Lecture 5

Alex Stephenson

9-3-2021

What did we do last time?

- Continued our discussion of SUTVA
- Discussed DAGs
- Began discussing what makes a good estimator

What are we doing today?

Announcing that PS1 is due on **9/17**

Announcing that Checkpoint 3 will be online by Sunday and due **9/10**

Finishing our explanation of how to evaluate estimators

Filling in the reasons why randomization is great

Estimators

An estimator is the procedure that generates a guess about the estimand

Example: Our estimand is the

$$ATE = E[Y_i(1)|D_i = 1] - E[Y_i(0)|D_i = 0]$$

With random sampling the plug-in estimator is the difference in means

$$\hat{E}_{DM} = \hat{E}[Y_i(1)] - \hat{E}[Y_i(0)$$

Estimators

There are three properties of an estimator that we care about when considering whether it is good.

Consistency

Unbiasedness

Precision

Estimators

There are three properties of an estimator that we care about when considering whether it is good.

Consistency: If we had enough data, the probability of our estimate would be far from the estimand goes to 0.

$$\lim_{h \to \infty} P(\hat{\theta}(D, Y) \in (\hat{\theta} - \epsilon, \hat{\theta} + \epsilon) = 1, \forall \epsilon > 0$$

Unbiasedness

Precision

Estimators

There are three properties of an estimator that we care about when considering whether it is good.

Consistency: If we had enough data, the probability of our estimate would be far from the estimand goes to 0.

Unbiasedness: The difference between the expected value of an estimator and the true value of the estimand in 0

$$E[\hat{\theta}] - \theta = 0$$

Precision

Estimators

There are three properties of an estimator that we care about when considering whether it is good.

Consistency: If we had enough data, the probability of our estimate would be far from the estimand goes to 0.

Unbiasedness: The difference between the expected value of an estimator and the true value of the estimand in 0

Precision An estimator is more precise than another estimator if it produces estimates closer to the estimand on average

Estimators

The most common metric for an estimator's precision is Mean Square Error

Formally for an estimator $\hat{\theta}$

$$MSE = E[(\hat{\theta} - \theta)^2] = V[\hat{\theta}] + (E[\hat{\theta}] - \theta)^2$$

An estimator with a lower MSE than another estimator is more efficient

Randomization

To this point we have been hand-wavy on how randomization eliminates selection bias.

We will see that randomization is more than simple chance, but also imminently related to the treatment assignment scheme.

When we say "randomized" we are therefore using a short-hand for how either.

- A mechanism that ensures $P[D_i = 1] = p, \forall i$
- A mechanism that ensures $P[D_i = 1 | X = x] = f(x), \forall i$

An English Definition of Random

A random procedure is a physical or electronic procedure by which randomization is conducted to ensure that assignment to the treatment group is statistically independent off all observed and unobserved variables

Formally

$$Y_i(0), Y_i(1) \perp\!\!\!\perp D_i, \forall i$$

or in the conditional independence assumption

$$Y_i(0), Y_i(1) \perp\!\!\!\perp D_i | \mathbf{X}_i, \forall i$$

Randomized Experimental Design

1. $P[D|X, Y(0), Y(1)]$ is controlled by the researcher and has a known functional form.

2. $P[D|X, Y(0), Y(1)]$ is probabilistic meaning that:
   $0 < p_i(X, Y(0), Y(1)) < 1$ for every i and for each $X, Y(0), Y(1)$

3. $P(D|X, Y(0), Y(1))$ is probabilistic by means of a randomization device whose physical features ensure that the assignment mechanism is unconfounded.

