ANOVA Comparison Simulations

Kellie Ottoboni

2016-11-29

The model

We assume the following linear data-generating process:

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

for individuals $i = 1, \dots, n_j$, $j = 1, \dots, J$. β_0 is the coefficient for the baseline measurement Y_{i0}, β_j is the mean effect of being at site j, Z_{ij} is the treatment level, γ_j is the effect of treatment at site j, and ε_{ij} is an error term. We will assume that $\beta_0 = 1$.

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

We use two designs:

- Design 1: $\gamma_1 = \gamma_2 = \gamma_3 = 1$. This is the standard assumption of a constant, additive treatment effect.
- Design 2: $\gamma_1 = \gamma > 0$, $\gamma_2 = \gamma_3 = 0$. This is a constant, additive treatment effect at site 1, but no treatment effect at sites 2 and 3. This is a simplistic case of heterogeneous treatment effects.

With these two designs, we vary the distribution of ε. In the first case, we use $\varepsilon \sim N(0, 1)$ to mimic the usual ANOVA assumptions. In the second case, $\varepsilon \sim t(2)$ so the errors are heavy-tailed. Thus, there are four total simulation designs.

We compare five tests:

- ANOVA: we fit a linear model of response Y_1 on baseline Y_0, treatment Z, and a dummy for site.
- Stratified permutation: we permute treatment assignment within site, 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 site dummies
- Freedman-Lane test: see the other Rmd document for a full description of this procedure

Throughout our simulations, we first fix Y_0 and site ID. Treatment Z and the errors ε are randomly drawn according to their respective distributions. Then, Y_1 is constructed using the linear data-generating process above. We regenerate Z, ε, and Y_1 100 times for each design, then compute the empirical power of the five tests.

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 site effect' - which sites have
    # a tr effect > 0?  errors = 'normal' or 'heavy' Returns: a
    # dataframe containing columns named Y1 (response), Y0
    # (baseline), Z (treatment), and SITEID (stratum)

    SITEID <- 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 <- rep(0:1, length.out = N)
    site_effect <- rep(beta, times = n)
    Y1 <- Y0 + gamma_vec * Z + site_effect + epsilon
    return(data.frame(Y1, Y0, Z, gamma_vec, SITEID, site_effect,
        epsilon))
}

generate_simulated_pvalues <- function(dataset, reps = 1000) {
    # Inputs: dataset = a dataframe containing columns named Y1
    # (response), Y0 (baseline), Z (treatment), and SITEID
    # (stratum) Returns: a vector of p-values first element is
    # the p-value from the ANOVA 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

    # ANOVA
    modelfit <- lm(Y1 ~ Y0 + Z + factor(SITEID), 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$SITEID, 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$diff, stratum = dataset$SITEID, 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$SITEID)
        lm1_perm <- lm(Y1 ~ Y0 + Z_perm + factor(SITEID), 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(SITEID), 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$SITEID)
        dataset$response_fl <- lm2_yhat + lm2_resid_perm
        lm2_perm <- lm(response_fl ~ Y0 + Z + factor(SITEID),
            data = dataset)
        summary(lm2_perm)[["coefficients"]]["Z", "t value"]
    })
    fl_pvalue <- t2p(observed_t1, lm2_t_distr, alternative = "two-sided")

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

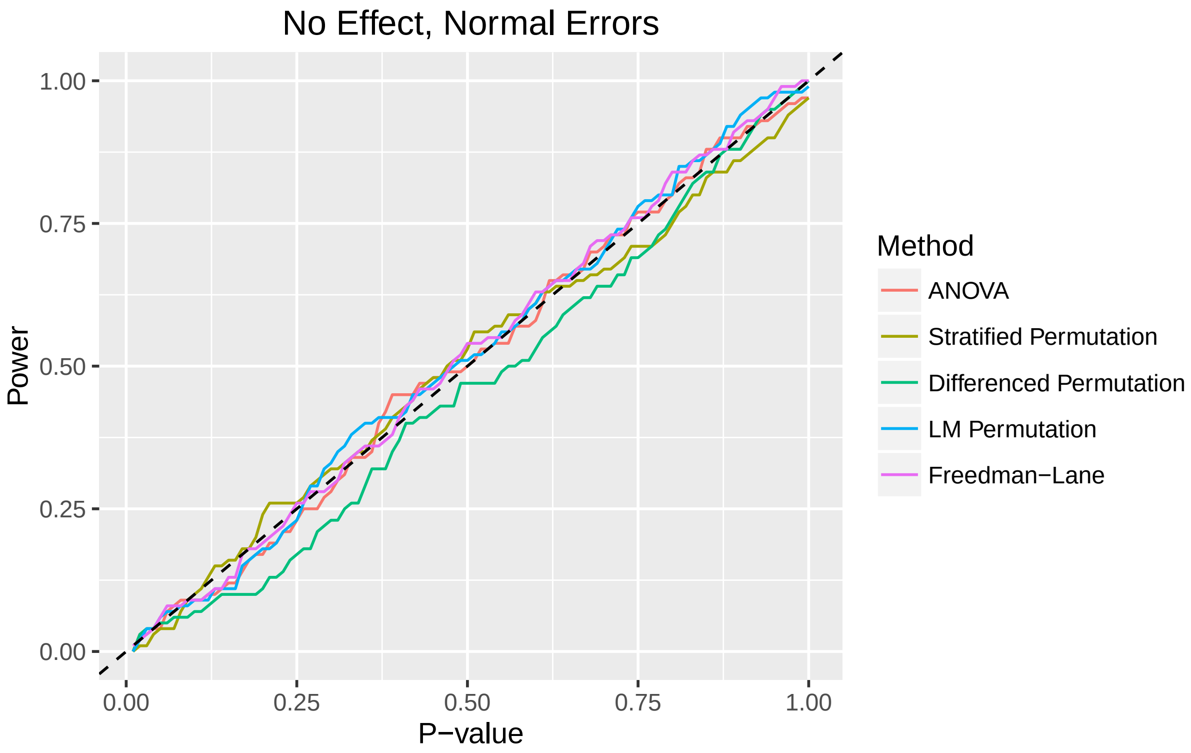
```
compute_power <- function(pvalues) {
    sapply((0:99)/100, function(p) mean(pvalues <= p, na.rm = TRUE))
}

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)
}

plot_pvalue_hist <- function(pvalue_mat, title) {
    melt(pvalue_mat) %>% mutate(Method = Var2) %>% ggplot(aes(x = value,
        fill = Method)) + geom_histogram() + facet_wrap(~Method) +
        ggtitle(title)
}

plot_pvalue_scatter <- function(pvalue_mat, title) {
    pvalue_mat %>% as.data.frame() %>% select(ANOVA, strat = starts_with("Stratified")) %>%
        ggplot(aes(x = ANOVA, y = strat)) + geom_point() + xlim(0,
        1) + ylim(0, 1) + ylab("Stratified Permutation") + geom_abline(intercept = 0,
        slope = 1, linetype = "dashed") + ggtitle(title)
}
```

