ANCOVA Comparison Simulations: Continuous Outcomes

Kellie Ottoboni

2017-09-05

Continuous data

Step 1: We generated continuous potential outcomes. We used two scenarios. In the first, the treatment effect was constant across strata. We drew a latent random variate v_{ij} from the uniform distribution on $[-4, 4]$. In the second scenario, the treatment effect varied across strata, and we drew the latent random variable according to

$$v_{ij} \sim \begin{cases} \text{Unif}[-4,-1] & : j = 1 \\ \text{Unif}[-1,1] & : j = 2 \\ \text{Unif}[1,4] & : j = 3 \end{cases}$$

We generated independent and identically distributed errors ε_{ij} and δ_{ij} and varied the error distribution. The errors were either standard normal (to mimic the usual ANCOVA assumptions), t distributed with two degrees of freedom, standard lognormal, or exponentially distributed, with scale parameter 1 and shifted to have mean zero. The observed $(v_{ij}, \varepsilon_{ij}, \delta_{ij})$ were independent across i and j.

The baseline value for individual i, j was

$$X_{ij} = \frac{\gamma e^{v_{ij}} + e^{v_{ij}/2}}{2} + \varepsilon_{ij}$$

.

Then we generated potential outcomes as

$$Y_{ij}(Z_{ij}) = \frac{(2Z_{ij}-1)\gamma e^{v_{ij}} + e^{v_{ij}/2}}{2} + \delta_{ij}.$$

The treatment effect for individual (i, j) is $\gamma e^{v_{ij}}$ and we would like to estimate the average treatment effect in the sample.

Assume that there are $n_j = 16$ individuals per stratum and treatment assignment is balanced, i.e. 8 people receive each treatment at each stratum. After sampling $(v_{ij}, \varepsilon_{ij}, \delta_{ij})$, we regenerate Z 10,000 times. We repeat this procedure for each distribution of latent variables v and of the errors ε and δ.

Step 2: We regenerate Z and recompute $Y_(Z)$ 10,000 times for each design.

Step 3: We repeat step 1 and 2 for different distributions on ε and δ.

In expectation, the average treatment effect is γ. We compare the empirical power of five tests to detect this treatment effect:

- ANCOVA: we fit a linear model of response Y_1 on baseline Y_0, treatment Z, and a dummy for stratum.
- Stratified permutation: we permute treatment assignment within stratum, then take the difference in means between treated and control outcomes Y_1
- Differenced permutation: we do the same permutation procedure as the stratified permutation test, except we use the difference between outcome and baseline, $Y_1 - Y_0$

- Linear model (LM) permutation: we use the same stratified permutation procedure as above, except use the t-statistic for the coefficient on treatment in the linear regression of Y_1 on Y_0, Z, and stratum dummies
- Freedman-Lane test: see the other Rmd document for a full description of this procedure