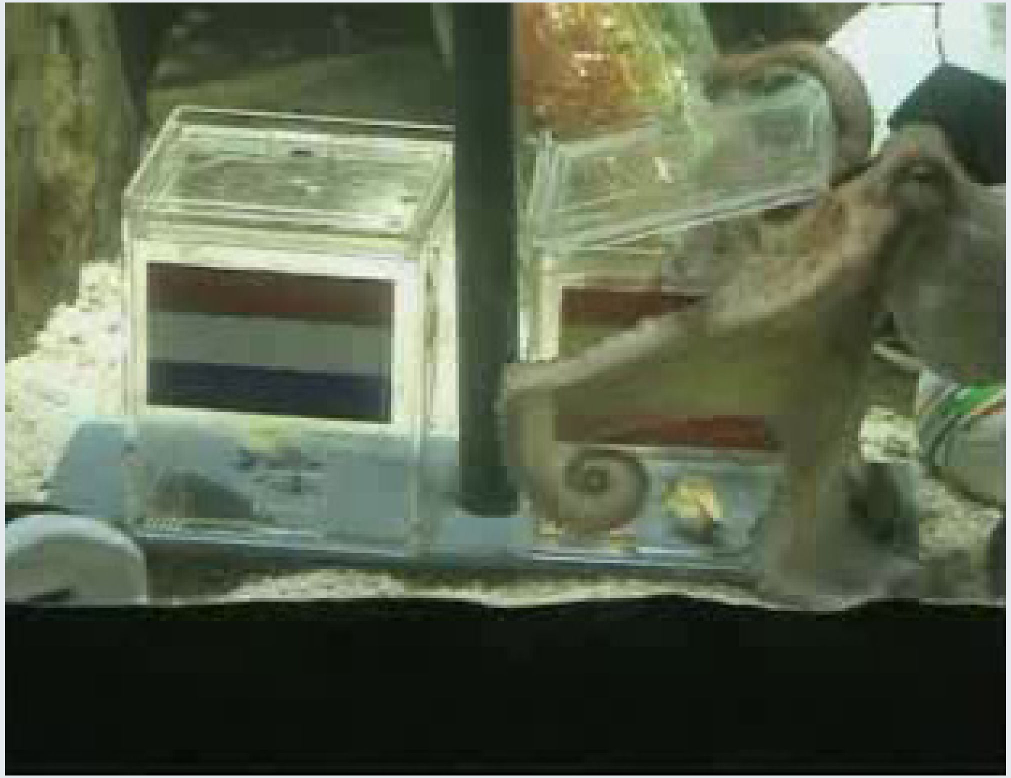
4. Unless stated otherwise we require that assignment is one of the class of "regular assignment mechanisms" (Imbens and Rubin 2015)

Two common types of randomization

Simple Random Assignment: a procedure that gives each unit an identical probability of being in treatment

Complete Random Assignment: a procedure that guarantees that exactly m of N total units are assigned to treatment with equal probability.

Randomization is more than chance



1. Arbitrary or inscrutable $\neq$ randomized
2. Assignment mechanism here is not known

Randomized Experimental Design

```r
D <- tibble(
  d1 = c(0,0,0),
  d2 = c(0,0,1),
  d3 = c(0,1,0),
  d4 = c(0,1,1),
  d5 = c(1,0,0),
  d6 = c(1,0,1),
  d7 = c(1,1,0),
  d8 = c(1,1,1)
)
```

```
## Rows: 3
## Columns: 8
## $ d1 <dbl> 0, 0, 0
## $ d2 <dbl> 0, 0, 1
## $ d3 <dbl> 0, 1, 0
## $ d4 <dbl> 0, 1, 1
## $ d5 <dbl> 1, 0, 0
## $ d6 <dbl> 1, 0, 1
## $ d7 <dbl> 1, 1, 0
## $ d8 <dbl> 1, 1, 1
```

Why Randomize?

The benefits of randomization is not in the uncertainty, but in the mechanism.

The mechanism is what ensures knowledge of the probability distribution of the assignment mechanism.

The power of a randomized assignment mechanism is that ensures that estimators we use in applied work are unbiased for the ATE no matter:

- what the treatment probabilities are
- how they were decided

Block Random Assignment

Definition: A procedure where units are partitioned into sub-groups (blocks) and complete random assignment occurs within each block.

Blocking reduces sampling variability and esures that certain subgroups are available for separate analysis

As a general rule of thumb, block on what you can randomize on what you can't

Block Random Assignment

```
set.seed(123)
vec <- c(rep(0,5),rep(1,5))
ta <- sample(vec, 10, replace = F)
tb <- sample(vec, 10, replace = F)
block_df <- tibble(id = 1:20,block = c(rep(0,10),rep(1,10)),
             treat = c(ta, tb))
```

Block Random Assignment

```
## # A tibble: 6 × 3
##       id block treat
##    <int> <dbl> <dbl>
## 1     15     1     0
## 2     13     1     0
## 3      3     0     0
## 4      1     0     0
## 5     11     1     1
## 6      5     0     1
```

Cluster Random Assignment

All units in the same cluster are placed as a group into either the treatment or control conditions.

Cluster assignment rules out possible allocations where individuals in the same cluster are assigned to different experimental conditions.

Analysis of these experiments will depend on the size and number of clusters.

Cluster Random Assignment

```
cluster <- tibble(
  school = c(rep("Berkeley",3), rep("Stanford",3)),
  student = c(1,2,3, 1,2,3),
  score = rnorm(n = 6, mean = 0, sd = 1),
  assign = c(rep(1,3), rep(0,3))
)
```

Cluster Random Assignment

```
## # A tibble: 6 × 4
##   school   student   score assign
##   <chr>      <dbl>   <dbl>  <dbl>
## 1 Berkeley       1  0.826       1
## 2 Berkeley       2 -0.0557      1
## 3 Berkeley       3 -0.784       1
## 4 Stanford       1 -0.734       0
## 5 Stanford       2 -0.216       0
## 6 Stanford       3 -0.335       0
```

Even though we have six students, we only have two groups. Each group has three students.

No Class on Monday

Read the Probability and Statistics Review from the Mixtape

We will pick back up on Wednesday with Randomization Inference