Test level: simulation under the null

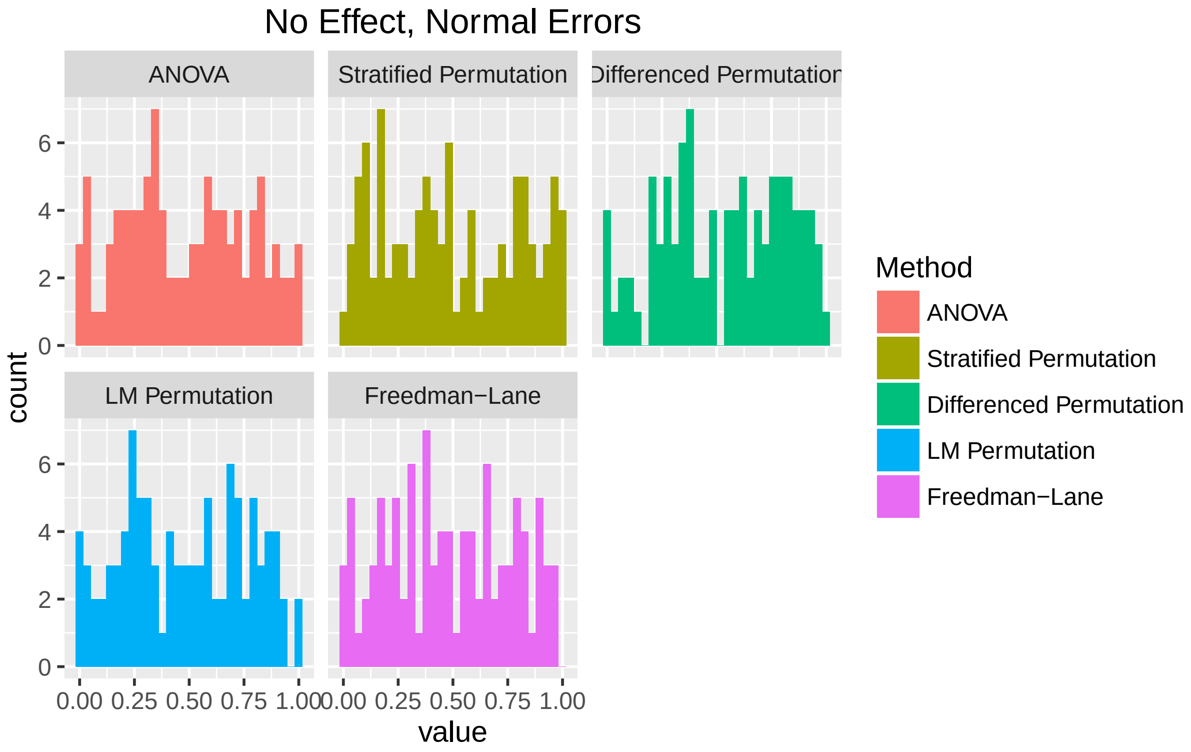
Before testing for different kinds of effects, we begin checking that the tests have the correct level. We follow the procedure described above, using an effect size of $\gamma = 0$ at all sites and using standard normal errors. To have the correct level means that the test rate of rejection at level α is $\alpha 100\%$. In other words, the p-values are uniformly distributed and the power curve should coincide with the line with slope 1 through the origin. Figure **??** demonstrates that this is the case. If anything, the differenced stratified permutation test has fewer than $\alpha 100\%$ false positives when using level α.

```
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(100, {
    tmp$epsilon <- rnorm(nrow(tmp))
    tmp$Z <- permute_within_groups(tmp$Z, tmp$SITEID)
    tmp$Y1 <- tmp$Y0 + tmp$site_effect + tmp$epsilon
    generate_simulated_pvalues(tmp)
})
design0_pvalues <- t(design0_pvalues)
colnames(design0_pvalues) <- c("ANOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design0_power <- apply(design0_pvalues, 2, compute_power)
```

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



```
plot_pvalue_hist(design0_pvalues, "No Effect, Normal Errors")
```

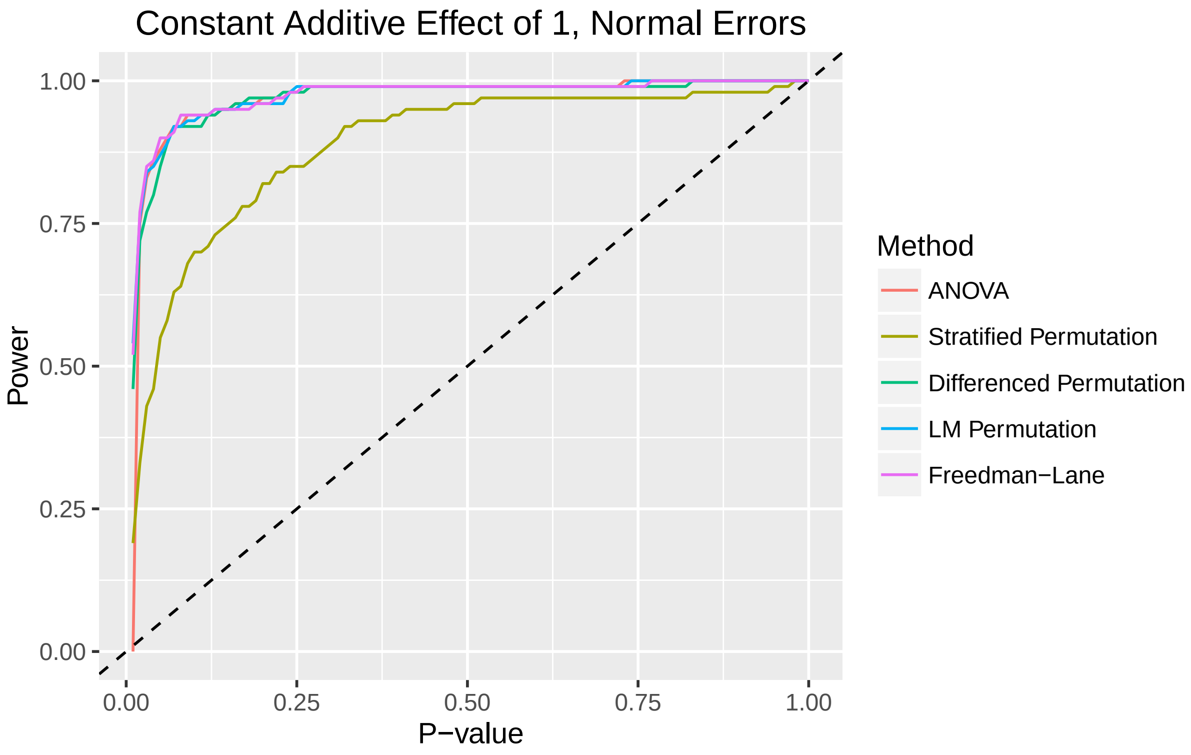


Design 1: Constant additive effect, normal errors

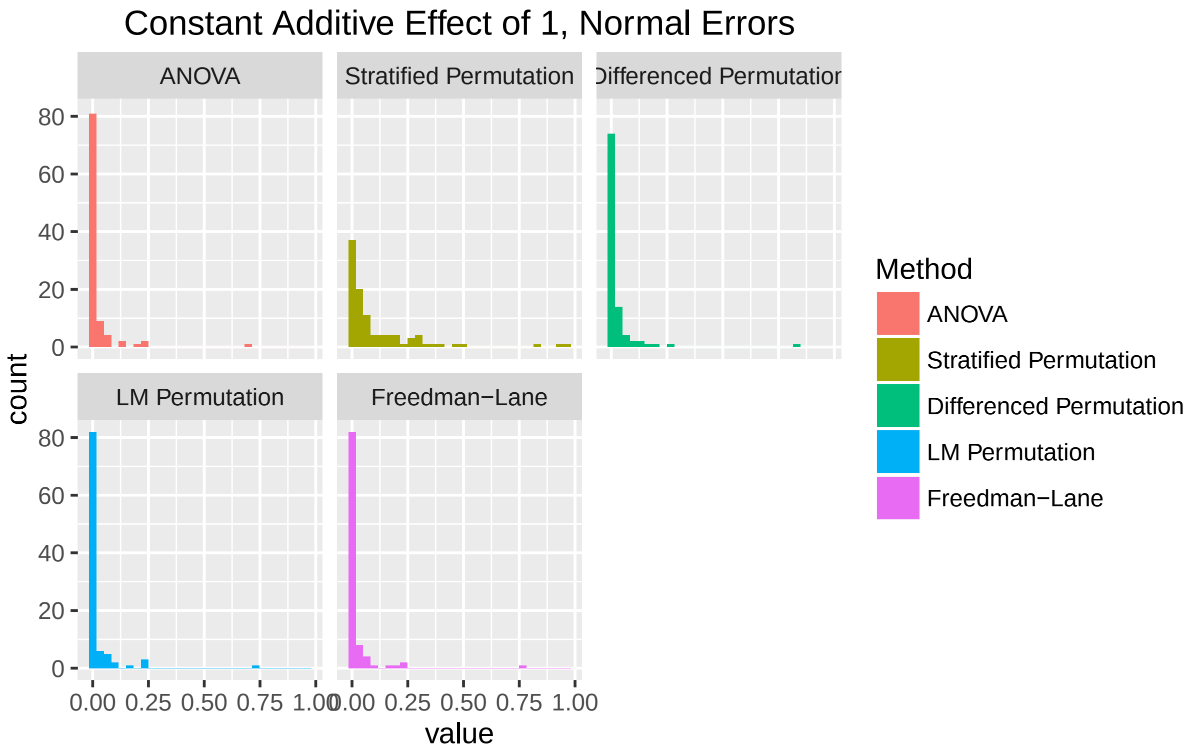
There is no discernable difference in power between the ANOVA, the differenced stratified permutation test, the LM permutations, or the Freedman-Lane test. However, the simple stratified permutation test of Y_1 has substantially less power than the other four. Without controlling for the baseline values,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

```r
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$Z, tmp$SITEID)
    tmp$Y1 <- tmp$Y0 + tmp$site_effect + tmp$gamma_vec * tmp$Z +
        tmp$epsilon
    generate_simulated_pvalues(tmp)
})

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

