ANCOVA Comparison Simulations: Skewed, Poisson Data

Kellie Ottoboni

2017-02-15

Poisson data

As in the other simulations that use Gaussian data, we assume the following linear data-generating process:

$$\tilde{Y}_{ij1} = \beta_0 Y_{ij0} + \beta_j + \gamma_j Z_{ij} + \varepsilon_{ij}$$

for individuals $i = 1, \dots, n_j$, $j = 1, \dots, J$. β_0 is the coefficient for the baseline measurement Y_{i0}, β_j is the mean effect of being at stratum j, Z_{ij} is the treatment level indicator,, γ_j is the effect of treatment at stratum j, and ε_{ij} takes the values 0 or 0.5 with equal probability. We will assume that $\beta_0 = 1$.

Now, instead of Y_{ij0} being normally distributed, we let these baseline measurements come from a truncated Poisson distribution with mean 4, with values greater than 10 censored and set to 10. This is reflective of some survey data where individuals are asked to rate their experience on a scale from 1 to 10. The data are not only discrete but also skewed: more values are in the lower half of the range.

We don't observe $\tilde{Y}_{ij1}$ but Y_{ij1}, which is defined as

$$Y_{ij1} = \begin{cases} 1 & \text{if } \tilde{Y}_{ij1} < 1 \\ 10 & \text{if } \tilde{Y}_{ij1} > 10 \\ \lfloor \tilde{Y}_{ij1} \rfloor & \text{otherwise} \end{cases}$$

Suppose there are three strata with $\beta_1 = 1, \beta_2 = 1.5$, and $\beta_3 = 2$. Assume that there are 16 individuals per stratum and treatment assignment is balanced, i.e. 8 people receive each treatment at each stratum.

We use the same simulation guidelines as detailed in the Gaussian simulation file. We omit comments.

Data-generation, tests, and plotting functions

```
generate_outcome <- function(baseline, tr_effect, stratum_effect,
    error) {
    Y1 <- baseline + tr_effect + stratum_effect + error
    Y1 <- floor(Y1)
    Y1 <- pmax(pmin(Y1, 10), 1)
    return(Y1)
}

generate_simulated_data <- function(gamma, effect, n = c(16,
    16, 16)) {
    # Input: gamma = the magnitude of the treatment effect effect
    # = 'same effect' or 'single stratum effect' - which strata
    # have a tr effect > 0?  n = number of individuals at each
    # stratum Returns: a dataframe containing columns named Y1
    # (response), Y0 (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)
    beta <- c(1, 1.5, 2)

    # What is the treatment effect?
    if (effect == "same effect") {
        gamma_vec <- rep(gamma, N)
    } else {
        gamma_vec <- rep(c(gamma, 0, 0), times = n)
    }

    # Generate covariates
    epsilon <- rbinom(N, 1, 0.5) - 0.5
    Y0 <- rpois(N, lambda = 4)
    Y0[Y0 > 10] <- 10
    Z <- rep(0:1, length.out = N)
    stratum_effect <- rep(beta, times = n)
    Y1 <- generate_outcome(Y0, gamma_vec * Z, stratum_effect,
        epsilon)
    return(data.frame(Y1, Y0, Z, gamma_vec, stratumID, stratum_effect,
        epsilon))
}

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]) -
```