Data-generation, tests, and plotting functions

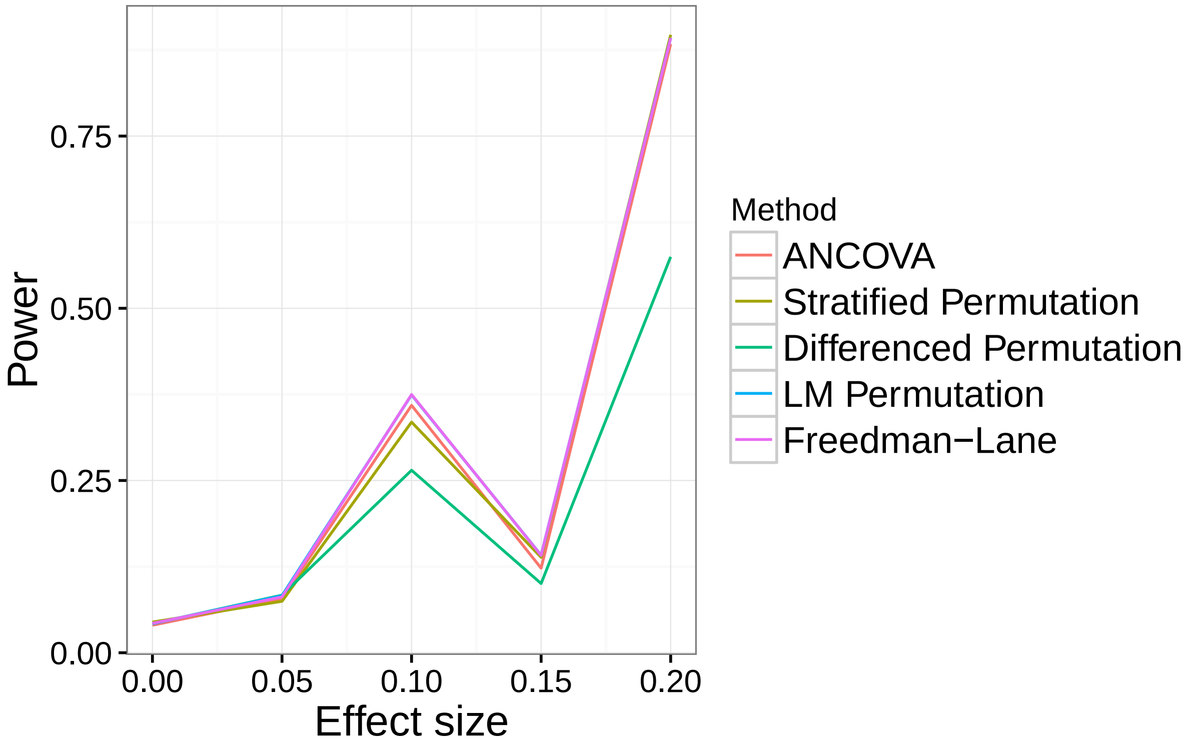
```
gen_y1 <- function(gamma, v, error, Z) {
    y1 <- 0.5 * ((2 * Z - 1) * gamma * exp(v) + exp(v/2)) + error
    return(y1)
}

gen_y0 <- function(gamma, v, error) {
    y0 <- 0.5 * (-1 * gamma * exp(v) + exp(v/2)) + error
    return(y0)
}

generate_simulated_data <- function(gamma, effect, errors, n = c(16,
    16, 16)) {
    # Input: gamma = multiplier for the magnitude of the
    # treatment effect effect = 'same effect' or 'heterogeneous'
    # errors = 'normal' or 'heavy' n = number of individuals at
    # each stratum Returns: a dataframe containing columns named
    # Y1 (response), YO (baseline), Z (treatment), gamma_vec
    # (treatment effect per individual), stratumID (stratum),
    # stratum_effect (beta coefficient per individual), and
    # epsilon (errors)

    stratumID <- rep(1:3, times = n)
    N <- sum(n)

    # What is the treatment effect?
    if (effect == "same effect") {
        v <- runif(N, min = -4, max = 4)
    } else if (effect == "heterogeneous") {
        v <- rep(0, N)
        v[stratumID == 1] <- runif(n[1], min = -4, max = -1)
        v[stratumID == 2] <- runif(n[2], min = -1, max = 1)
        v[stratumID == 3] <- runif(n[3], min = 1, max = 4)
    } else {
        stop("invalid parameter effect")
    }

    # Generate errors
    if (errors == "normal") {
        epsilon <- rnorm(N)
        delta <- rnorm(N)
    } else if (errors == "t") {
        epsilon <- rt(N, df = 2)
        delta <- rt(N, df = 2)
    } else if (errors == "lognormal") {
        epsilon <- rlnorm(N)
```

```r
        delta <- rlnorm(N)
    } else if (errors == "exponential") {
        epsilon <- rexp(N) - 1
        delta <- rexp(N) - 1
    } else {
        stop("invalid errors parameter")
    }

    # Generate covariates
    Z <- rep(0:1, length.out = N)
    Y0 <- gen_y0(gamma, v, epsilon)