```r
plot_power_curves(design1_power, "Constant Additive Effect of 1, Normal Errors")
```



```r
plot_pvalue_hist(design1_pvalues, "Constant Additive Effect of 1, Normal Errors")
```

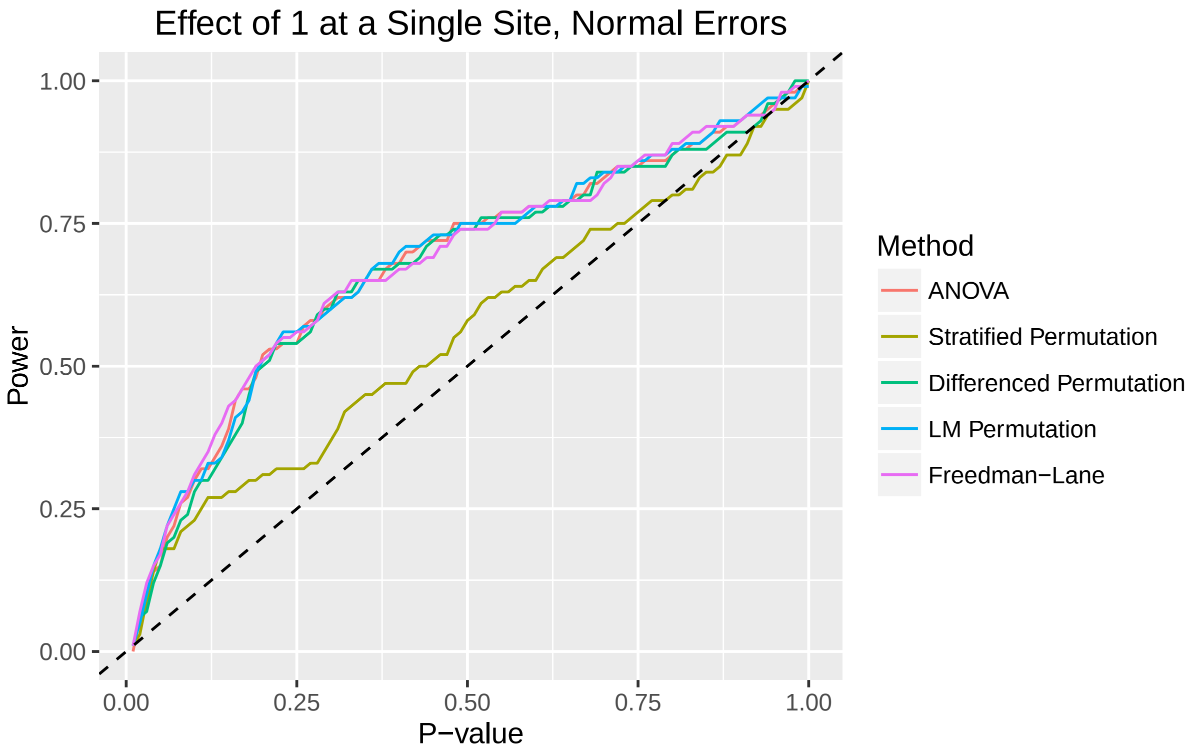


Design 2: Single site effect, normal errors

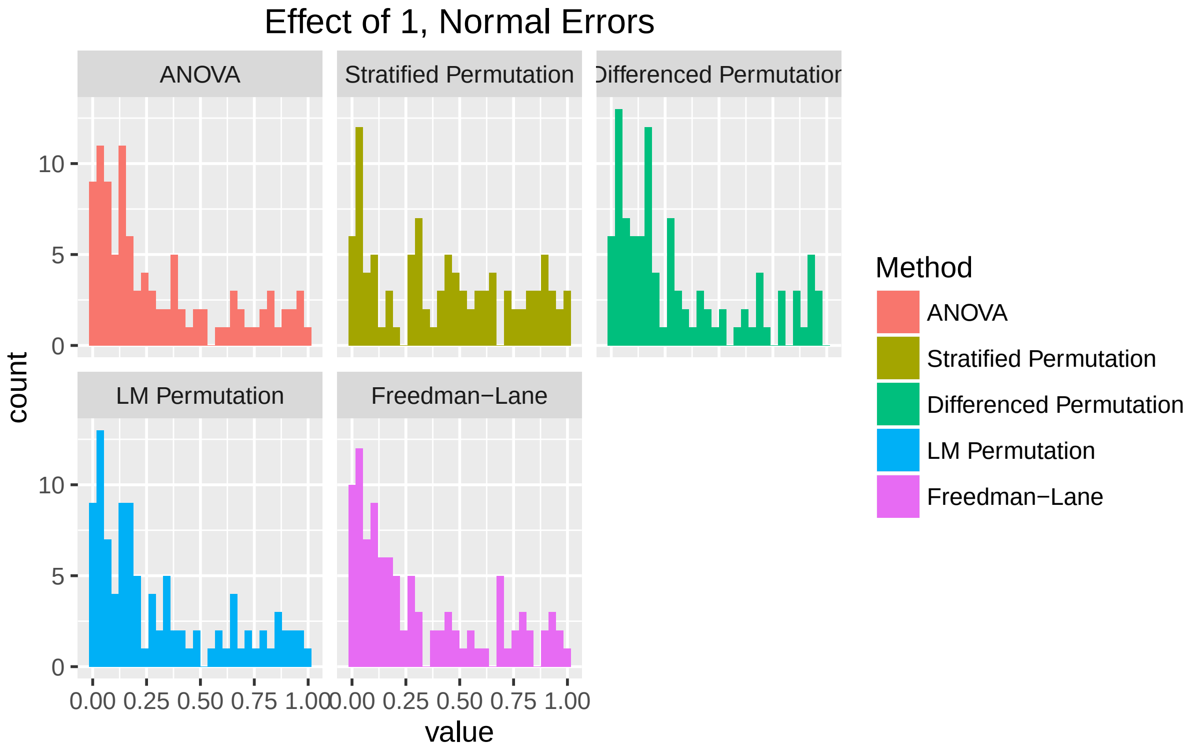
A similar pattern emerges: the simple stratified permutation test has low power, while the other four power curves roughly coincide. The Freedman-Lane test may have the highest power for small p-values, but this could also just be noise. All five power curves are closer to the line passing through the origin: since the effect is only present at one site, it is more difficult to detect.

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

tmp <- generate_simulated_data(gamma = 1, effect = "single site effect",
    errors = "normal")
design2_pvalues <- replicate(100, {
    tmp$epsilon <- rnorm(nrow(tmp))
    tmp$Z <- permute_within_groups(tmp$Z, tmp$SITEID)
    tmp$Y1 <- tmp$Y0 + tmp$site_effect + tmp$gamma_vec * tmp$Z +
        tmp$epsilon
    generate_simulated_pvalues(tmp)
})
design2_pvalues <- t(design2_pvalues)
colnames(design2_pvalues) <- c("ANOVA", "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 Site, Normal Errors")
```



```
plot_pvalue_hist(design2_pvalues, "Effect of 1, Normal Errors")
```

