ANCOVA Comparison Simulations: Imbalanced Design

Kellie Ottoboni

2017-02-15

Normal data

We assume the following linear data-generating process:

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

for individuals $i = 1, \ldots, n_j$, $j = 1, \ldots, J$. β_0 is the coefficient for the standard normally-distributed baseline measurement Y_{i0}, β_j is the mean effect of being in stratum j, Z_{ij} is the treatment level, γ_j is the effect of treatment in stratum j, and ε_{ij} is an error term. We will assume that $\beta_0 = 1$. The observed $(Y_{ij0}, \varepsilon_{ij})$ are independent across i and j.

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 **imbalanced**: 12 people receive each treatment 0 and 4 people receive treatment 1.

The same simulation guidelines as detailed in the Gaussian simulation file.

Data-generation, tests, and plotting functions

```
generate_simulated_data <- function(gamma, effect, errors, 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?  errors = 'normal' or 'heavy' 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 errors
    if (errors == "normal") {
        epsilon <- rnorm(N)
```

```
    } else {
        epsilon <- rt(N, df = 2)
    }

    # Generate covariates
    Y0 <- rnorm(N)
    Z <- do.call(c, lapply(n, function(x) rep(0:1, times = c(x *
        (3/4), x * (1/4)))))
    stratum_effect <- rep(beta, times = n)
    Y1 <- 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$Z == 1]) -
        mean(dataset$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))
}
```

```
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))
}
```

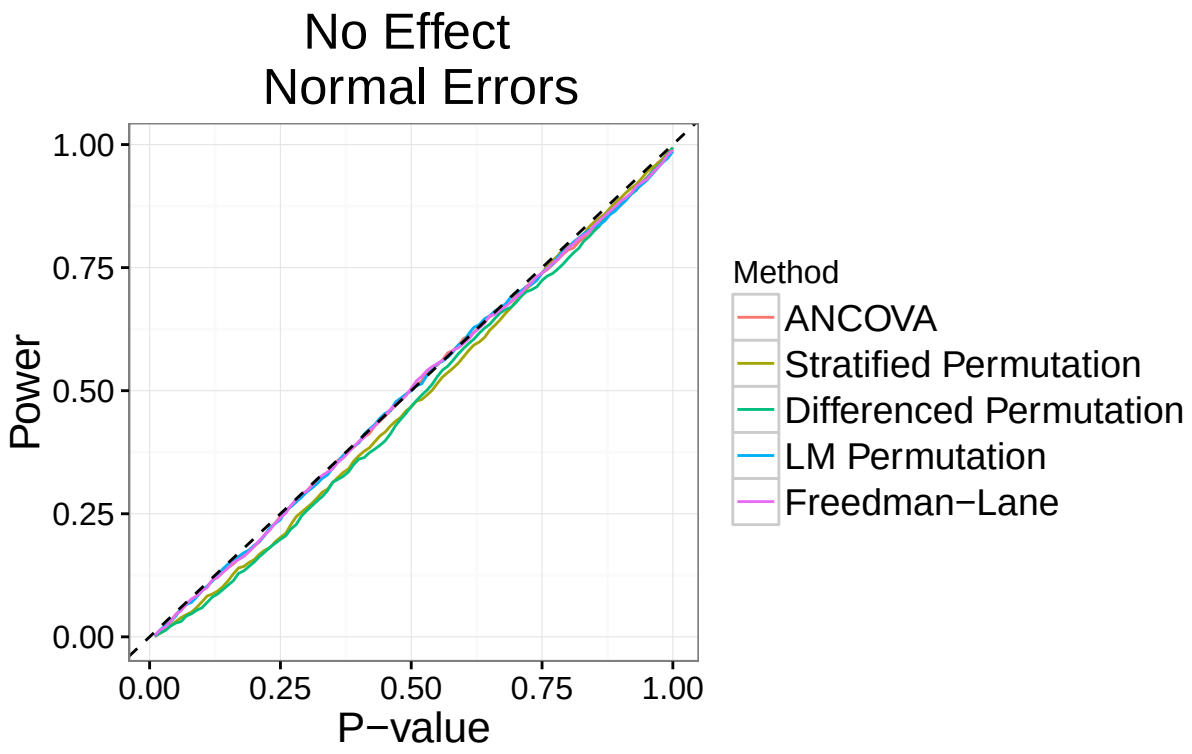
Test level: simulation under the null

The figures below show that the test has the correct level under the null for both error distributions.

