et al. (2014) findings

Design 1: No Pretreatment Covariate

- Assume the following Potential Outcomes Function:

$$\text{Turnout}_i = 60 - 1.5 \times \text{Treatment}_i + 0.5 \times \text{Placebo}_i + \epsilon_i$$

- We can enter this entire design with the following code:

```
three_arm_des_1 <- m_arm_template(
      N           = 600,           # 600 munis in pop.
      n           = 450,           # 450 in sample
      m           = 3,             # 3 arms, 150/arm
      mu_Y0       = 60,            # 60% turnout in ctrl
      ATEs        = c(-1.5, 0.5), # Treatment effects
      noise_scale = 8)             # SD of error term
```

- Estimands are ATEs, Estimator is OLS

Design 2: Lagged Turnout as Pretreatment Covariate

Assume the following Potential Outcomes Function:

$$\text{Turnout}_i = 28.5 - 1.5 \times \text{Treatment}_i + 0.5 \times \text{Placebo}_i + 0.5 \times \text{Turnout}_{t-1} + \epsilon_i$$

Declare design without covariate adjustment:

```
three_arm_des_2 <- m_arm_template(
 N                  = 600,          # 600 in population
 n                  = 450,          # 450 in sample
 m                  = 3,            # 3 arms, 150/arm
 mu_Y0              = 28.5,         # basline in ctrl
 ATEs               = c(-1.5, 0.5), # Treatment effects
 noise_scale        = 4,            # SD of error term
 coef_X             = 0.5,          # Coef. on turnout, t-1
 location_scale_X = c(65, 8),       # Mean, SD of turnout, t
 cov_adjustment     = FALSE)        # No covariate adjustmen
```

Design 3: Lagged Turnout as Pretreatment Covariate

Assume the following Potential Outcomes Function:

$$\text{Turnout}_i = 28.5 - 1.5 \times \text{Treatment}_i + 0.5 \times \text{Placebo}_i + 0.5 \times \text{Turnout}_{t-1} + \epsilon_i$$

Declare design **with** covariate adjustment:

```
three_arm_des_3 <- m_arm_template(
 N                 = 600,            # 600 in population
 n                 = 450,            # 450 in sample
 m                 = 3,              # 3 arms, 150/arm
 mu_Y0             = 28.5,           # basline in ctrl
 ATEs              = c(-1.5, 0.5), # Treatment effects
 noise_scale       = 4,              # SD of error term
 coef_X            = 0.5,            # Coef. on turnout, t-1
 location_scale_X = c(65, 8),       # Mean, SD of turnout, t
 cov_adjustment    = TRUE)          # No covariate adjustment
```

On what basis should we be comparing designs?

Assume that we run the analysis N times, collecting estimates $\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_N$

Most important/standard:

- Bias: $\mathbb{E}[\hat{\theta} - \theta]$
- RMSE: $\sqrt{\mathbb{E}[(\hat{\theta} - \theta)^2]}$
- Coverage: Proportion of simulations in which this estimated 95% confidence intervals contains true estimand (θ)
- Power: Proportion of simulations in which $\hat{\theta}$ is significant at the $\alpha = 0.05$ level

Others:

- Type-S error rate: Proportion of simulations in which the sign of $\hat{\theta}$ is different from the true estimand θ
- Exaggeration ratio: $\mathbb{E}[\hat{\theta}]/\theta$

Diagnosis of designs:

- Ancillary benefit:

```
data <- draw_data(three_arm_des_1) # Not diagnosis
```

- Diagnosis of designs

```
d1 <- diagnose_design(three_arm_des_1, population_draws = 5
d2 <- diagnose_design(three_arm_des_2, population_draws = 5
d3 <- diagnose_design(three_arm_des_3, population_draws = 5
```

Comparison of 3 designs

- Compare just the ATE of the treatment (corruption information) versus pure control

```r
knitr::kable(df, digits = 3)
```

	Design_1	Design_2	Design_3
Mean, SATE	-1.500	-1.500	-1.500
S.D., SATE	0.000	0.000	0.000
Mean Sample RMSE	0.928	0.651	0.455
Mean Sample Bias	0.087	0.012	0.028
Mean Sample Coverage	0.916	0.948	0.936
Mean Sample Type S Rate	0.068	0.008	0.000
Mean Sample Exaggeration Ratio	0.942	0.992	0.981
Mean, Power	0.308	0.624	0.884
S.D., Power	0.190	0.234	0.122
Mean, Estimate	-1.413	-1.488	-1.472
S.D., Estimate	0.985	0.687	0.473

Take Aways

1. Framework to think about complete research designs
   - Move toward qustion "what could I learn given my design"
   - Way to conceptualize differences between different approaches
2. Tools for examining research designs *ex-ante*
   - Creating mock datasets
   - Diagnosing designs
3. Further steps (that I know of)
   - Development of templates for different research designs (suggestions welcome)
   - Some ammendments to current package to improve functionality/eliminate some issues
   - Programming work to increase speed of simulations, potentially expand to Stata