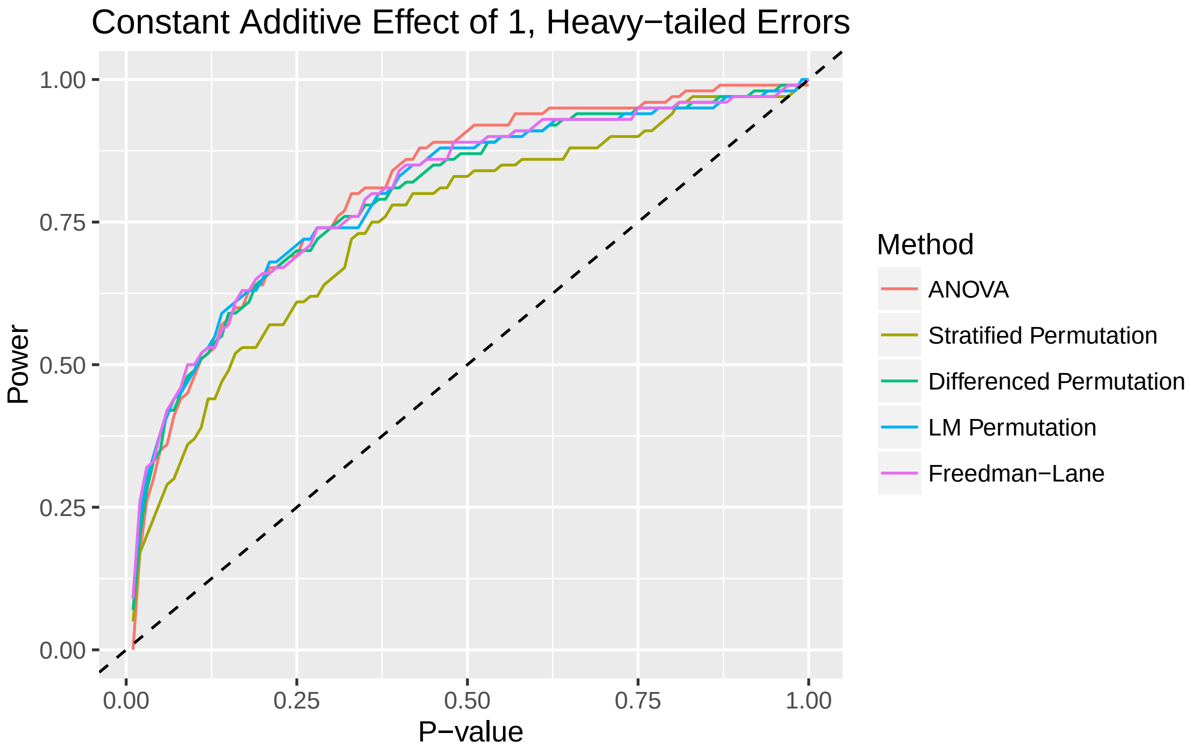


Design 3: Constant additive effect, heavy-tailed errors

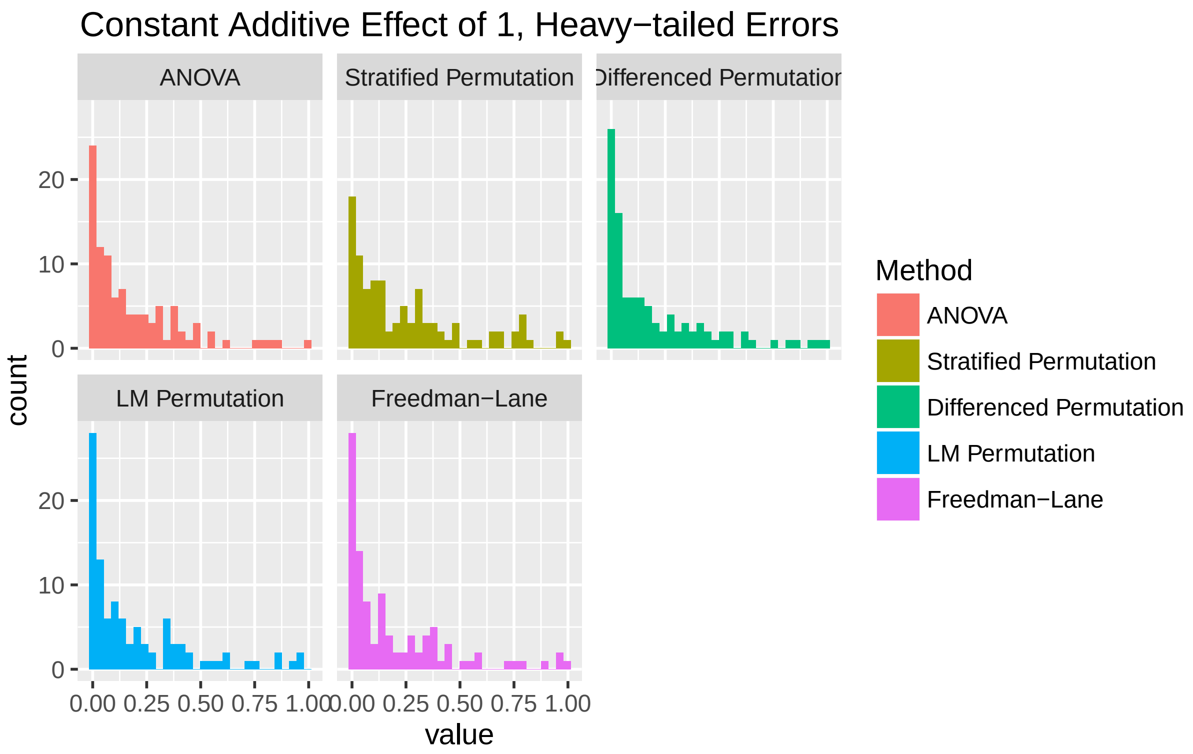
Again, we find that the four power curves coincide while the curve for the stratified permutation test is below the others. Here, the difference between the curves is not large. Controlling for baseline Y_0 does not substantially help reduce variance.

```r
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$Z, tmp$SITEID)
    tmp$Y1 <- tmp$Y0 + tmp$site_effect + tmp$gamma_vec * tmp$Z +
        tmp$epsilon
    generate_simulated_pvalues(tmp)
})
design3_pvalues <- t(design3_pvalues)
colnames(design3_pvalues) <- c("ANOVA", "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 1, Heavy-tailed Errors")
```



```r
plot_pvalue_hist(design3_pvalues, "Constant Additive Effect of 1, Heavy-tailed Errors")
```