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

tmp <- generate_simulated_data(gamma = 0, effect = "same effect",
    errors = "normal")
design0_pvalues <- replicate(1000, {
    tmp$epsilon <- rnorm(nrow(tmp))
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$Y1 <- tmp$Y0 + tmp$stratum_effect + tmp$epsilon
    generate_simulated_pvalues(tmp)
})
design0_pvalues <- t(design0_pvalues)
colnames(design0_pvalues) <- c("ANCOVA", "Stratified Permutation",
```

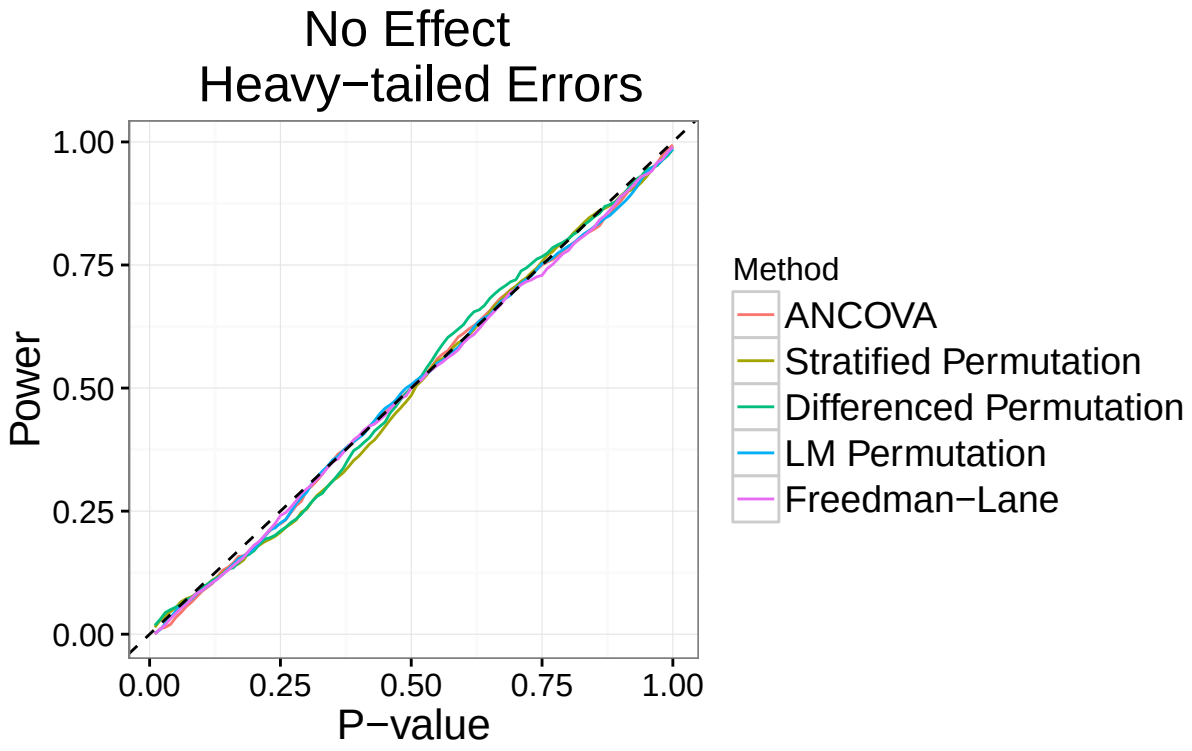
```
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design0_power <- apply(design0_pvalues, 2, compute_power)
```