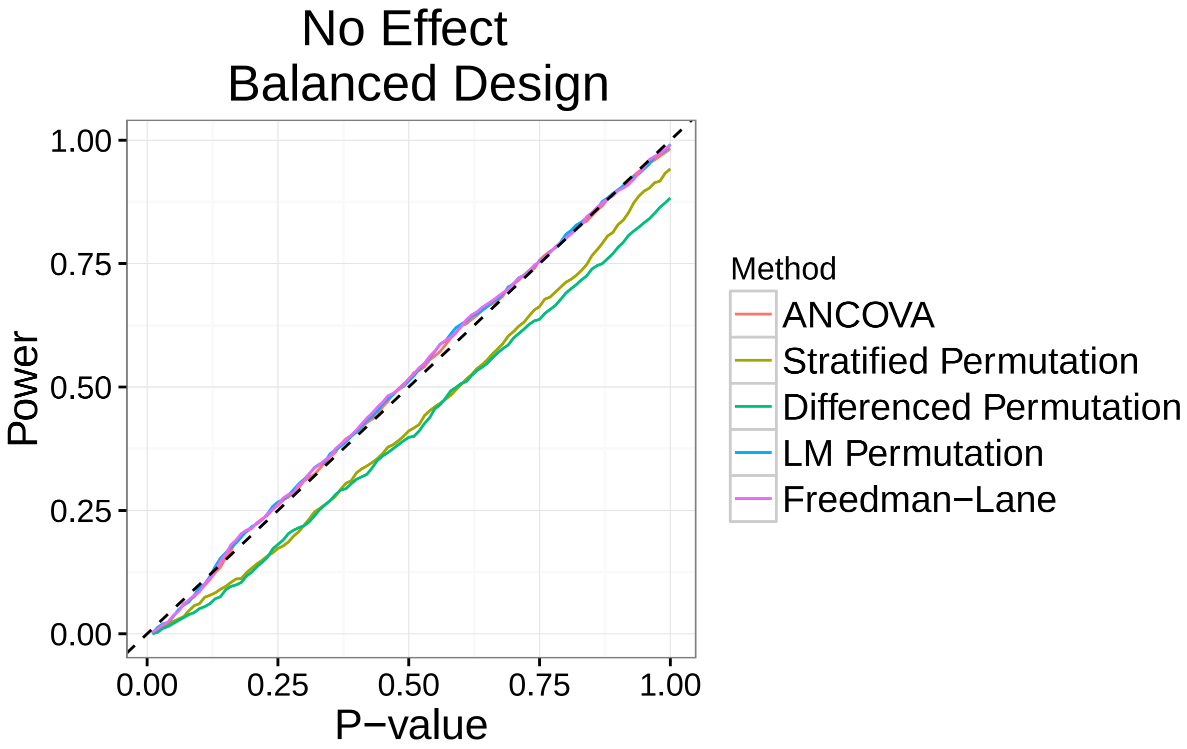
```r
        mean(dataset$diff[dataset$Z == 0])
    diff_means_distr2 <- stratified_two_sample(group = dataset$Z,
        response = dataset$diff, stratum = dataset$stratumID,
        reps = reps)
    perm_pvalue2 <- t2p(observed_diff_means2, diff_means_distr2,
        alternative = "two-sided")

    # Permutation of treatment in linear model
    observed_t1 <- summary(modelfit)[["coefficients"]]["Z", "t value"]
    lm1_t_distr <- replicate(reps, {
        dataset$Z_perm <- permute_within_groups(dataset$Z, dataset$stratumID)
        lm1_perm <- lm(Y1 ~ Y0 + Z_perm + factor(stratumID),
            data = dataset)
        summary(lm1_perm)[["coefficients"]]["Z_perm", "t value"]
    })
    lm_pvalue <- t2p(observed_t1, lm1_t_distr, alternative = "two-sided")

    # Freedman-Lane linear model residual permutation
    lm2_no_tr <- lm(Y1 ~ Y0 + factor(stratumID), data = dataset)
    lm2_resid <- residuals(lm2_no_tr)
    lm2_yhat <- fitted(lm2_no_tr)
    lm2_t_distr <- replicate(reps, {
        lm2_resid_perm <- permute_within_groups(lm2_resid, dataset$stratumID)
        dataset$response_fl <- lm2_yhat + lm2_resid_perm
        lm2_perm <- lm(response_fl ~ Y0 + Z + factor(stratumID),
            data = dataset)
        summary(lm2_perm)[["coefficients"]]["Z", "t value"]
    })
    fl_pvalue <- t2p(observed_t1, lm2_t_distr, 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))
}
```

Test level: simulation under the null

```
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC

tmp <- generate_simulated_data(gamma = 0, effect = "same effect")
design0_pvalues <- replicate(1000, {
    tmp$epsilon <- rnorm(nrow(tmp))
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$Y1 <- generate_outcome(tmp$Y0, rep(0, nrow(tmp)), tmp$stratum_effect,
        tmp$epsilon)
    generate_simulated_pvalues(tmp)
})
design0_pvalues <- t(design0_pvalues)
colnames(design0_pvalues) <- c("ANCOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design0_power <- apply(design0_pvalues, 2, compute_power)
```