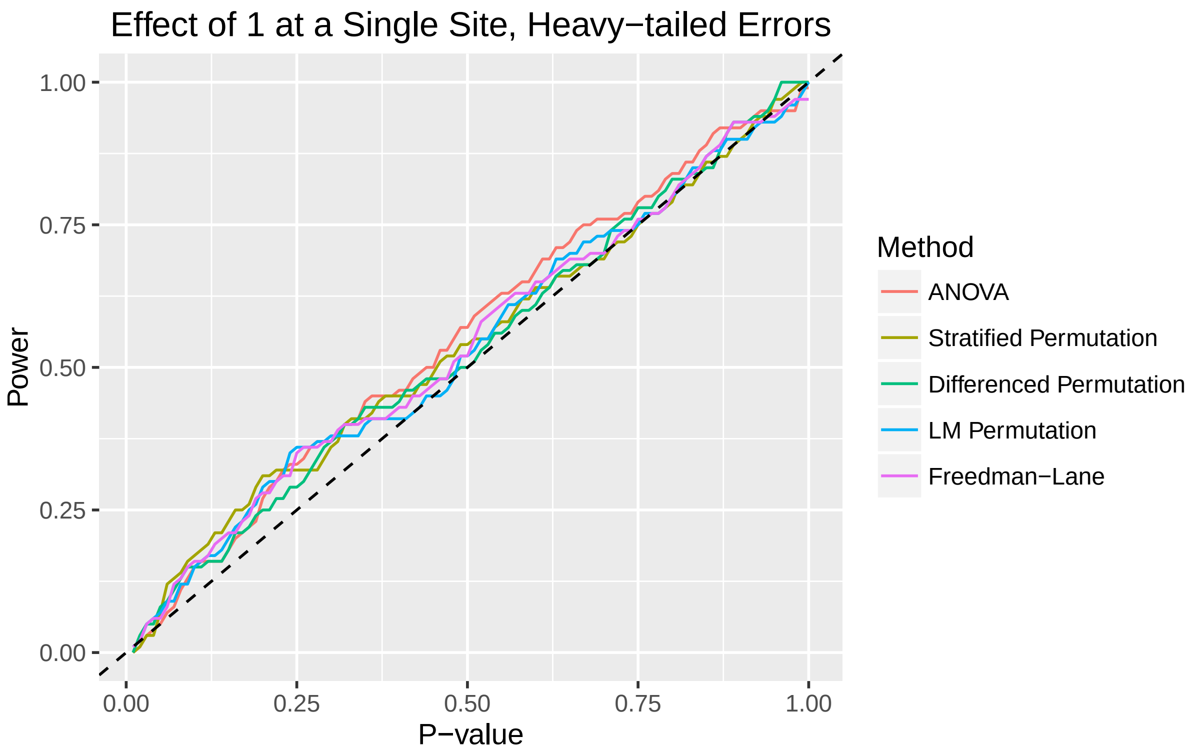


Design 4: single site effect, heavy-tailed errors

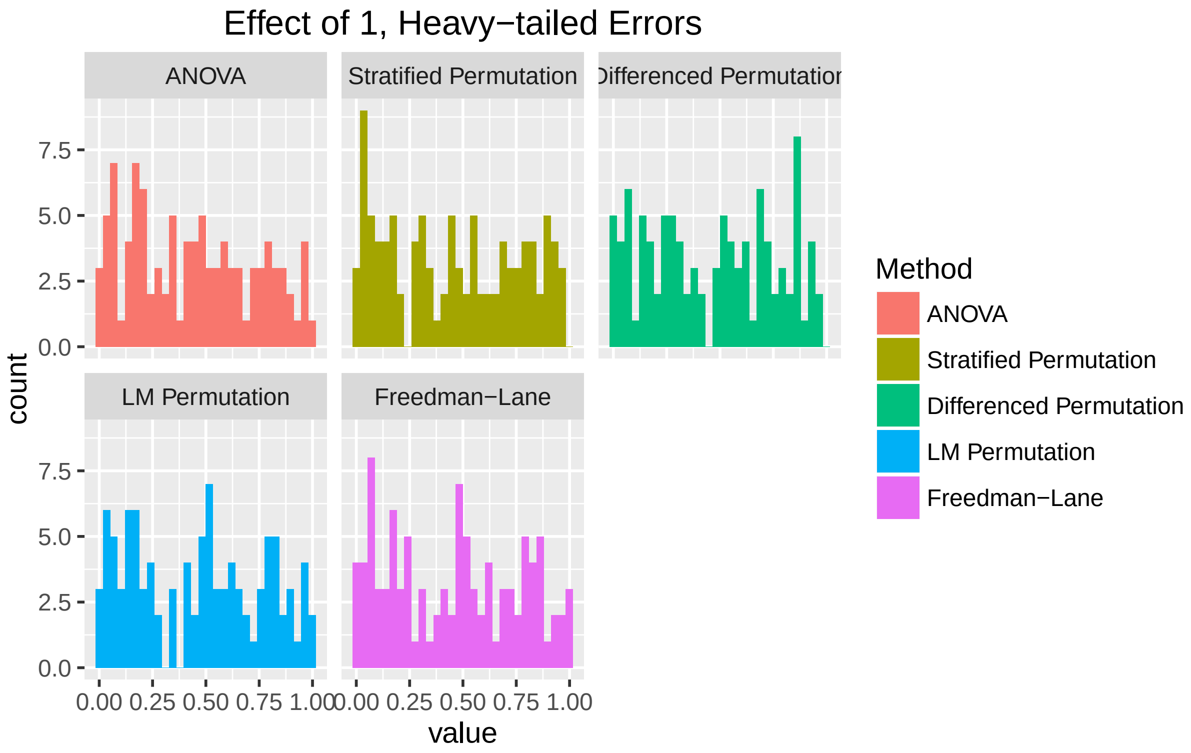
The noise from the heavy-tailed errors masks the treatment effect so much that controlling for baseline measures makes no difference. There is almost no power to detect an effect using any of the five tests.

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

tmp <- generate_simulated_data(gamma = 1, effect = "single site effect",
    errors = "heavy")
design4_pvalues <- replicate(100, {
    tmp$epsilon <- rt(nrow(tmp), df = 2)
    tmp$Z <- permute_within_groups(tmp$Z, tmp$SITEID)
    tmp$Y1 <- tmp$Y0 + tmp$site_effect + tmp$gamma_vec * tmp$Z +
        tmp$epsilon
    generate_simulated_pvalues(tmp)
})
design4_pvalues <- t(design4_pvalues)
colnames(design4_pvalues) <- c("ANOVA", "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 Site, Heavy-tailed Errors")
```



