

Data Generating Mechanism Testing

We create a single instantiation of the simulation in this block:

```
df <- make_regions(global_params)
head(df, 2)

##   region_id S      b W X1 X2      U delta A Yb.pre Yb.post Yb.post0 Yb.post1
## 1         1 1 0.285 3   1   0 1.10 0.318 0    125    125    125    125.1
## 2         1 1 0.260 1   0   0 1.24 0.196 0     62     62     62     62.2

dim(df)

## [1] 50000    13

df <- tibble::as_tibble(df)
```

Here, I test our ability to estimate the PATT (using 3 estimators) and also calculate the true PATT:

```
estimate_patt(df)

## [1] 0.276 0.276 0.276

true_patt(df)

## [1] 0.0356
```

Below, I print out simulated results of probabilities we have from the literature that we can use to help calibrate our parameters.

Table 1: B, SSP, System, and A by CPC+ or non-CPC+ region

S	P(B=1 S)	P(SSP S)	P(system S)	P(A S)
0	0.301	0.377	0.381	0.202
1	0.276	0.416	0.412	0.200

Table 2: Race by CPC+ Participation

A	P(B=1 S=1,A)
0	0.276
1	0.275

Table 3: Proportion of CPC+ Participation (in SSP, System, and Overall)

P(A=1 SSP=1, S=1)	P(A=1 sys=1, S=1)	P(A=1)
0.2	0.195	0.201

Diff-in-diff code

```
df_long <- gather(df, time, Y, Yb.pre:Yb.post)
df_long$time[df_long$time=="Yb.pre"] <- 0
df_long$time[df_long$time=="Yb.post"] <- 1

lm.1 <- lm(Y ~ A + time + A * time +
           X1+X2,
           data = df_long %>% filter(S==1) )

# Effect of CPC+ participation on expenditures
lm.1$coefficients["A"]
```

```
##           A
## -0.0181
```

Plotting DiD; code adapted from <https://rpubs.com/andrewheiss/did>

```
plot_data <- df_long %>%
  mutate(A = factor(A, labels = c("non-CPC+ participant", "CPC+ participant")),
         time = factor(time, labels = c("Pre-period", "Post-period"))) %>%
  group_by(A, time) %>%
  summarize(mean_Y = mean(Y),
            se_Y = sd(Y) / sqrt(n()),
            upper = mean_Y + (-1.96 * se_Y),
            lower = mean_Y + (1.96 * se_Y))
```

```
## `summarise()` has grouped output by 'A'. You can override using the `.groups`
## argument.
```

```
ggplot(plot_data, aes(x = time, y = mean_Y, color = A)) +
  geom_pointrange(aes(ymin = lower, ymax = upper), size = 1) +
  geom_line(aes(group = A))
```