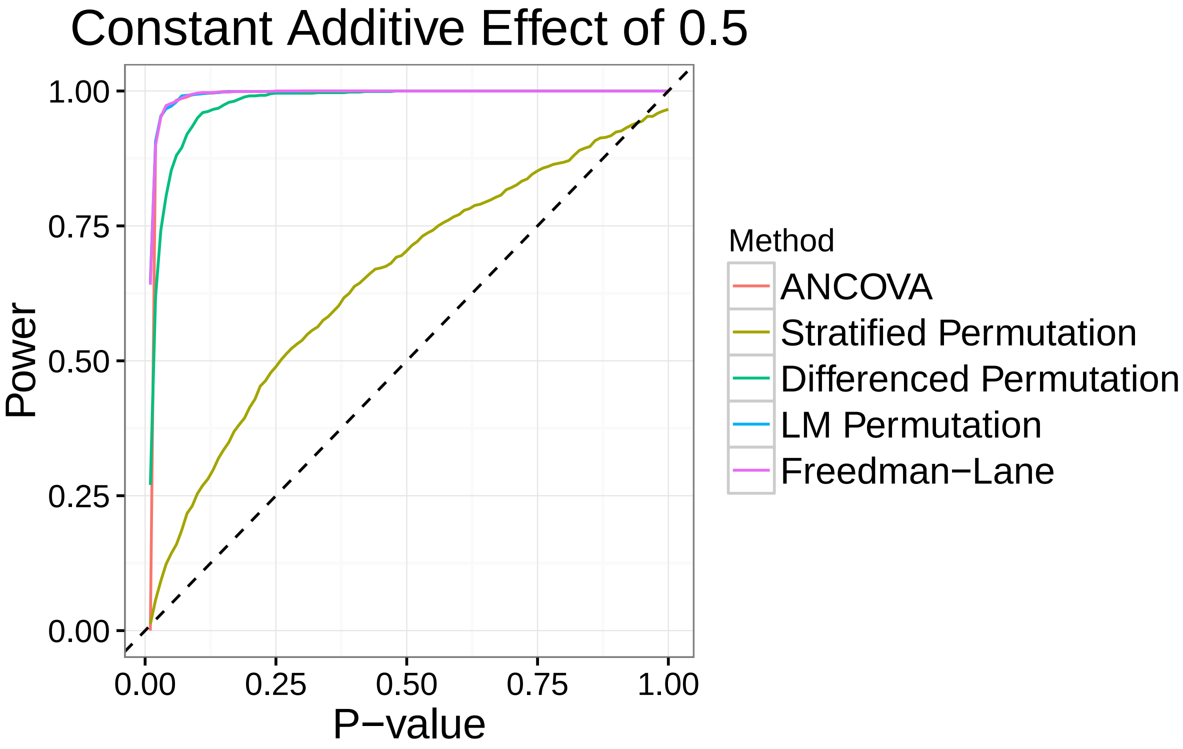
```
plot_power_curves(design0_power, "No Effect \n Balanced Design")
```



Design 1: Constant additive effect

```
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC

tmp <- generate_simulated_data(gamma = 0.5, effect = "same effect")
design1_pvalues <- replicate(1000, {
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$epsilon <- rbinom(nrow(tmp), 1, 0.5) - 0.5
    tmp$Y1 <- generate_outcome(tmp$Y0, tmp$Z * tmp$gamma_vec,
        tmp$stratum_effect, tmp$epsilon)
    generate_simulated_pvalues(tmp)
})

design1_pvalues <- t(design1_pvalues)
colnames(design1_pvalues) <- c("ANCOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design1_power <- apply(design1_pvalues, 2, compute_power)
```

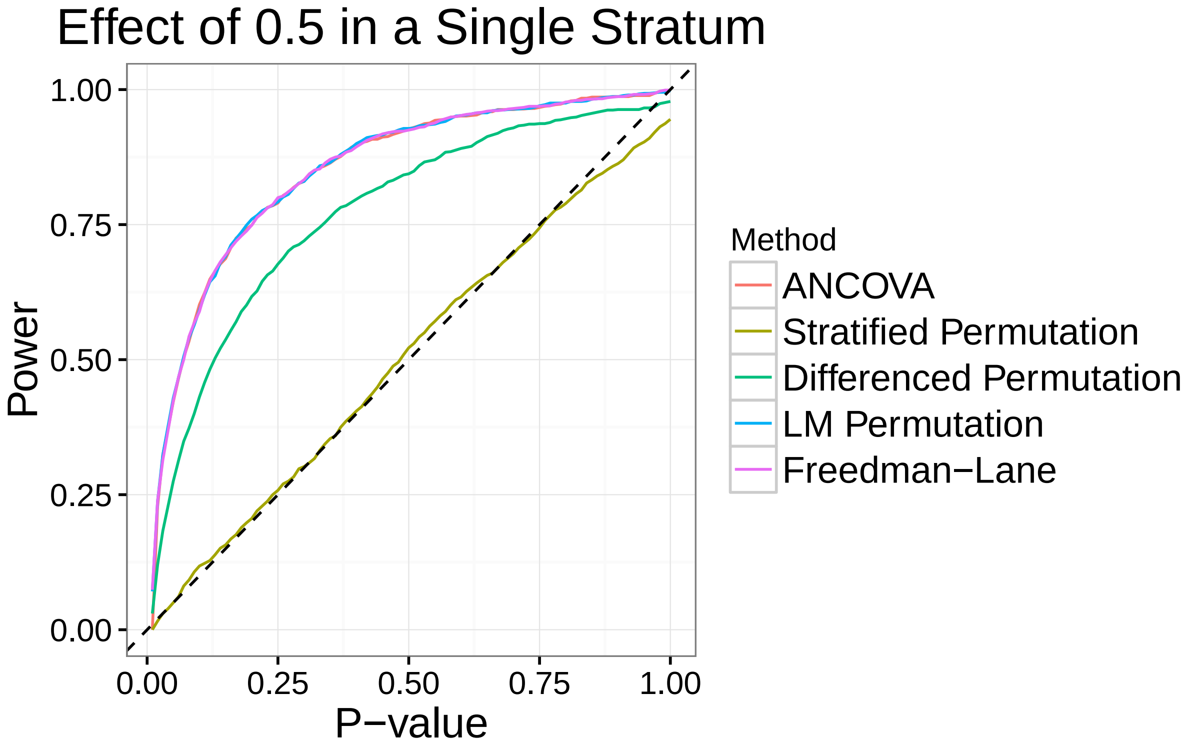
```
plot_power_curves(design1_power, "Constant Additive Effect of 0.5")
```