```
plot_power_curves(design0_power, "No Effect \n Normal Errors")
```




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

tmp <- generate_simulated_data(gamma = 0, effect = "same effect",
    errors = "heavy")
design00_pvalues <- replicate(1000, {
    tmp$epsilon <- rt(nrow(tmp), df = 2)
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$Y1 <- tmp$Y0 + 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)
```

```
plot_power_curves(design00_power, "No Effect \n Heavy-tailed Errors")
```

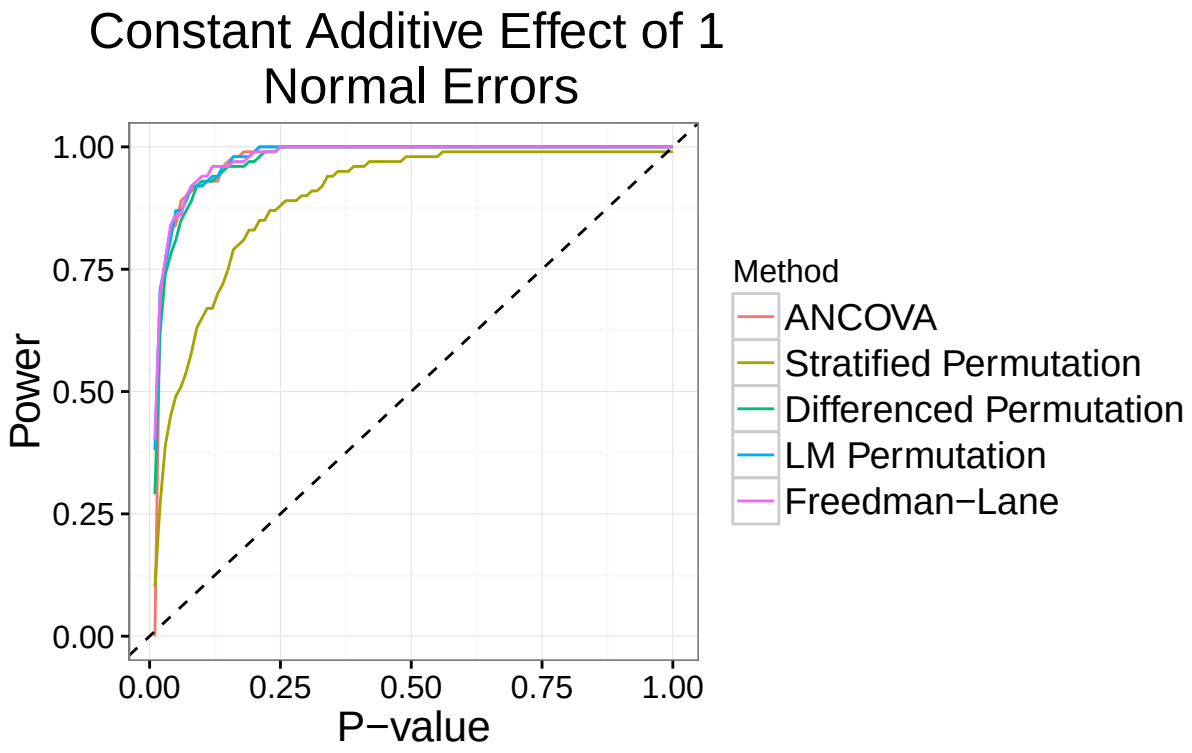


Design 1: Constant additive effect, normal errors

There is no discernable difference in power between the ANCOVA, the differenced stratified permutation test, the LM permutations, or the Freedman-Lane test. Only the simple stratified permutation test of Y_1 has substantially less power than the other four. Without controlling for the baseline values, the variance in Y_1 masks the treatment effect.