```
plot_pvalue_hist(design4_pvalues, "Effect of 1, Heavy-tailed Errors")
```



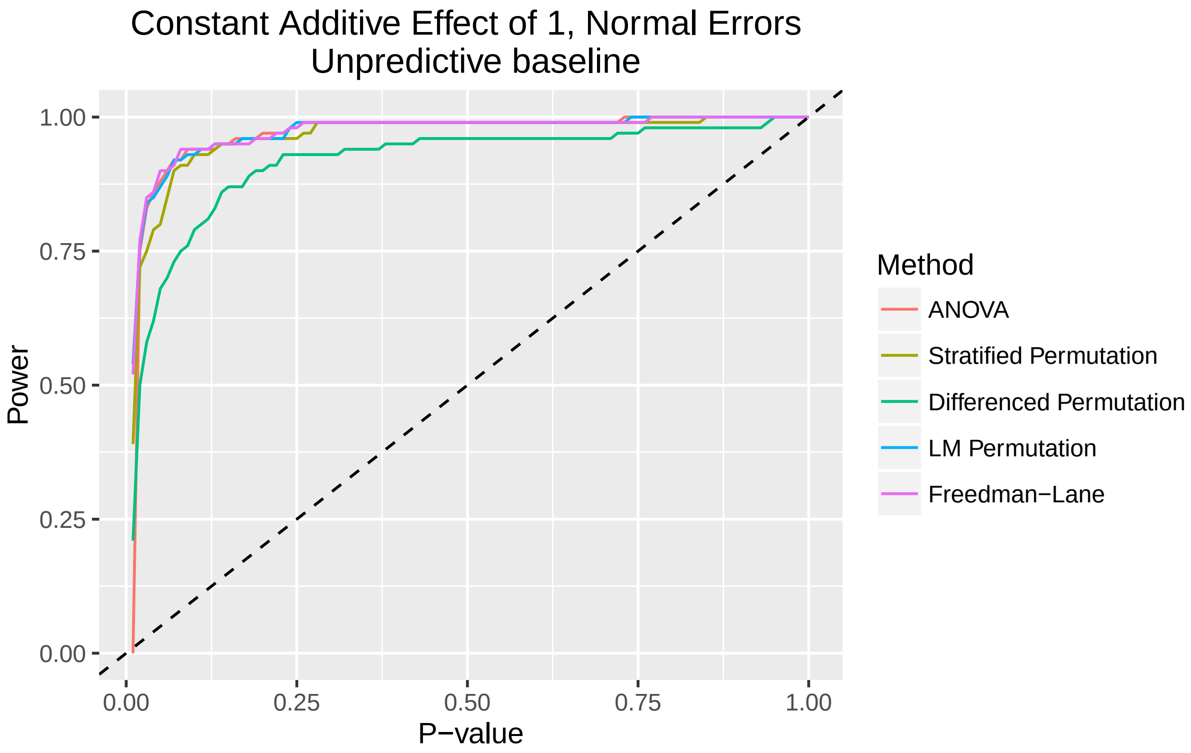
Design 5: Unpredictive baseline

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.

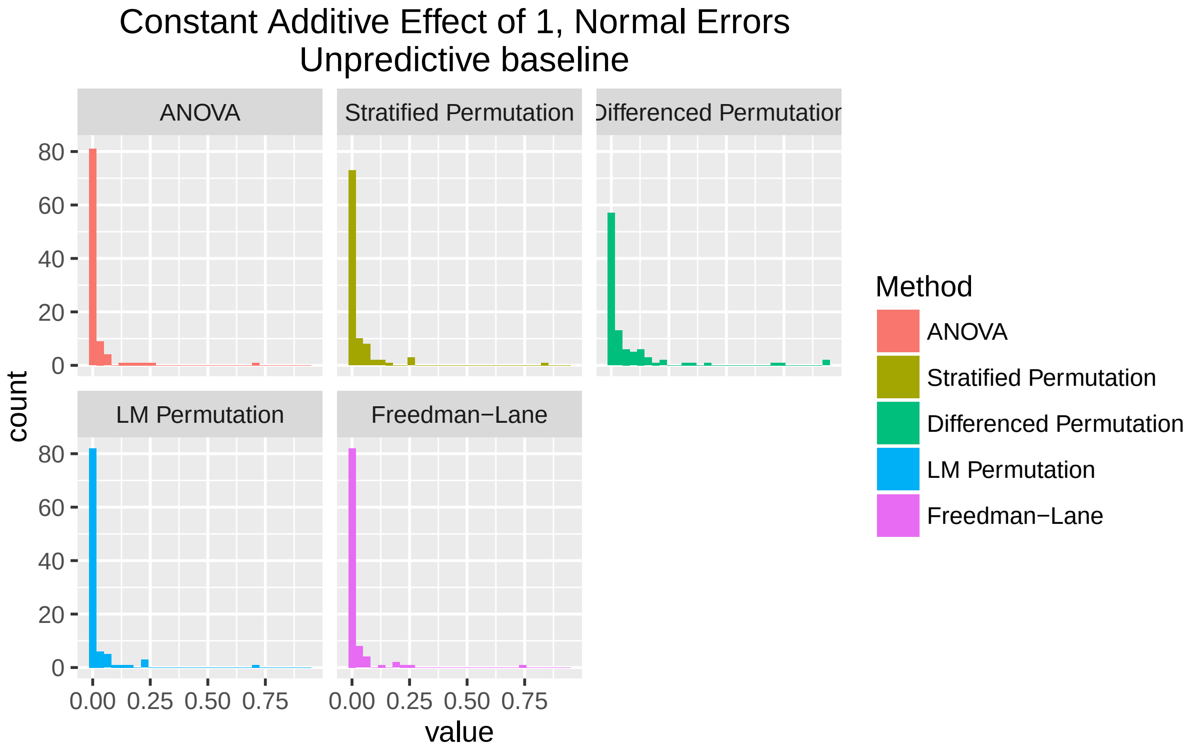
The power curves for this design look similar to those from Design 1 with one important difference: the stratified permutation test has power very close to ANOVA and the two linear model tests, while the differenced permutation test has lower power. This suggests that one should be careful when incorporating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

```r
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")
design5_pvalues <- replicate(100, {
    tmp$epsilon <- rnorm(nrow(tmp))
    tmp$Z <- permute_within_groups(tmp$Z, tmp$SITEID)
    tmp$Y1 <- 0.25 * tmp$Y0 + tmp$site_effect + tmp$gamma_vec *
        tmp$Z + tmp$epsilon
    generate_simulated_pvalues(tmp)
})

design5_pvalues <- t(design5_pvalues)
colnames(design5_pvalues) <- c("ANOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design5_power <- apply(design5_pvalues, 2, compute_power)
```

```r
plot_power_curves(design5_power, "Constant Additive Effect of 1, Normal Errors \n Unpredictive baseline
```



```r
plot_pvalue_hist(design5_pvalues, "Constant Additive Effect of 1, Normal Errors \n Unpredictive baselin
```



Imbalanced designs

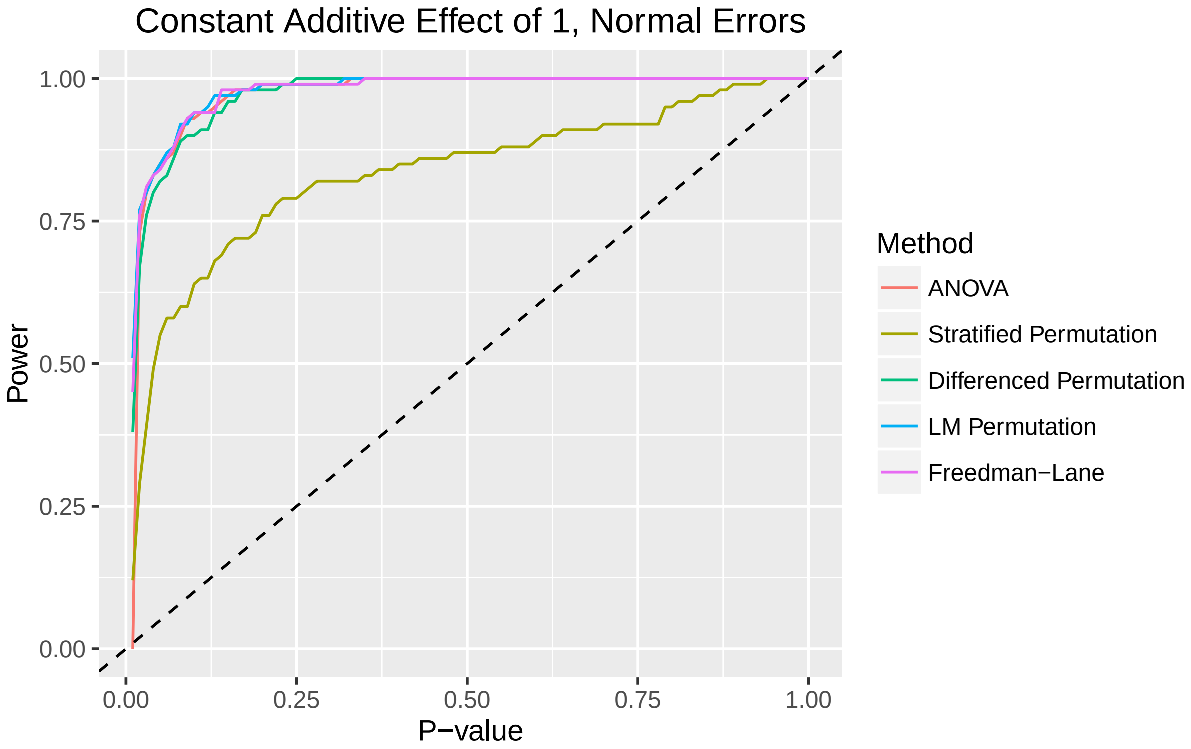
Suppose that instead of having 16 individuals per site, we distribute them unequally across sites: site 1 has 8 patients, site 2 has 16 patients, and site 3 has 24 patients. This imbalance does not violate any assumptions of the ANOVA. However, it may reduce power if the effect is concentrated in sites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

Design 6: Constant additive effect, normal errors

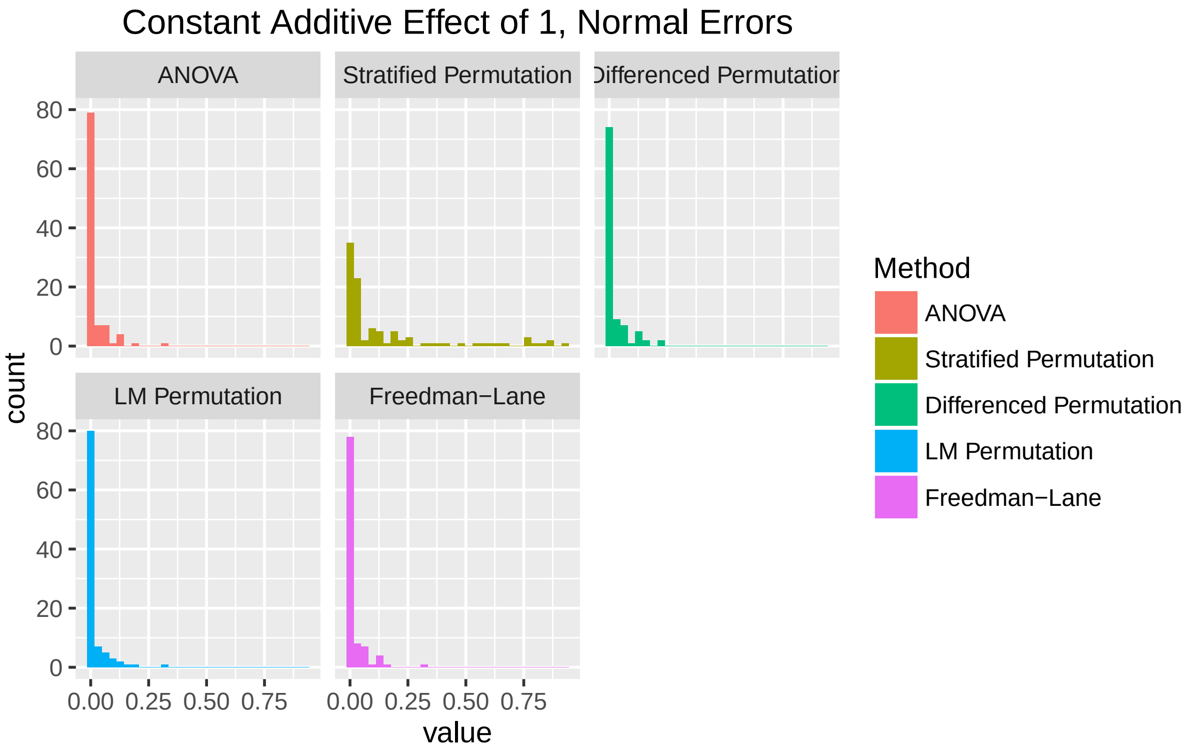
The power curves here are very similar to those in Design 1. This result is expected, since the effect is still present among all patients who received treatment.

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

tmp <- generate_simulated_data(gamma = 1, effect = "same effect",
    errors = "normal", n = c(8, 16, 24))
design6_pvalues <- replicate(100, {
    tmp$epsilon <- rnorm(nrow(tmp))
    tmp$Z <- permute_within_groups(tmp$Z, tmp$SITEID)
    tmp$Y1 <- tmp$Y0 + tmp$site_effect + tmp$gamma_vec * tmp$Z +
        tmp$epsilon
    generate_simulated_pvalues(tmp)
})

design6_pvalues <- t(design6_pvalues)
colnames(design6_pvalues) <- c("ANOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design6_power <- apply(design6_pvalues, 2, compute_power)
```

```
plot_power_curves(design6_power, "Constant Additive Effect of 1, Normal Errors")
```



```
plot_pvalue_hist(design6_pvalues, "Constant Additive Effect of 1, Normal Errors")
```