Design 2: Single stratum effect

```
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC

tmp <- generate_simulated_data(gamma = 0.5, effect = "single stratum effect")
design2_pvalues <- replicate(1000, {
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$epsilon <- rbinom(nrow(tmp), 1, 0.5) - 0.5
    tmp$Y1 <- generate_outcome(tmp$Y0, tmp$Z * tmp$gamma_vec,
        tmp$stratum_effect, tmp$epsilon)
    generate_simulated_pvalues(tmp)
})
design2_pvalues <- t(design2_pvalues)
colnames(design2_pvalues) <- c("ANCOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design2_power <- apply(design2_pvalues, 2, compute_power)
```

```
plot_power_curves(design2_power, "Effect of 0.5 in a Single Stratum")
```



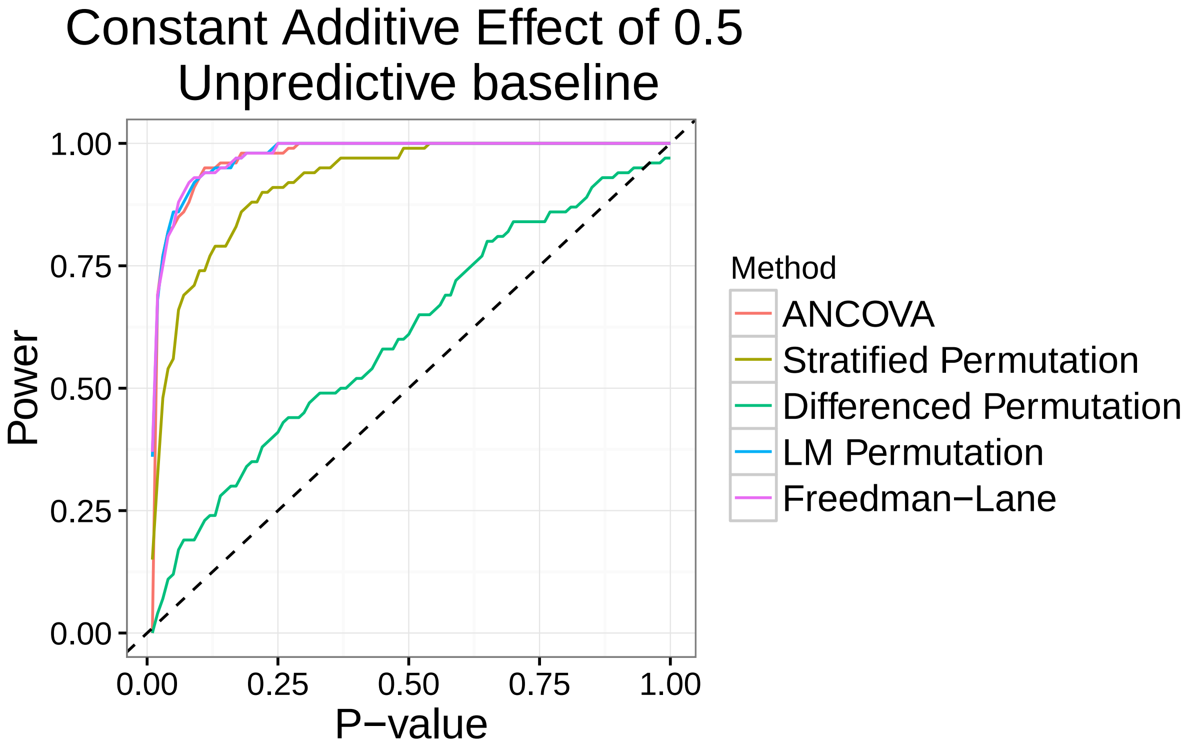
Design 3: Unpredictive baseline

Suppose that the baseline measure is not strongly predictive of the outcome. Then, we'd expect that controlling for baseline will not improve the power of the test, and may even make it worse. Suppose that in the linear data-generating process above, we have $\beta_0 = 0.25$, meaning that Y_1 and Y_0 have a correlation of 0.25.