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

tmp <- generate_simulated_data(gamma = 1, effect = "same effect",
    errors = "normal")
design1_pvalues <- replicate(100, {
    tmp$epsilon <- rnorm(nrow(tmp))
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$Y1 <- tmp$Y0 + tmp$stratum_effect + tmp$gamma_vec * tmp$Z +
        tmp$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 1 \n Normal Errors")
```

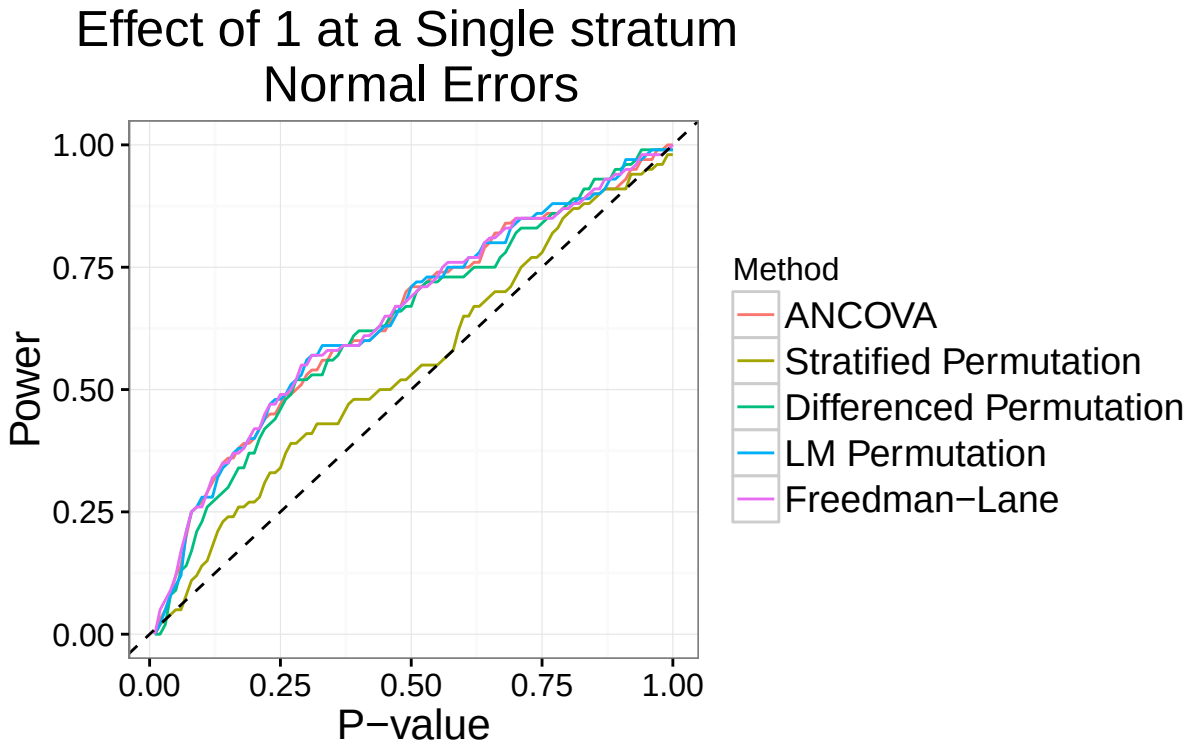


Design 2: Single stratum effect, normal errors

The results here are similar to the Gaussian simulations. We omit comments for the rest: these is simply a loss in power for all tests relative to the balanced design.

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

tmp <- generate_simulated_data(gamma = 1, effect = "single stratum effect",
    errors = "normal")
design2_pvalues <- replicate(100, {
    tmp$epsilon <- rnorm(nrow(tmp))
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$Y1 <- tmp$Y0 + tmp$stratum_effect + tmp$gamma_vec * tmp$Z +
        tmp$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)
```

