An Annotated R Script to Briefly Illustrate G-computation and Propensity Score-based Weighting

2018-10-04

Author:
Jim Rogers, Ph.D.
Metrum Research Group LLC
Tunxis Road, Suite 112
Tariffville, CT 06081

Contents

0.1 Credit

The following illustration is loosely based on remarks made in Pearl, Glymour and Jewell, Causal inference in statistics: a primer. Wiley, 2016. In that text they analyze an anonymized version of the data below (same numbers, but no mention of kidney stones or the original publication). They also don't use the term "G-computation", referring to the same concept as just "the adjustment formula".

0.2 Kidney Stone Treatment Data

We begin by "reverse engineering" the data in Table II of Charig et al., Comparison of treatment of renal calculi by open surgery, percutanesous nephrolithotomy, and extracorporeal shockwave lithotripsy. *BMJ* 1986;**292**:879–882. We restrict attention to the the rows of that table comparing open surgery to percutaneous nephrolithotomy.

```
## Enter tabulated BMJ data
n <- matrix(c(81, 234, 192, 55), nrow = 2)
N <- matrix(c(87, 270, 263, 80), nrow = 2)
surgery <- matrix(rep(c("open", "percutaneous"), 2), nr = 2)
stone_size <- matrix(rep(c("small", "large"), each = 2), nr = 2)

## Reverse engineer primitive BMJ data
dat <- data.frame(
                  Surgery = rep(surgery, N),
                  Size = rep(stone_size, N),
                  Success = do.call("c",
                                    lapply(seq(n),
                                         function(i) c(rep(0, N[i]-n[i]),
                                                       rep(1, n[i]))
                                    )
                  )
)
```

The following tables essentially just reproduce the information in the selected rows of the original tabulation:

```
dat %>% filter(Success == 1) %>% with(table(Surgery, Size))
```

```
##                Size
## Surgery         large small
##   open            192    81
##   percutaneous     55   234
```

```
dat %>% with(table(Surgery, Size))
```

```
##                Size
## Surgery         large small
##   open            263    87
##   percutaneous     80   270
```

0.3 Naive Analysis

If treatment had been randomized, the difference in the following point estimates would be a valid estimate of the causal effect of treatment:

```
naive_analysis <-
  dat %>%
  group_by(Surgery) %>%
  summarise(P_Success = sum(Success)/length(Success))
naive_analysis
```

```
## # A tibble: 2 x 2
##   Surgery      P_Success
##   <fct>            <dbl>
## 1 open             0.78
## 2 percutaneous     0.826
```

However, the observational nature of this data set has corrupted this comparison. We see the "paradox" if we split things out by size of the kidney stone:

```
 dat %>% filter(Size == "small") %>%
  group_by(Surgery) %>%
  summarise(P_Success = sum(Success)/length(Success))
```

```
## # A tibble: 2 x 2
##   Surgery      P_Success
##   <fct>            <dbl>
## 1 open             0.931
## 2 percutaneous     0.867
```

```
 dat %>% filter(Size == "large") %>%
  group_by(Surgery) %>%
  summarise(P_Success = sum(Success)/length(Success))
```

```
## # A tibble: 2 x 2
##   Surgery      P_Success
##   <fct>            <dbl>
## 1 open             0.730
## 2 percutaneous     0.688
```

To summarize: based on point estimates,

- Open surgery has better efficacy for subjects with small stones,
- Open surgery has better efficacy for subjects with large stones,
- Each subject falls into one of those two categories . . . and yet:
- Point estimates from the naive analysis imply that percutaneous surgery is better "overall".

So let's not do that naive "overall" analysis! Intuitively, the correct thing to do is take a weighted average of the conditional probabilities instead:

```
cond_probs <-
  bind_cols(
    dat %>% filter(Size == "large") %>%
      group_by(Surgery) %>%
      summarise(P_Success = sum(Success)/length(Success)) %>%
      select(-Surgery),
    dat %>% filter(Size == "small") %>%
      group_by(Surgery) %>%
      summarise(P_Success = sum(Success)/length(Success)) %>%
      select(-Surgery)
  ) %>%
  as.matrix()

cov_dist <-
  dat %>%
  group_by(Size) %>%
  summarise(P_Size = n()/nrow(dat)) %>%
  pull("P_Size")

simple_analysis <- cond_probs %*% cov_dist
simple_analysis <- data.frame(
  Surgery = levels(dat$Surgery),
  P_Success = simple_analysis
)
simple_analysis
```

```
##        Surgery P_Success
## 1         open 0.8325462
## 2 percutaneous 0.7788750
```

