9.3 randomized experiments

Chris Parrish

July 3, 2016

Contents

9.3 randomized experiments

reference:
- ARM chapter 09, github

```
library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
library(ggplot2)
```

randomized experiments

data

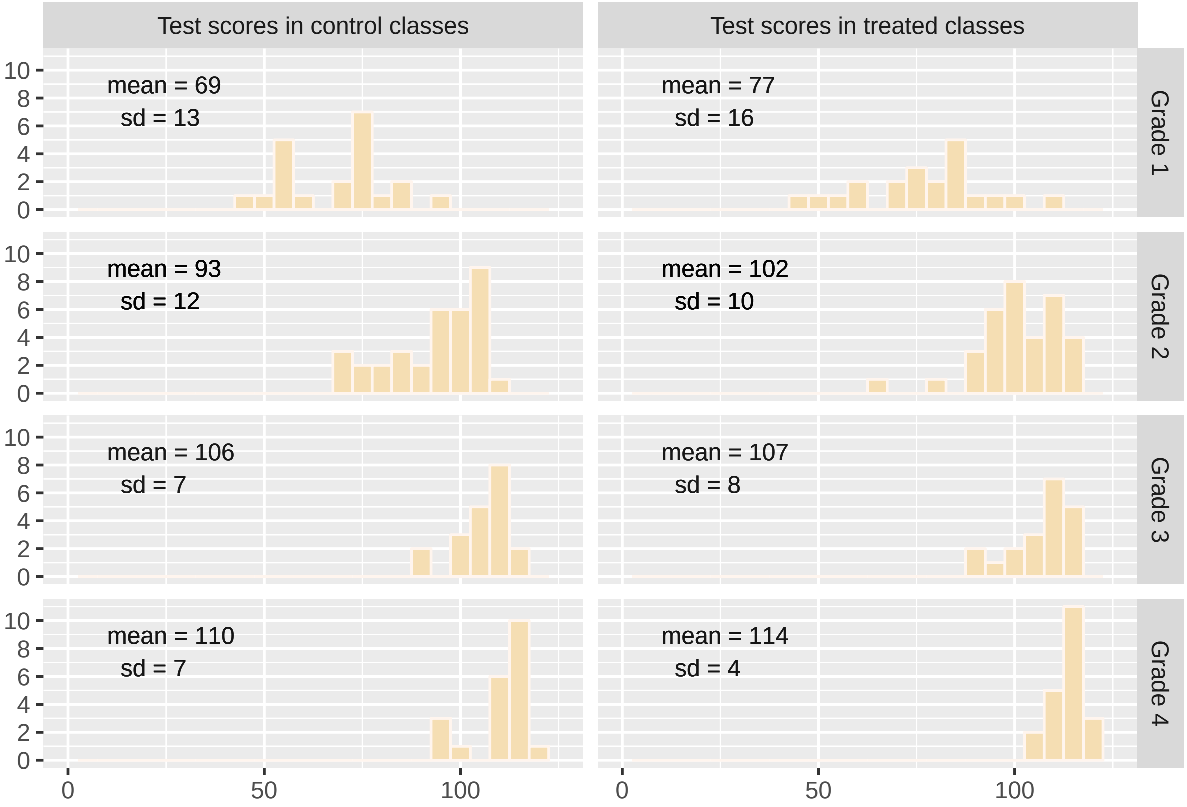
```
# data <- read.table("electric.dat", header=TRUE)
# electric <- data.frame(post_test = c(data$treated.Posttest, data$control.Posttest),
#                        pre_test = c(data$treated.Pretest, data$control.Pretest),
#                        grade = rep(data$Grade, 2),
#                        treatment = rep (c(1,0), rep(length(data$treated.Posttest),2)),
#                        supp = c(as.numeric(data[,"Supplement."])-1, rep(NA,nrow(data))))
# for (i in 1:4) {
#     temp <- electric[electric$grade == i,]
#     N <- nrow(temp)
#     attach(temp)
#     stan_rdump(c("N", "post_test", "pre_test", "grade", "treatment"),
#                 file=paste("electric_grade", i, ".data.R", sep=""))
#     detach(temp)
#     temp <- electric[(electric$grade == i) & (!is.na(electric$supp)),]
#     N <- nrow(temp)
#     attach(temp)
#     stan_rdump(c("N", "post_test", "pre_test", "grade", "treatment", "supp"),
#                 file=paste("electric_grade", i, "_supp.data.R", sep=""))
#     detach(temp)
# }
# rm(temp)
```

```
### Data
# electric_gradeX.data.R, where X = 1, 2, 3, 4
```

```
# Plot of the raw data (Figure 9.4)
electric.ggdf <- data.frame(c(), c(), c(), c(), c()) # empty data frame
for (i in 1:4) {
    source(paste("electric_grade", i, ".data.R", sep = ""))
    means <- round(as.vector(by(post_test, treatment, mean)), 0);
    sds <- round(as.vector(by(post_test, treatment, sd)), 0);
    electric.ggdf <- rbind(electric.ggdf, data.frame(
        post_test, pre_test, grade = paste("Grade", grade), treatment,
        mean = means[treatment+1], sd = sds[treatment+1]))
}
electric.ggdf$treatment <- factor(electric.ggdf$treatment)
levels(electric.ggdf$treatment) <- c("Test scores in control classes",
                                     "Test scores in treated classes")
```