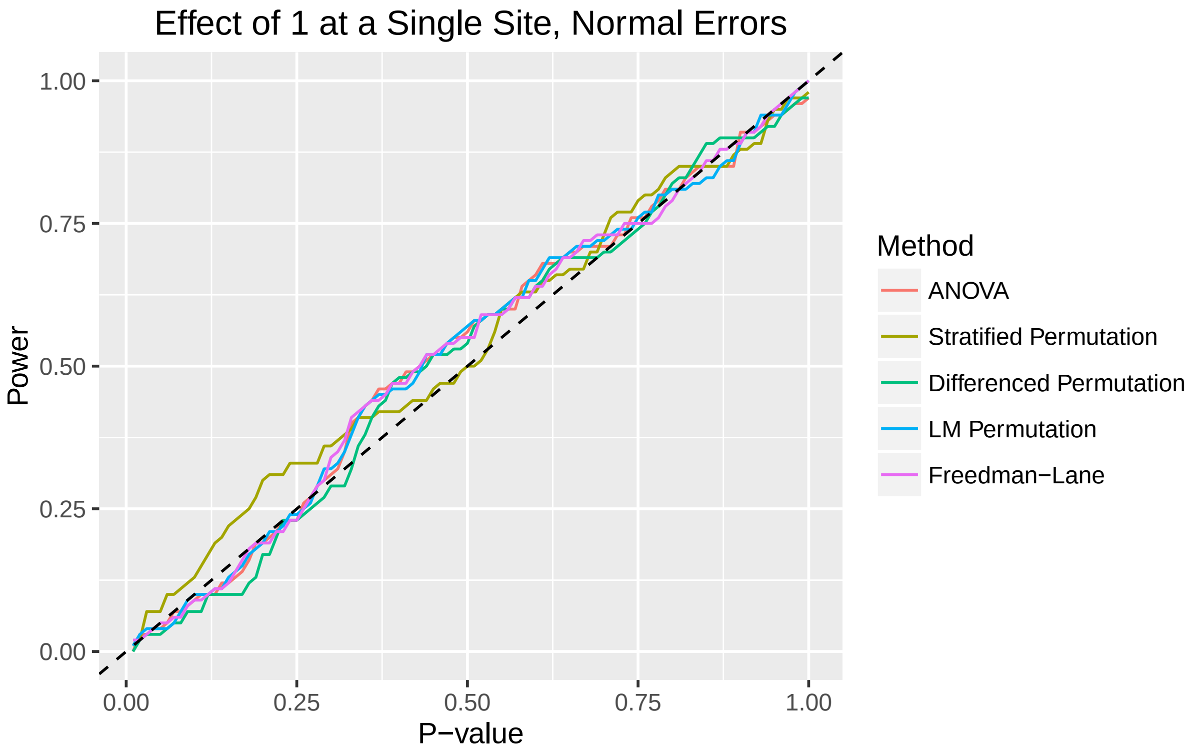


Design 7: Single site effect, normal errors

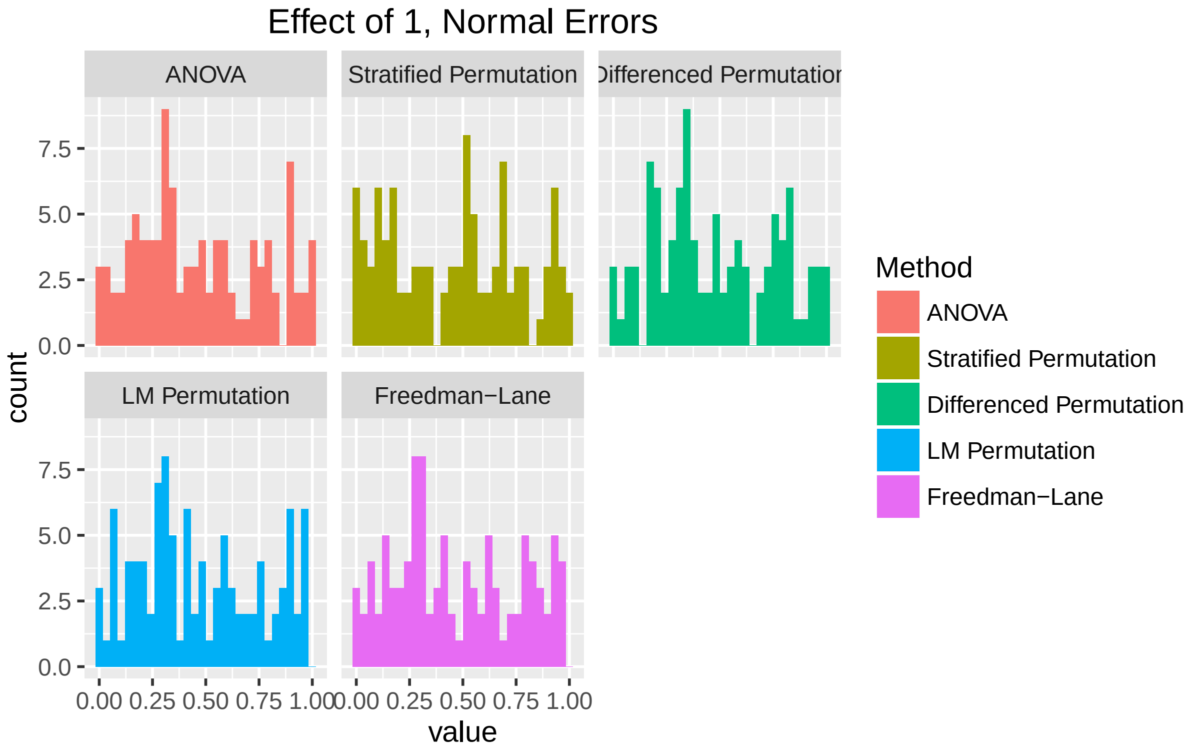
There is very low power to detect the effect here, since in the simulated data, the site with only 8 patients was the only site with a nonzero effect of treatment. The treatment effect is on the same order of magnitude as the error variance.