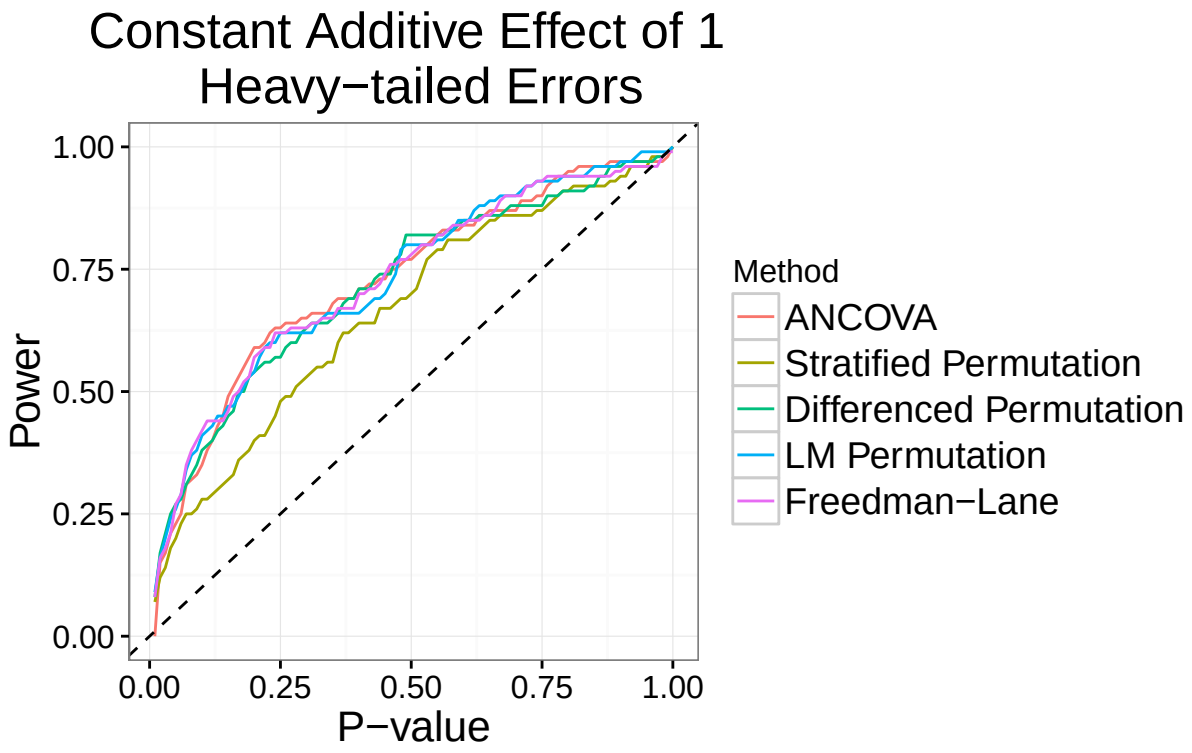
```
plot_power_curves(design2_power, "Effect of 1 at a Single stratum \n Normal Errors")
```