    Y1 <- gen_y1(gamma, v, delta, Z)
    return(data.frame(Y1, Y0, Z, v, stratumID, epsilon, delta))
}

generate_simulated_pvalues <- function(dataset, reps = 1000) {
    # Inputs: dataset = a dataframe containing columns named Y1
    # (response), Y0 (baseline), Z (treatment), and stratumID
    # (stratum) Returns: a vector of p-values first element is
    # the p-value from the ANCOVA second element is the p-value
    # from the stratified two-sample permutation test third
    # element is the p-value from the linear model test,
    # permuting treatment fourth element is the p-value from the
    # Freedman-Lane linear model test, permuting residuals

    # ANCOVA
    modelfit <- lm(Y1 ~ Y0 + Z + factor(stratumID), data = dataset)
    resanova <- summary(aov(modelfit))
    anova_pvalue <- resanova[[1]]["Z", "Pr(>F)"]

    # Stratified permutation test of Y1
    observed_diff_means <- mean(dataset$Y1[dataset$Z == 1]) -
        mean(dataset$Y1[dataset$Z == 0])
    diff_means_distr <- stratified_two_sample(group = dataset$Z,
        response = dataset$Y1, stratum = dataset$stratumID, reps = reps)
    perm_pvalue <- t2p(observed_diff_means, diff_means_distr,
        alternative = "two-sided")

    # Diffed permutation test of Y1-Y0
    dataset$diff <- dataset$Y1 - dataset$Y0
    observed_diff_means2 <- mean(dataset$diff[dataset$Z == 1]) -
        mean(dataset$diff[dataset$Z == 0])
    diff_means_distr2 <- stratified_two_sample(group = dataset$Z,
        response = dataset$diff, stratum = dataset$stratumID,
        reps = reps)
    perm_pvalue2 <- t2p(observed_diff_means2, diff_means_distr2,
        alternative = "two-sided")

    # Permutation of treatment in linear model
    observed_t1 <- summary(modelfit)[["coefficients"]]["Z", "t value"]

    # Freedman-Lane linear model residual permutation
    lm2_no_tr <- lm(Y1 ~ Y0 + factor(stratumID), data = dataset)
```