0.4 A modeling-and-simulation-based analysis

We can do essentially the same thing within a more routine "pharmacometrics" workflow, as follows:

1. Fit a model that takes into account the known explanatory factors affecting the response.
2. Create a virtual population with a covariate distribution reflecting the empirical covariate distribution.
3. Use the model to simulate the outcome for each subject in the virtual population if s/he receives open surgery. Compute the proportion of successes.
4. Use the model to simulate the outcome for each subject in the virtual population if s/he receives percutaneous surgery. Compute the proportion of successes.
5. Compare the proportions of success computed in the previous two steps.

The code version of that is:

```
model <- dat %>%
  group_by(Surgery, Size) %>%
  summarise(P_Success = sum(Success) / length(Success))

nsim <- 1e6

cov_dist <-
  dat %>%
  group_by(Size) %>%
  summarise(P_Size = n()/nrow(dat))

virtual_pop <-
  data.frame(
    SUBJID = 1:(nsim),
    Size = rep(cov_dist$Size,
               ## each covariate level represented the expected number
               ## of times in a population of nsim subjects:
               times = cov_dist$P_Size * nsim
               )
  )
```

```
pop_sim_open <-
  virtual_pop %>%  mutate(Surgery = "open") %>%
  left_join(model) %>%
  ## If we were clever we could skip the following step of adding residual
  ## variability, but let's not use our brains as a crutch :-)
  mutate(SimSuccess = rbinom(n(), 1, P_Success)) %>%
  group_by(Surgery) %>%
  summarise(P_Success = sum(SimSuccess)/length(SimSuccess))
pop_sim_percutaneous <-
  virtual_pop %>% mutate(Surgery = "percutaneous") %>%
  left_join(model) %>%
  mutate(SimSuccess = rbinom(n(), 1, P_Success)) %>%
  group_by(Surgery) %>%
  summarise(P_Success = sum(SimSuccess)/length(SimSuccess))
pmx_analysis <- bind_rows(pop_sim_open, pop_sim_percutaneous)
pmx_analysis
```

```
## # A tibble: 2 x 2
##   Surgery      P_Success
##   <chr>             <dbl>
## 1 open              0.833
## 2 percutaneous      0.779
```

That gives us the same answer as the previous "simple" analysis, plus or minus a bit of Monte-Carlo error from the simulation.

0.5 A propensity-score based analysis:

The `simple` analysis and the `pmx` analysis above are both flavors of what is called G-computation. That approach can be thought of as focusing on the outcome model. An ostensibly very different approach is to develop a treatment model (or a "propsensity model"), and make weighting adjustments based on that.

First note that we can think of our earlier "naive" analysis as the application of a weighted procedure where subject received a weight of 1:

```
dat %>%
  mutate(Weight = 1) %>%
  group_by(Surgery) %>%
  summarise(P_Success = sum(Success * Weight)/sum(Weight))
```

```
## # A tibble: 2 x 2
##   Surgery      P_Success
##   <fct>            <dbl>
## 1 open             0.78
## 2 percutaneous     0.826
```

The inverse probability weighting (IPW) approach instead uses weights related to propensity scores:

```
prop_scores <-
  dat %>%
  group_by(Size) %>%
  summarise(Prop_Score = sum(Surgery == "percutaneous") / length(Surgery))

ipw_analysis <-
  dat %>%
  left_join(prop_scores) %>%
  mutate(Weight = case_when(
    .$Surgery == "open" ~ (1 / (1-Prop_Score)),
    .$Surgery == "percutaneous" ~ (1 / Prop_Score)
  )) %>%
  group_by(Surgery) %>%
  summarise(P_Success = sum(Success * Weight)/sum(Weight))
```

