Post shock prediction simulations

Daniel Eck

November 26, 2019

We load in the R packages that we need.

```
library(matrixStats)
library(parallel)
library(doParallel)
library(foreach)
library(tidyverse)
```

Simulations with simple model

We set the random seed and declare the number of cores that our multicore implementation will use.

```
ncores <- detectCores() - 1
registerDoParallel(cores = ncores)
set.seed(13)
RNGkind("L'Ecuyer-CMRG")
nsim <- 2e3
```

Simple model simulations with normal errors

Combinations

```
ns <- c(5, 1:4 * 10)
Ts <- c(1:4 * 10)
mus <- 1:5 / 2
sds <- 1:4
sim_params <- expand.grid(list(ns = ns, Ts = Ts, mus = mus, sds = sds))
```

```
system.time({
  output_basic <- lapply(1:nrow(sim_params), FUN = function(j){
      n <- sim_params[j, 1]
      T <- sim_params[j, 2]
      mu.alpha <- sim_params[j, 3]
      sigma.alpha <- sim_params[j, 4]
      out <- foreach(i = 1:nsim, .combine = rbind) %dopar% {
        return(simstudy_normal(n = n, T = T, mu.alpha = mu.alpha,
        sigma.alpha = sigma.alpha, sigma.X = 1, sigma = 1))
      }
      out
  })
})
```

```
##      user    system   elapsed
## 25058.202   184.343  3937.391
```

```
dat_basic <- cbind(sim_params, do.call(rbind, lapply(output_basic, colMeans)),
  do.call(rbind, lapply(output_basic, function(x) colSds(x) / sqrt(nsim) )))
```

```r
colnames(dat_basic)[5:8] <- c("mean_noadj", "mean_adj", "sd_noadj", "sd_adj")
ggplot(dat_basic, aes(x = mus, y = mean_noadj - mean_adj,
  group = as.factor(sds), color = as.factor(sds))) +
  labs(title="Performance of adjustment via disparate information",
    subtitle = "(rows increase in T, columns increase in n)",
    x ="shock means", y = "error without adjusting minus error after adjusting",
    color = "shock sd") +
  geom_line() + geom_hline(yintercept = 0, color = "red") +
  theme_minimal() +
  scale_x_continuous(breaks=c(1,2), minor_breaks = NULL) +
  scale_y_continuous(minor_breaks = NULL) +
  facet_grid(Ts ~ ns)
```