Design 3: Constant additive effect, heavy-tailed errors

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

tmp <- generate_simulated_data(gamma = 1, effect = "same effect",
    errors = "heavy")
design3_pvalues <- replicate(100, {
    tmp$epsilon <- rt(nrow(tmp), df = 2)
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$Y1 <- tmp$Y0 + tmp$stratum_effect + tmp$gamma_vec * tmp$Z +
        tmp$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)
```

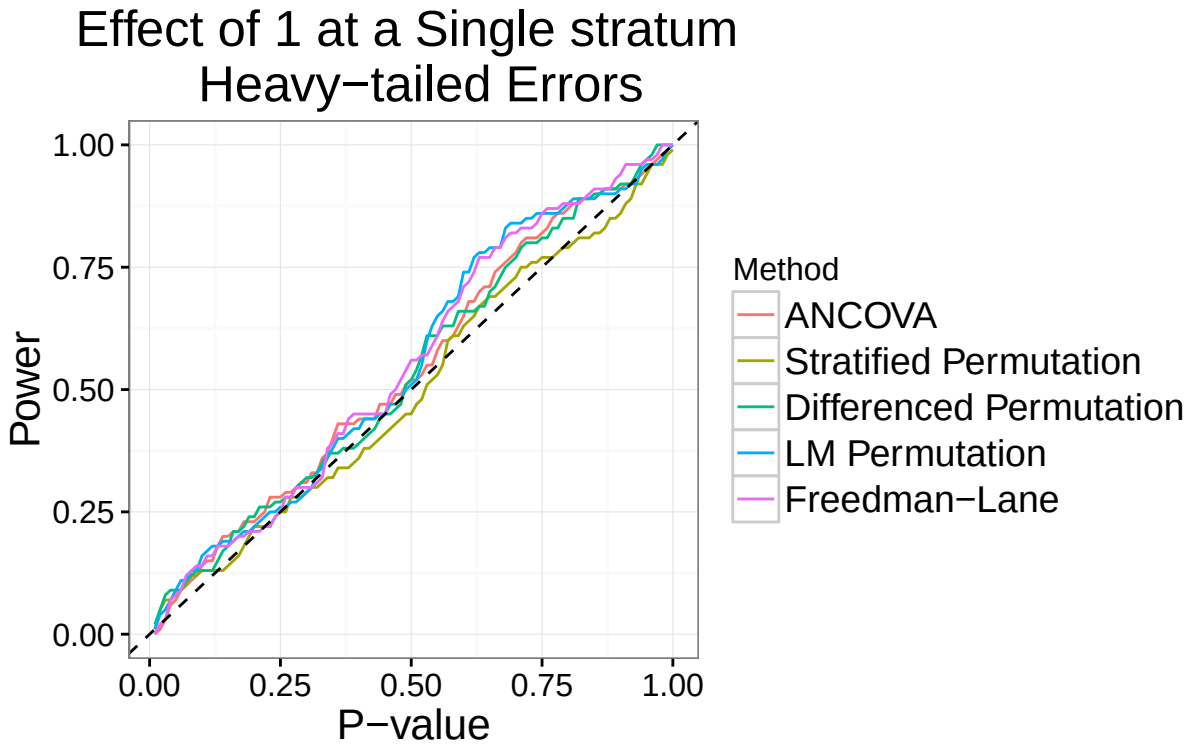
```
plot_power_curves(design3_power, "Constant Additive Effect of 1 \n Heavy-tailed Errors")
```



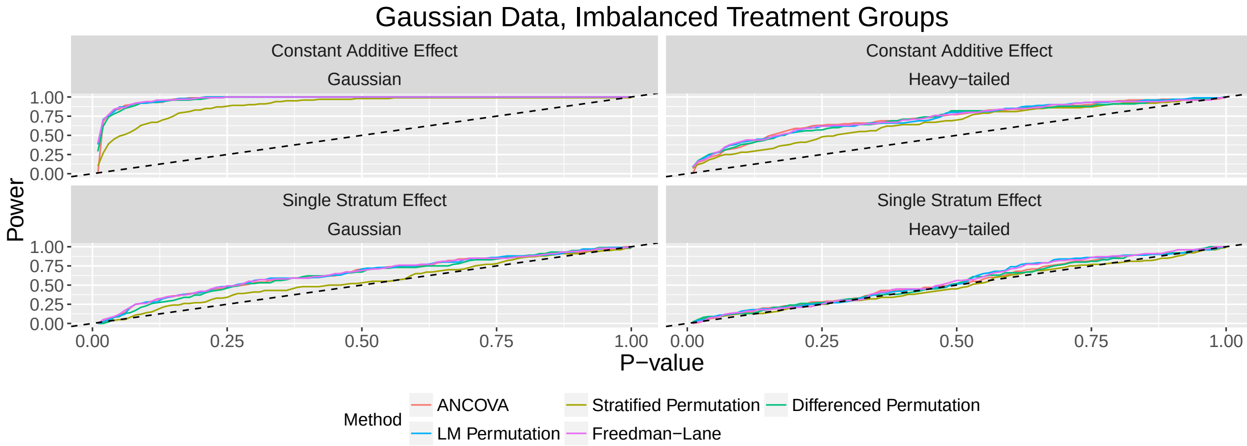
Design 4: single stratum effect, heavy-tailed errors

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

tmp <- generate_simulated_data(gamma = 1, effect = "single stratum effect",
    errors = "heavy")
design4_pvalues <- replicate(100, {
    tmp$epsilon <- rt(nrow(tmp), df = 2)
    tmp$Z <- permute_within_groups(tmp$Z, tmp$stratumID)
    tmp$Y1 <- tmp$Y0 + tmp$stratum_effect + tmp$gamma_vec * tmp$Z +
        tmp$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, "Effect of 1 at a Single stratum \n Heavy-tailed Errors")
```




```
powers <- list(design1_power %>% as.data.frame() %>% mutate(Treatment = rep("Constant Additive Effect",
    nrow(design1_power)), Errors = rep("Gaussian", nrow(design1_power))),
    design2_power %>% as.data.frame() %>% mutate(Treatment = rep("Single Stratum Effect",
        nrow(design2_power)), Errors = rep("Gaussian", nrow(design2_power))),
    design3_power %>% as.data.frame() %>% mutate(Treatment = rep("Constant Additive Effect",
        nrow(design3_power)), Errors = rep("Heavy-tailed", nrow(design3_power))),
    design4_power %>% as.data.frame() %>% mutate(Treatment = rep("Single Stratum Effect",
        nrow(design4_power)), Errors = rep("Heavy-tailed", nrow(design4_power))),