The power curves for this design look similar to those from Design 1 with one important difference: the stratified permutation test has power very close to ANCOVA and the two linear model tests, while the differenced permutation test has lower power. This suggests that one should be careful when incorporating control variables; naively taking the difference $Y_1 - Y_0$ does not capture the correct relationship between baseline and outcome.

```
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC

tmp <- generate_simulated_data(gamma = 0.5, effect = "same effect")
design3_pvalues <- replicate(100, {
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$epsilon <- rbinom(nrow(tmp), 1, 0.5) - 0.5
    tmp$Y1 <- generate_outcome(0.1 * tmp$Y0, tmp$Z * tmp$gamma_vec,
        tmp$stratum_effect, tmp$epsilon)
    generate_simulated_pvalues(tmp)
})

design3_pvalues <- t(design3_pvalues)
colnames(design3_pvalues) <- c("ANCOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design3_power <- apply(design3_pvalues, 2, compute_power)
```

```r
plot_power_curves(design3_power, "Constant Additive Effect of 0.5 \n Unpredictive baseline")
```

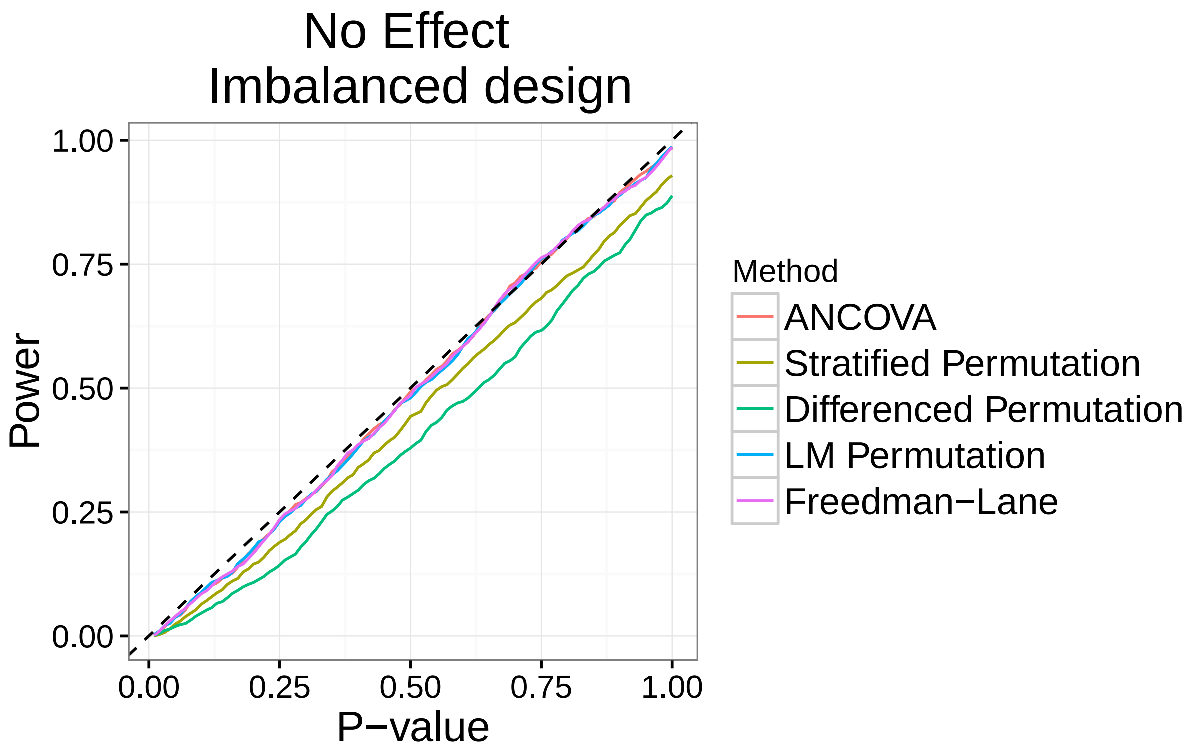


Test level: imbalanced design

Suppose that instead of having 16 individuals per stratum, we distribute them unequally across strata: stratum 1 has 8 patients, stratum 2 has 16 patients, and stratum 3 has 24 patients. This imbalance does not violate any assumptions of the ANCOVA. However, it may reduce power if the effect is concentrated in strata with fewer patients.

```r
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC

tmp <- generate_simulated_data(gamma = 0, effect = "same effect",
    n = c(8, 16, 24))
design00_pvalues <- replicate(1000, {
    tmp$epsilon <- rnorm(nrow(tmp))
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$Y1 <- generate_outcome(tmp$Y0, rep(0, nrow(tmp)), tmp$stratum_effect,
        tmp$epsilon)
    generate_simulated_pvalues(tmp)
})
design00_pvalues <- t(design00_pvalues)
colnames(design00_pvalues) <- c("ANCOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design00_power <- apply(design00_pvalues, 2, compute_power)
```

```r
plot_power_curves(design00_power, "No Effect \n Imbalanced design")
```