figure

```
p1 <- ggplot(electric.ggdf, aes(x = post_test)) +
    geom_histogram(color = "seashell", fill = "wheat", binwidth = 5) +
    geom_text(aes(x = 10, y = 6, label = paste("mean =", mean, "\n  sd =", sd)),
              hjust = 0, vjust = 0, size = 3, alpha = 0.1) +
    facet_grid(grade ~ treatment) +
    scale_x_continuous("", limits = c(0, 125), breaks = c(0, 50, 100)) +
#    scale_y_continuous("", breaks = c())
    scale_y_continuous("", breaks = seq(0, 10, 2))
print(p1)
```



```
### Basic analysis of a completely randomized experiment

## Models:
#  post_test ~ treatment
#  post_test ~ treatment + pre_test
```

model

electric_tr.stan

```
data {
  int<lower=0> N;
  vector[N] post_test;
  vector[N] treatment;
}
parameters {
  vector[2] beta;
  real<lower=0> sigma;
}
model {
  post_test ~ normal(beta[1] + beta[2] * treatment, sigma);
}
```

fit

model

electric_trpre.stan

```
data {
  int<lower=0> N;
  vector[N] post_test;
  vector[N] treatment;
  vector[N] pre_test;
}
parameters {
  vector[3] beta;
  real<lower=0> sigma;
}
model {
  post_test ~ normal(beta[1] + beta[2] * treatment + beta[3] * pre_test, sigma);
}
```