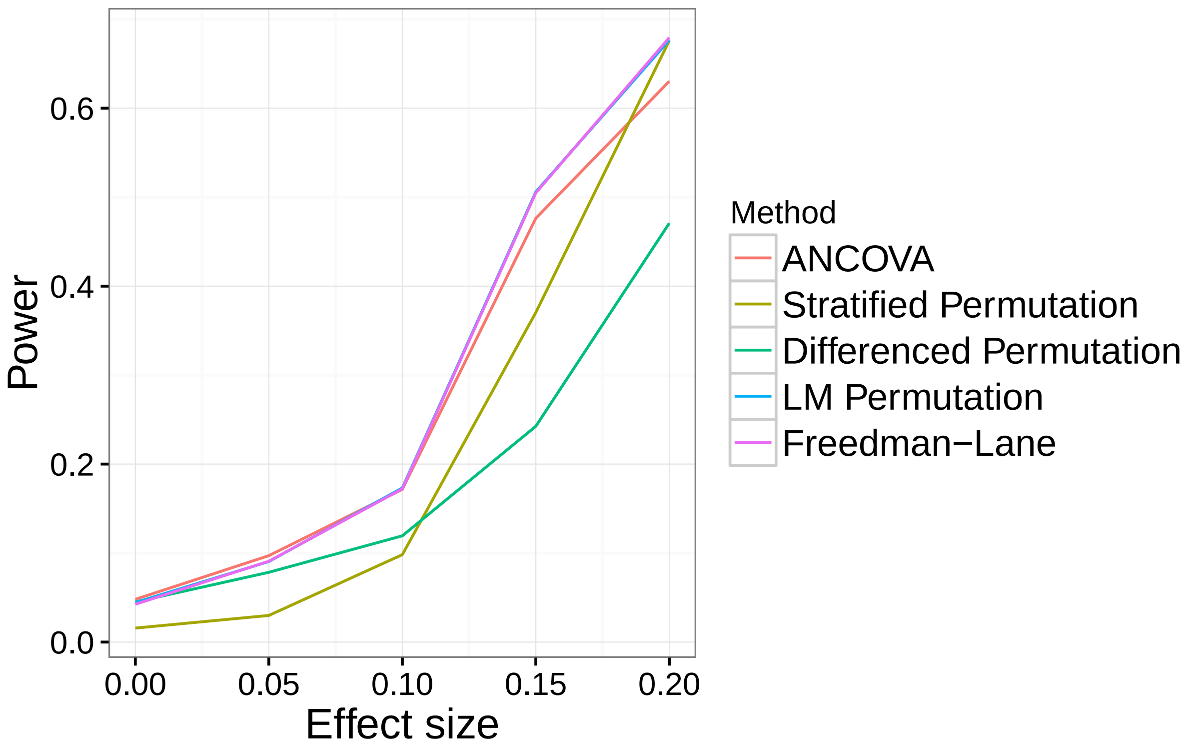
```r
    dataset$lm2_resid <- residuals(lm2_no_tr)
    lm2_yhat <- fitted(lm2_no_tr)

    lm1and2_t_distr <- replicate(reps, {
        dataset[, c("Z_perm", "lm2_resid_perm")] <- permute_within_groups(dataset[,
            c("Z", "lm2_resid")], dataset$stratumID)
        lm1_perm <- lm(Y1 ~ Y0 + Z_perm + factor(stratumID),
            data = dataset)

        dataset$response_fl <- lm2_yhat + dataset$lm2_resid_perm
        lm2_perm <- lm(response_fl ~ Y0 + Z + factor(stratumID),
            data = dataset)

        c(summary(lm1_perm)[["coefficients"]]["Z_perm", "t value"],
            summary(lm2_perm)[["coefficients"]]["Z", "t value"])
    })
    lm_pvalue <- t2p(observed_t1, lm1and2_t_distr[1, ], alternative = "two-sided")
    fl_pvalue <- t2p(observed_t1, lm1and2_t_distr[2, ], alternative = "two-sided")

    return(c(ANCOVA = anova_pvalue, `Stratified Permutation` = perm_pvalue,
        `Differenced Permutation` = perm_pvalue2, `LM Permutation` = lm_pvalue,
        `Freedman-Lane` = fl_pvalue))
}
```

```r
compute_power <- function(pvalues) {
    sapply((0:99)/100, function(p) mean(pvalues <= p, na.rm = TRUE))
}

plot_power_curves <- function(power_mat, title) {
    melt(power_mat) %>% mutate(pvalue = Var1/100) %>% mutate(Method = Var2) %>%
        ggplot(aes_string(x = "pvalue", y = "value", color = "Method")) +
        geom_line() + geom_abline(intercept = 0, slope = 1, linetype = "dashed") +
        xlab("P-value") + ylab("Power") + ggtitle(title) + theme_bw() +
        theme(axis.text.x = element_text(size = 12), axis.text.y = element_text(size = 12),
            axis.title = element_text(size = 16), title = element_text(size = 16),
            legend.title = element_text(size = 12), legend.text = element_text(size = 14),
            strip.text.x = element_text(size = 12))
}

plot_power_gamma <- function(powermat_list, gamma_vec, alpha,
    title) {
    gamma_power_alpha <- t(sapply(powermat_list, function(x) x[floor(alpha *
        nrow(x)), ]))
    colnames(gamma_power_alpha) <- c("ANCOVA", "Stratified Permutation",
        "Differenced Permutation", "LM Permutation", "Freedman-Lane")
    melt(gamma_power_alpha, value.name = "Power") %>% mutate(Method = Var2) %>%
        mutate(Effect = rep(gamma_vec, 5)) %>% ggplot(aes(x = Effect,
        y = Power)) + geom_line(aes(color = Method)) + xlab("Effect size") +
        theme_bw() + theme(axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12), axis.title = element_text(size = 16),
        title = element_text(size = 16), legend.title = element_text(size = 12),
        legend.text = element_text(size = 14), strip.text.x = element_text(size = 12))