    design0_power %>% as.data.frame() %>% mutate(Treatment = rep("No Effect",
        nrow(design0_power)), Errors = rep("Gaussian", nrow(design0_power))),
    design00_power %>% as.data.frame() %>% mutate(Treatment = rep("No Effect",
        nrow(design00_power)), Errors = rep("Heavy-tailed", nrow(design00_power))))

all_power_curves <- do.call(rbind, powers)
twobytwo <- all_power_curves %>% filter(Treatment != "No Effect") %>%
    melt(id.vars = c("Treatment", "Errors")) %>% 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 ~ Errors) + ggtitle("Gaussian Data, Imbalanced Treatment Groups") +
    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 = 12),
        strip.text.x = element_text(size = 12), legend.position = "bottom") +
    guides(color = guide_legend(nrow = 2, byrow = TRUE))
twobytwo
```




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

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

```r
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 = "p{1.25in}|p{0.7in}|p{0.6in}p{0.8in}p{0.8in}p{0.8in
    caption = "Empirical power at level $0.05$ for Gaussian simulated data with imbalanced treatment gro
    label = "tab:imbalanced_power", hline.after = c(-1, 0, 3))
print(summarytab, hline.after = c(-1, 0, nrow(summarytab)),
    type = "latex", file = "../ms/fig/imbalanced_simulation_power_summary.tex")
summarytab
```

Treatment	Errors	ANCOVA	Stratified Permutation	Differenced Permutation	LM Permutation	Freedman-Lane
No Effect	Gaussian	0.046	0.031	0.028	0.042	0.045
Constant Additive Effect	Gaussian	0.840	0.490	0.810	0.870	0.860
Single Stratum Effect	Gaussian	0.120	0.050	0.090	0.100	0.120
No Effect	Heavy-tailed	0.034	0.054	0.055	0.047	0.042
Constant Additive Effect	Heavy-tailed	0.230	0.200	0.270	0.260	0.270
Single Stratum Effect	Heavy-tailed	0.070	0.080	0.090	0.090	0.080

Table 1: Empirical power at level 0.05 for Gaussian simulated data with imbalanced treatment groups