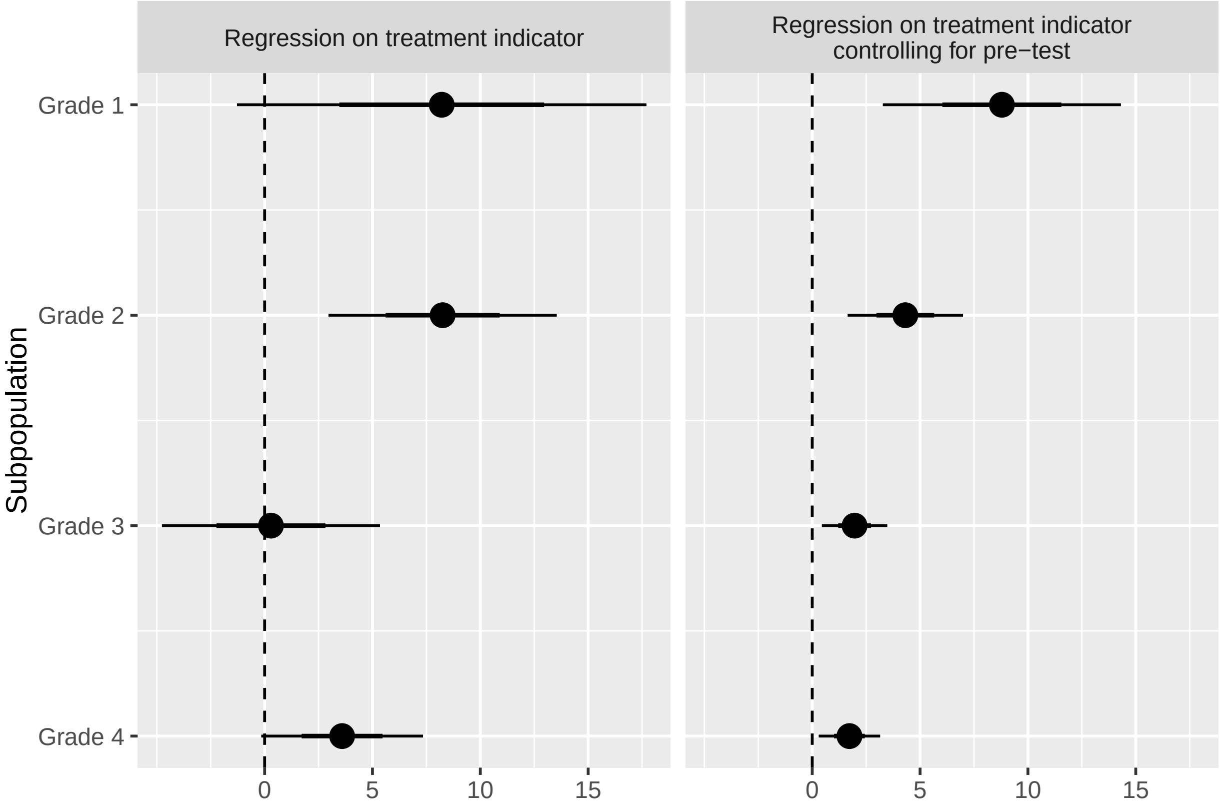
fit

```r
## Plot of the regression results (Figure 9.5)

alpha2 <- theta1 <- theta2 <- beta2 <- rep(NA, 4)
se1  <- se2 <- rep(NA,4)

# empty data frame (for Figure 9.6)
prepost.ggdf <- data.frame(c(), c(), c(), c(), c(), c(), c())

for (i in 1:4) {
    source(paste("electric_grade", i, ".data.R", sep = ""))
    temp <- data.frame(post_test, pre_test, grade, treatment)
    data.list <- c("N", "post_test", "pre_test", "treatment")
    sf.1 <- stan(file='electric_tr.stan', data=data.list,
                 iter=1000, chains=4)
    beta.post <- extract(sf.1, "beta")$beta
    theta1[i] <- mean(beta.post[,2])
    se1[i]    <- sd(beta.post[,2])
    sf.2 <- stan(file='electric_trpre.stan', data=data.list,
                 iter=1000, chains=4)
    beta.post <- extract(sf.2, "beta")$beta
    alpha2[i] <- mean(beta.post[,1])
    theta2[i] <- mean(beta.post[,2])
    beta2[i]  <- mean(beta.post[,3])
    se2[i]    <- sd(beta.post[,2])
    temp$alpha <- alpha2[i]
    temp$theta <- theta2[i]
    temp$beta  <- beta2[i]
    prepost.ggdf <- rbind(prepost.ggdf, temp)
}

theta.ggdf <- data.frame(c(), c(), c()) # empty data frame
```