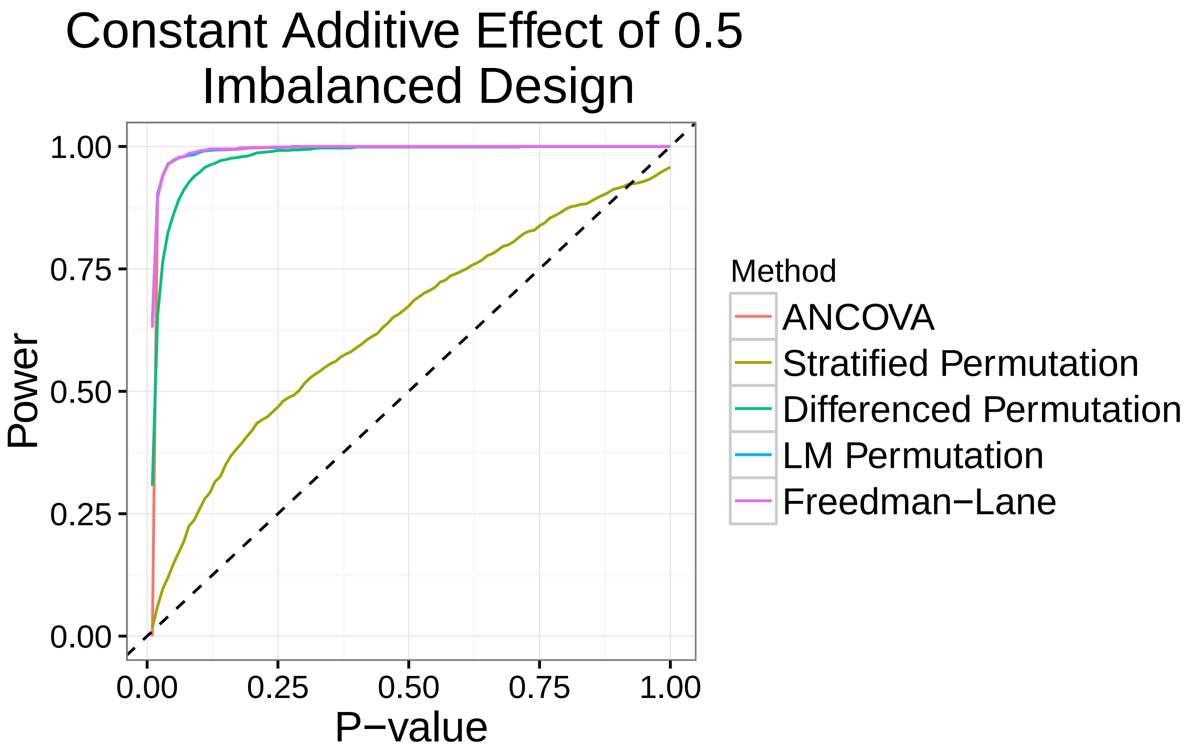


Design 4: Constant additive effect, imbalanced design

The power curves here are very similar to those in Design 1. This result is expected, since the effect is still present among all patients who received treatment.

```
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC

tmp <- generate_simulated_data(gamma = 0.5, effect = "same effect",
    n = c(8, 16, 24))
design4_pvalues <- replicate(1000, {
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$epsilon <- rbinom(nrow(tmp), 1, 0.5) - 0.5
    tmp$Y1 <- generate_outcome(tmp$Y0, tmp$Z * tmp$gamma_vec,
        tmp$stratum_effect, tmp$epsilon)
    generate_simulated_pvalues(tmp)
})

design4_pvalues <- t(design4_pvalues)
colnames(design4_pvalues) <- c("ANCOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design4_power <- apply(design4_pvalues, 2, compute_power)
```

