DeclareDesign: : CHEAT SHEET

Model

What is your model of the world, including how outcomes respond to interventions in the world?

Population

Define the size of the population, hierarchical structure (if any), and background variables.

Simple dataset with no background variables

```
pop <- declare_population(N = 100)
pop()
```

Simple dataset with background variables

```
declare_population(N = 100,
                   X = rnorm(N))
```

Two-level dataset

```
declare_population(
  schools =
    add_level(N = 10,
              funding = rnorm(N)),
  students =
    add_level(N = 100,
              scores = rnorm(N))
)
```

Outcomes

Outcomes that depend on a treatment (Z)

Using a formula

```
declare_potential_outcomes(
  Y ~ .5 * Z + rnorm(N))
```

As separate variables

```
declare_potential_outcomes(
  Y_Z_0 = rnorm(N),
  Y_Z_1 = Y_Z_0 + .5)
```

Outcomes that do not depend on treatment

```
declare_potential_outcomes(
  Y = rnorm(N))
```

Inquiry

What is the research question you want to answer?

Causal inquiries

```
declare_estimand(
  ATE = mean(Y_Z_1 - Y_Z_0))
```

Descriptive inquiries

```
declare_estimand(
  Y_median = median(Y))
```

Conditional estimands

```
declare_estimand(
  LATE = mean(Y_Z_1 - Y_Z_0),
  subset = complier == TRUE)
```

Data Strategy

How will you generate data to answer your inquiry?

Sampling

```
declare_sampling(n = 100)

declare_sampling(
  strata_n = 20,
  strata = urban_area)
```

Treatment assignment

```
declare_assignment(m = 100)

declare_assignment(
  clusters = villages,
  m = 10)
```

Answer Strategy

How will you generate an answer to your inquiry?

OLS with robust standard errors

```
declare_estimator(
  Y ~ Z, model = lm_robust)
```

2SLS instrumental variables regression with robust SEs

```
declare_estimator(
  Y ~ D | Z, model = iv_robust)
```

Difference-in-means

```
declare_estimator(
  Y ~ Z,
  model = difference_in_means)
```

DeclareDesign is a software implementation of the MIDA framework, according to which research designs have a **M**odel of the world, an **I**nquiry about that model, a **D**ata strategy that generates information about the world, and an **A**nswer strategy that uses data to make a guess about the **I**nquiry. Declared designs can be "diagnosed" to calculate the properties of the design such as power and bias using Monte Carlo simulation.

All `declare_*` functions return *functions*. Most functions take a data.frame and return a data.frame.

Design Declaration

Put together all the steps into a declared design using the + operator

```
design <-
  declare_population(N = 200, X = rnorm(N)) +
  declare_potential_outcomes(Y ~ .5 * Z + X) +
  declare_estimand(ATE = mean(Y_Z_1 - Y_Z_0)) +
  declare_sampling(n = 100) +
  declare_assignment(m = 50) +
  declare_estimator(Y ~ Z, model = lm_robust)
```

```
draw_data(design)
draw_estimates(design)
get_estimates(design, data = real_data)
draw_estimands(design)
run_design(design)
summary(design)
compare_designs(design_1, design_2)
```

Design Diagnosis

Diagnose the properties of your design

```
diagnosis <- diagnose_design(
  design, sims = 100, bootstrap_sims = 100)
```

```
summary(diagnosis)
get_diagnosands(diagnosis)
get_simulations(diagnosis)
```

Custom diagnosands

```
diagnose_design(
  design,
  diagnosands = declare_diagnosands(
    sig_pos = mean(p.value < .05 & estimate > 0)))
```