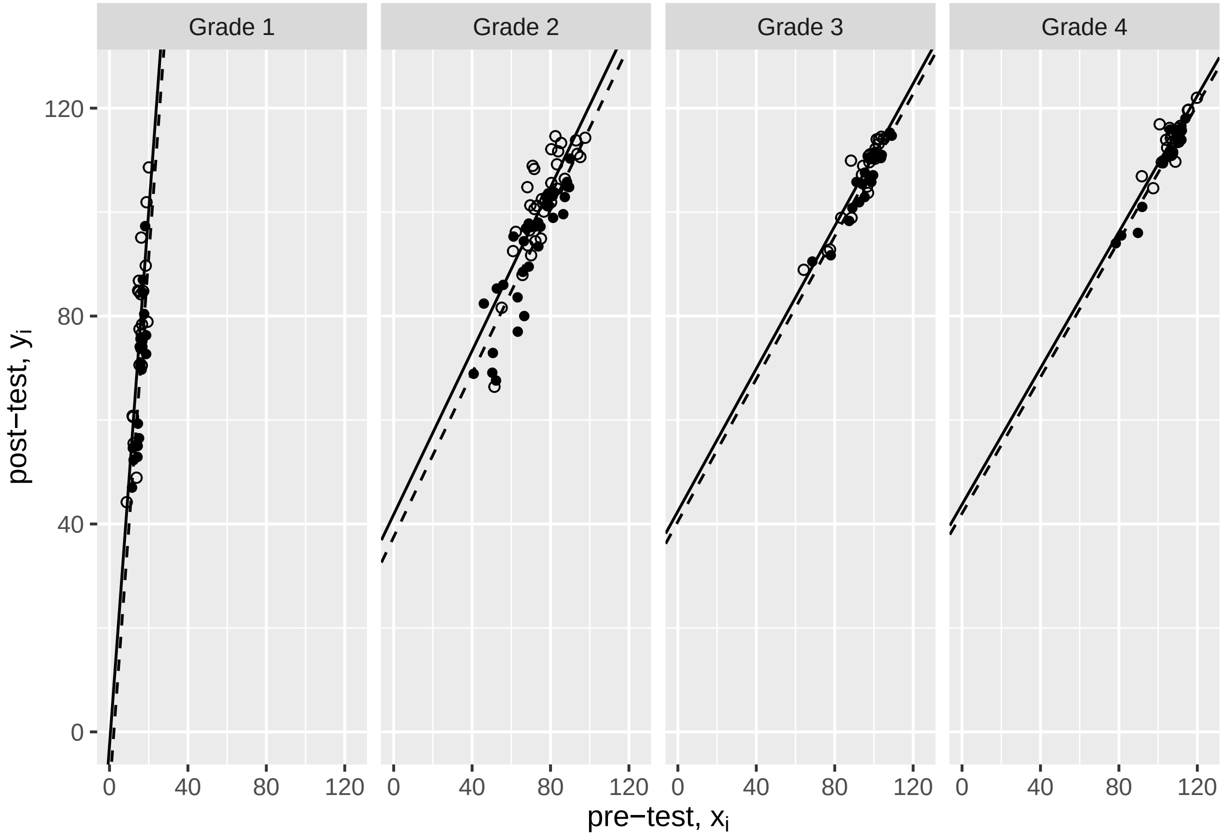
```
for (i in 1:4) {
    theta.ggdf <- rbind(theta.ggdf,
                        data.frame(est = theta1[i], se = se1[i], pretest = 0, grade = i))
    theta.ggdf <- rbind(theta.ggdf,
                        data.frame(est = theta2[i], se = se2[i], pretest = 1, grade = i))
}

theta.ggdf$pretest <- factor(theta.ggdf$pretest)
levels(theta.ggdf$pretest) <- c("Regression on treatment indicator",
                      "Regression on treatment indicator\ncontrolling for pre-test")
```

figure

```
# dev.new()
p2 <- ggplot(theta.ggdf, aes(x = 5 - grade, y = est)) +
    geom_pointrange(aes(ymin = est - se, ymax = est + se), size = 0.8) +
    geom_pointrange(aes(ymin = est - 2 * se, ymax = est + 2 * se), size = 0.5) +
    geom_hline(aes(yintercept = 0), linetype = "dashed") +
    facet_grid(. ~ pretest) +
    scale_x_continuous("Subpopulation", breaks = seq(1,4),
                       labels = paste("Grade", seq(4,1,-1))) +
    scale_y_continuous("", breaks = seq(0, 15, 5)) +
    coord_flip()
print(p2)
```



```r
## Figure 9.6
prepost.ggdf$grade <- factor(prepost.ggdf$grade)
levels(prepost.ggdf$grade) <- paste("Grade", levels(prepost.ggdf$grade))
prepost.ggdf$treatment <- factor(prepost.ggdf$treatment)
# dev.new()
p3 <- ggplot(prepost.ggdf, aes(x = pre_test, y = post_test)) +
    geom_point(aes(shape=treatment)) +
    scale_shape_manual(values = c(16, 1)) +
    geom_abline(aes(intercept = alpha + theta * (as.numeric(treatment)-1),
                    slope = beta, linetype = treatment)) +
    scale_linetype_manual(values = c(2, 1)) +
    facet_grid(. ~ grade) +
    scale_x_continuous(expression(paste("pre-test, ", x[i])),
                       limits = c(0, 125)) +
    scale_y_continuous(expression(paste("post-test, ", y[i])),
                       limits = c(0, 125)) +
    theme(legend.position = "none")
print(p3)
```


Grade 1
Grade 2
Grade 3
Grade 4
post−test, y_i
pre−test, x_i