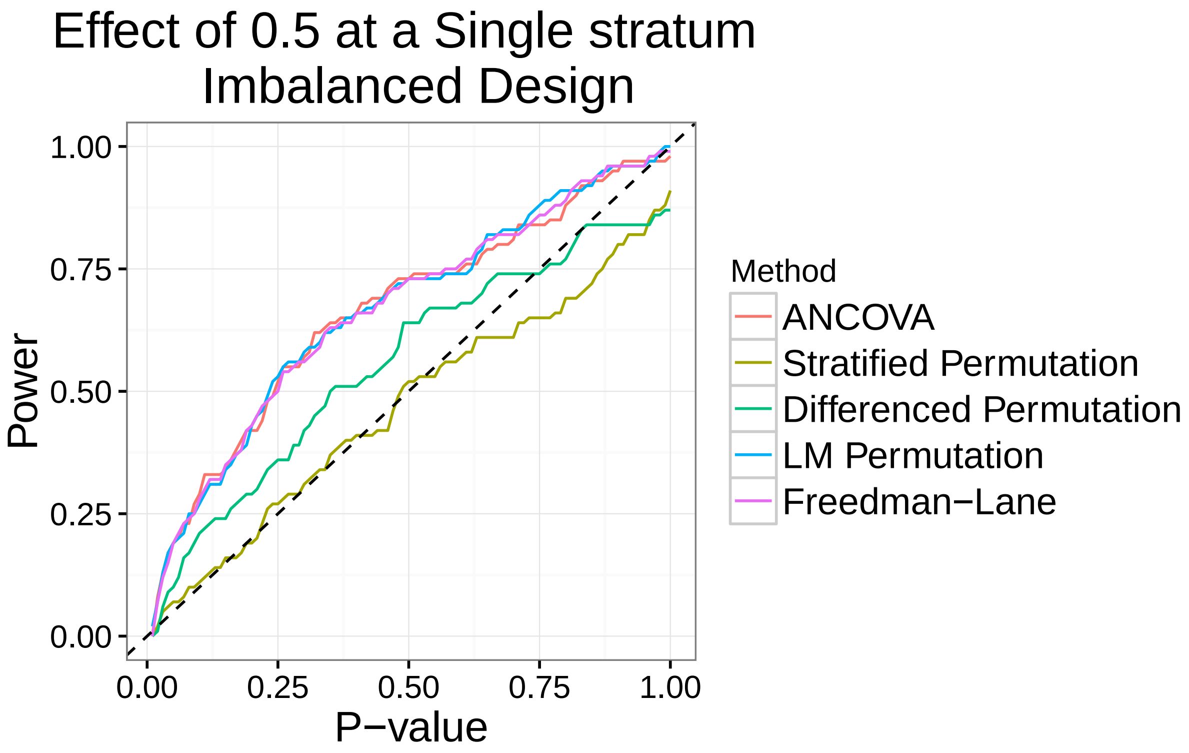
```
plot_power_curves(design4_power, "Constant Additive Effect of 0.5 \n Imbalanced Design")
```