0.6 Compare:

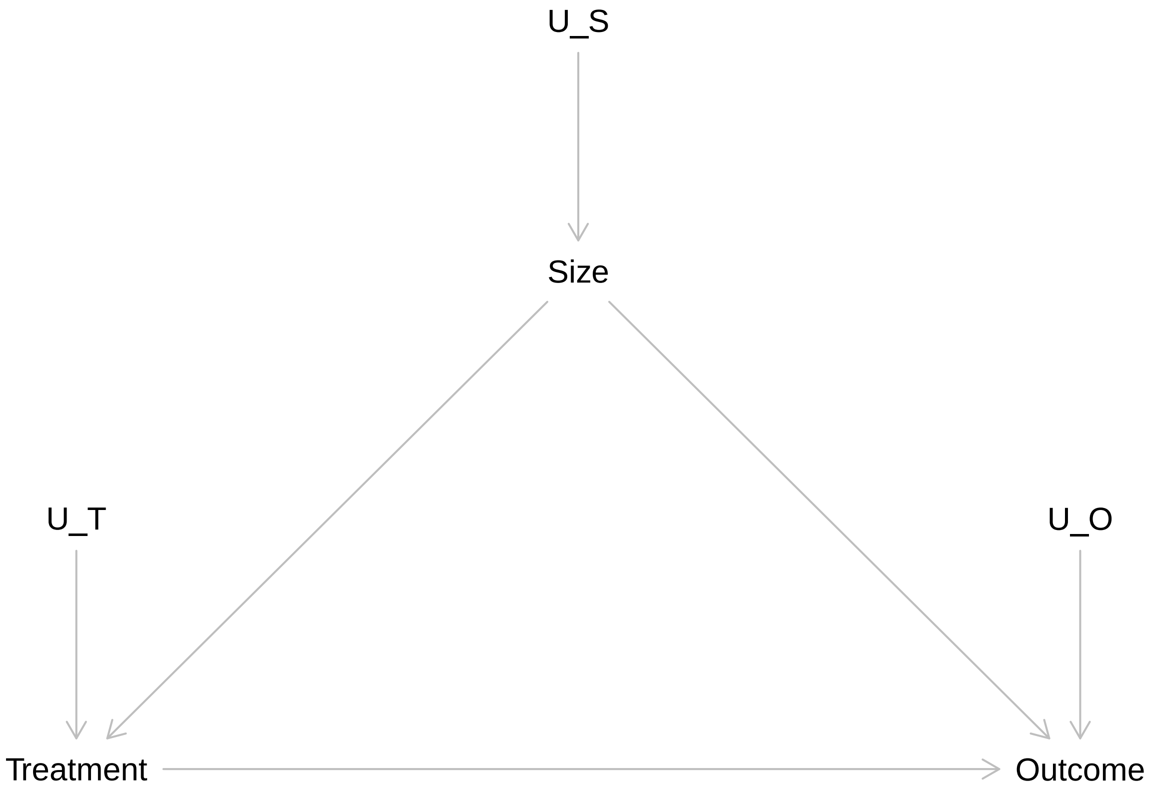
```
compare <- bind_cols(
  naive = naive_analysis %>% rename(Naive = P_Success),
  simple = simple_analysis %>% select(-Surgery, Simple = P_Success),
  pmx = pmx_analysis %>% select(-Surgery, PMX = P_Success) ,
  ipw = ipw_analysis %>% select(-Surgery, IPTW = P_Success)
)
compare
```

```
## # A tibble: 2 x 5
##   Surgery      Naive Simple   PMX  IPTW
##   <fct>        <dbl>  <dbl> <dbl> <dbl>
## 1 open         0.78   0.833 0.833 0.833
## 2 percutaneous 0.826  0.779 0.779 0.779
```

0.7 Necessary Causal Assumptions

The underlying causal scaffolding that must be assumed for any of the above analyses to be correct (with the exception of the naive analysis, which is incorrect in any case) is as follows:

```
scm <- dagitty('dag{
                  Treatment <- Size -> Outcome <- Treatment
                  U_T -> Treatment
                  U_S -> Size
                  U_O -> Outcome
                  }')
coordinates(scm) <- list(
 x = c(U_T = -1,  Treatment = -1, Size = 0, Outcome = 1, U_S = 0, U_O = 1 ),
 y = c(U_T = -sqrt(3)/2, Treatment = 0, Size = -sqrt(3),
       Outcome = 0, U_S = -sqrt(3)*1.5, U_O = -sqrt(3)/2)
)
plot(scm)
```


U_S
Size
U_T
U_O
Treatment
Outcome