}
```

```r
load("continuous_outcomes/vary_gamma_results.Rda")
gamma_vec <- seq(0, 0.2, by = 0.05)
gamma_power <- lapply(gamma_res, function(x) apply(x, 2, compute_power))
plot_power_gamma(gamma_power, gamma_vec, 0.05, "Power at level 5%, Constant Treatment Effect")
```




```r
load("continuous_outcomes/vary_gamma_het_results.Rda")
gamma_vec <- seq(0, 0.2, by = 0.05)
gamma_power_het <- lapply(gamma_res_het, function(x) apply(x,
    2, compute_power))
plot_power_gamma(gamma_power_het, gamma_vec, 0.05, "Power at level 5%, Heterogeneous Treatment Effect")
```




```
gaussian_homogeneous <- gamma_power[[which(abs(gamma_vec - 0.2) <
    1e-06)]]
```

```
load("continuous_outcomes/gaussian_heterogeneous.Rda")
```

```
load("continuous_outcomes/t_homogeneous.Rda")
```

```
load("continuous_outcomes/t_heterogeneous.Rda")
```

```
load("continuous_outcomes/lognormal_homogeneous.Rda")
```

```
load("continuous_outcomes/lognormal_heterogeneous.Rda")
```

```
load("continuous_outcomes/exponential_homogeneous.Rda")
```

```
load("continuous_outcomes/exponential_heterogeneous.Rda")
```

```
powers <- list(gaussian_homogeneous, t_homogeneous, lognormal_homogeneous,
    exponential_homogeneous, gaussian_heterogeneous, t_heterogeneous,
    lognormal_heterogeneous, exponential_heterogeneous)
summary05 <- t(sapply(powers, function(x) x[5, ]))
summary05 <- data.frame(rep(c("Normal", "t(2)", "Log Normal",
    "Exponential"), 2), summary05)
colnames(summary05) <- c("Errors", "ANCOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
```

```
summarytab_constant <- xtable(summary05[1:4, ], digits = 3, caption = "Empirical power at level $0.05$ f
    label = "tab:power_grid1")
print(summarytab_constant, include.rownames = FALSE)
```

```
summarytab_het <- xtable(summary05[5:8, ], digits = 3, caption = "Empirical power at level $0.05$ for si
    label = "tab:power_grid2")
print(summarytab_het, include.rownames = FALSE)
```

Errors	ANCOVA	Stratified Permutation	Differenced Permutation	LM Permutation	Freedman-Lane
Normal	0.884	0.897	0.575	0.893	0.892
t(2)	0.286	0.316	0.146	0.292	0.291
Log Normal	0.450	0.472	0.178	0.463	0.465
Exponential	0.540	0.588	0.468	0.552	0.552

Table 1: Empirical power at level 0.05 for simulated data with constant additive treatment effects

Errors	ANCOVA	Stratified Permutation	Differenced Permutation	LM Permutation	Freedman-Lane
Normal	0.658	0.548	0.317	0.669	0.669
t(2)	0.359	0.315	0.139	0.365	0.362
Log Normal	0.487	0.361	0.155	0.507	0.509
Exponential	0.631	0.543	0.407	0.636	0.633

Table 2: Empirical power at level 0.05 for simulated data with heterogeneous treatment effects