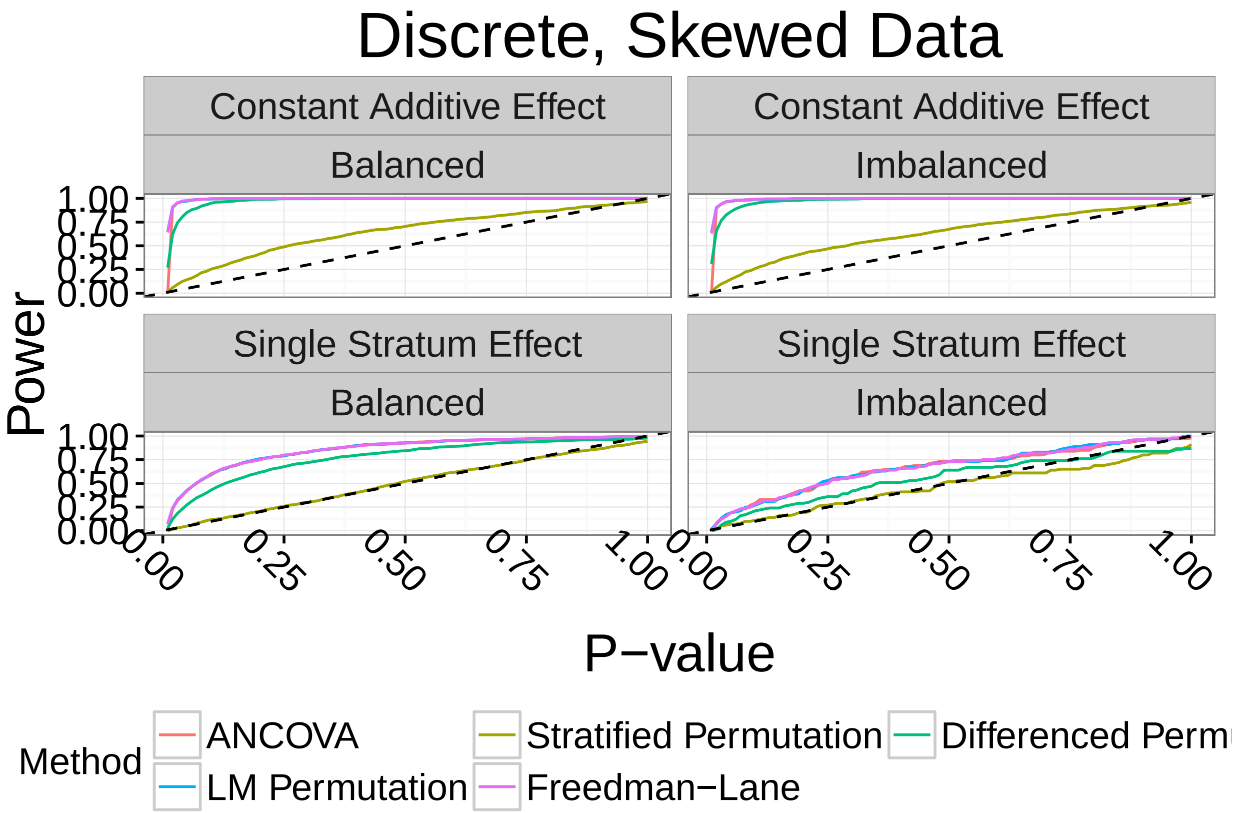
Design 5: Single stratum effect, imbalanced design

```
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC

tmp <- generate_simulated_data(gamma = 0.5, effect = "single stratum effect",
    n = c(8, 16, 24))
design5_pvalues <- replicate(100, {
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$epsilon <- rbinom(nrow(tmp), 1, 0.5) - 0.5
    tmp$Y1 <- generate_outcome(tmp$Y0, tmp$Z * tmp$gamma_vec,
        tmp$stratum_effect, tmp$epsilon)
    generate_simulated_pvalues(tmp)
})
design5_pvalues <- t(design5_pvalues)
colnames(design5_pvalues) <- c("ANCOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design5_power <- apply(design5_pvalues, 2, compute_power)
```

```
plot_power_curves(design5_power, "Effect of 0.5 at a Single stratum \n Imbalanced Design")
```



```
powers <- list(design1_power %>% as.data.frame() %>% mutate(Treatment = rep("Constant Additive Effect",
    nrow(design1_power)), Design = rep("Balanced", nrow(design1_power))),
    design2_power %>% as.data.frame() %>% mutate(Treatment = rep("Single Stratum Effect",
        nrow(design2_power)), Design = rep("Balanced", nrow(design2_power))),
    design4_power %>% as.data.frame() %>% mutate(Treatment = rep("Constant Additive Effect",
        nrow(design4_power)), Design = rep("Imbalanced", nrow(design4_power))),
    design5_power %>% as.data.frame() %>% mutate(Treatment = rep("Single Stratum Effect",
        nrow(design5_power)), Design = rep("Imbalanced", nrow(design5_power))),
    design0_power %>% as.data.frame() %>% mutate(Treatment = rep("No Effect",
        nrow(design0_power)), Design = rep("Balanced", nrow(design0_power))),
    design00_power %>% as.data.frame() %>% mutate(Treatment = rep("No Effect",
        nrow(design00_power)), Design = rep("Imbalanced", nrow(design00_power))))

all_power_curves <- do.call(rbind, powers)
twobytwo <- all_power_curves %>% filter(Treatment != "No Effect") %>%
    melt(id.vars = c("Treatment", "Design")) %>% mutate(pvalue = rep((1:100)/100,
    5 * 4)) %>% mutate(Method = variable) %>% ggplot(aes_string(x = "pvalue",
    y = "value", color = "Method")) + geom_line() + geom_abline(intercept = 0,
    slope = 1, linetype = "dashed") + xlab("P-value") + ylab("Power") +
    facet_wrap(Treatment ~ Design) + ggtitle("Discrete, Skewed Data") +
    report_theme + theme(legend.position = "bottom") + guides(color = guide_legend(nrow = 2,
    byrow = TRUE))

twobytwo
```



```
pdf(file = "../ms/fig/skewed_simulation_power.pdf", width = 8)
twobytwo
dev.off()
```

```
## pdf
##   2
```

```
summary05 <- t(sapply(powers, function(x) x[5, c(6, 7, 1:5)]))
summary05 <- summary05[c(5, 1, 2, 6, 3, 4), ]
summarytab <- xtable(summary05, digits = 3, align = "rp{1.25in}|p{0.7in}|p{0.6in}p{0.8in}p{0.8in}p{0.8i
    caption = "Empirical power at level $0.05$ for discrete, skewed simulated data",
    label = "tab:skewed_power")
print(summarytab, hline.after = c(-1, 0, 3, nrow(summarytab)),
    type = "latex", file = "../ms/fig/skewed_simulation_power_summary.tex")
summarytab
```

Treatment	Design	ANCOVA	Stratified Permutation	Differenced Permutation	LM Permutation	Freedman-Lane
No Effect	Balanced	0.036	0.025	0.021	0.035	0.037
Constant Additive Effect	Balanced	0.974	0.143	0.853	0.972	0.977
Single Stratum Effect	Balanced	0.422	0.050	0.275	0.429	0.424
No Effect	Imbalanced	0.038	0.024	0.019	0.037	0.039
Constant Additive Effect	Imbalanced	0.971	0.147	0.860	0.970	0.972
Single Stratum Effect	Imbalanced	0.190	0.070	0.100	0.190	0.190

Table 1: Empirical power at level 0.05 for discrete, skewed simulated data