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

tmp <- generate_simulated_data(gamma = 1, effect = "single site effect",
    errors = "normal", n = c(8, 16, 24))
design7_pvalues <- replicate(100, {
    tmp$epsilon <- rnorm(nrow(tmp))
    tmp$Z <- permute_within_groups(tmp$Z, tmp$SITEID)
    tmp$Y1 <- tmp$Y0 + tmp$site_effect + tmp$gamma_vec * tmp$Z +
        tmp$epsilon
    generate_simulated_pvalues(tmp)
})
design7_pvalues <- t(design7_pvalues)
colnames(design7_pvalues) <- c("ANOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design7_power <- apply(design7_pvalues, 2, compute_power)
```

```
plot_power_curves(design7_power, "Effect of 1 at a Single Site, Normal Errors")
```



```
plot_pvalue_hist(design7_pvalues, "Effect of 1, Normal Errors")
```



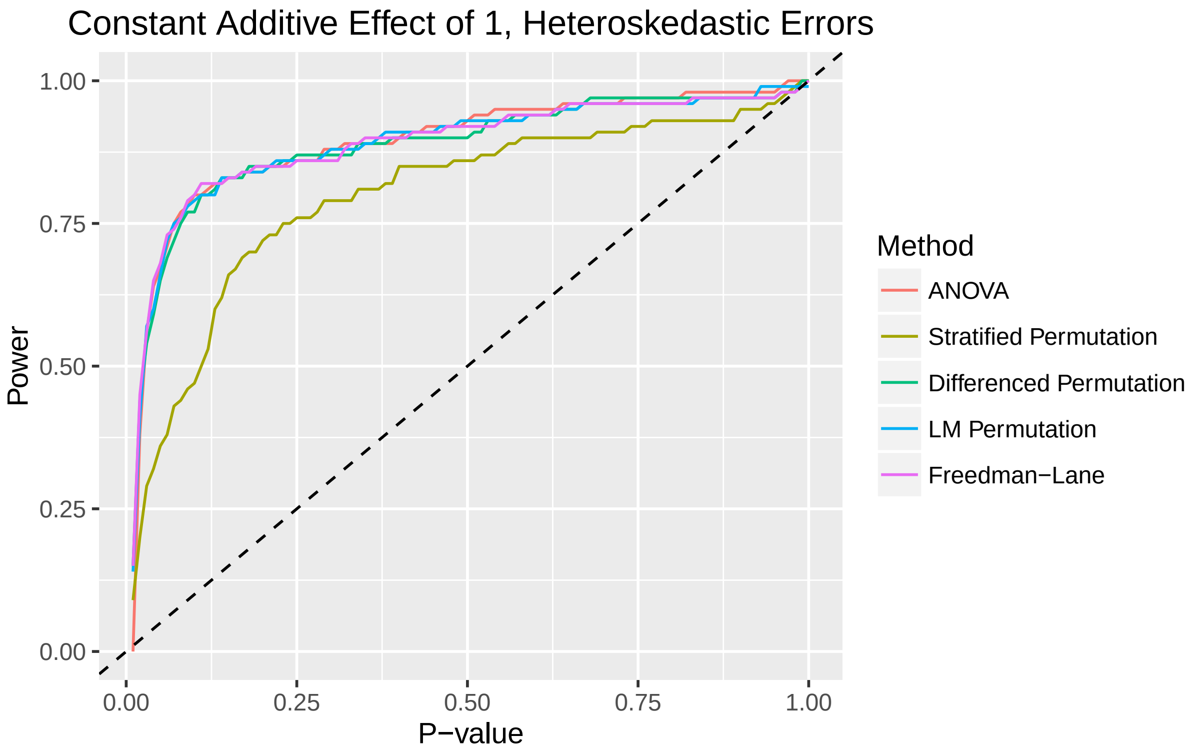
Heteroskedastic errors

One assumption of ANOVA is that errors are homoskedastic: they have the same variance across groups. Suppose that this does not hold in our data. Imagine we have three sites with 16 patients each, but they have different error distributions: site 1 has standard normal errors, site 2 has normally distributed errors with mean 0 and variance 2, and site 3 has errors with a *t*-distribution on 2 degrees of freedom.

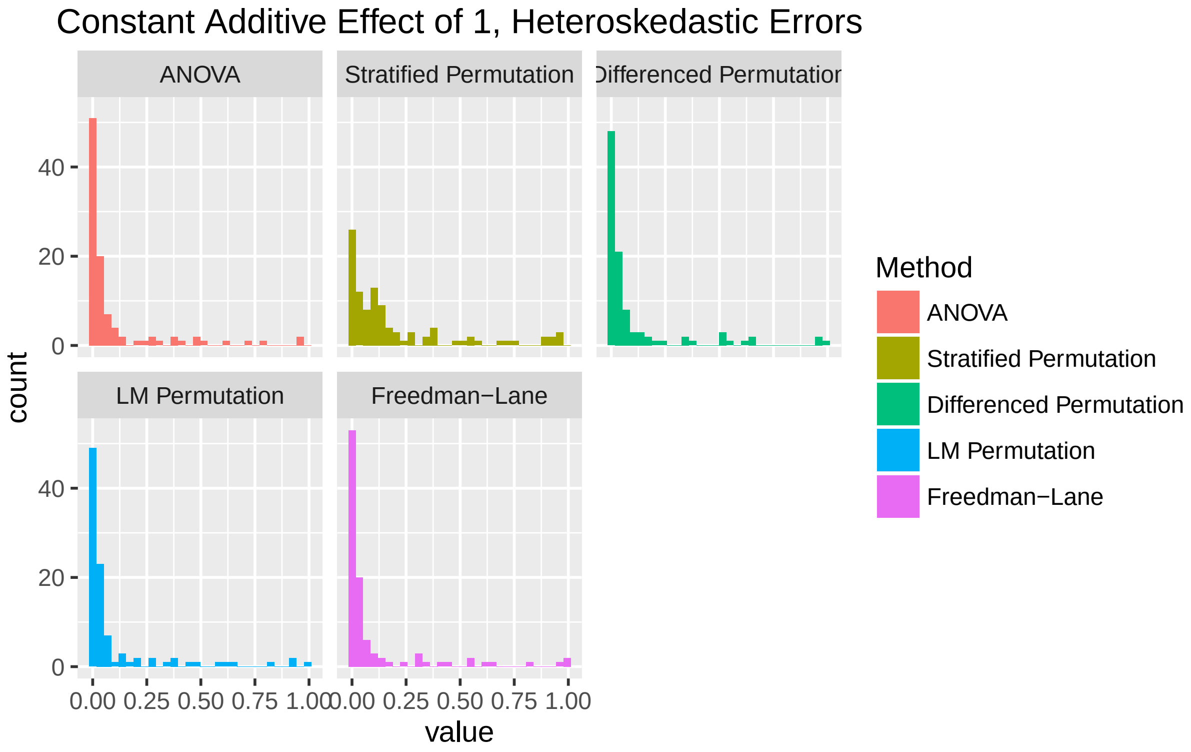
Design 8: heteroskedastic errors, constant treatment effect

```
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC
nn <- 16

tmp <- generate_simulated_data(gamma = 1, effect = "same effect",
    errors = "normal", n = rep(nn, 3))
design8_pvalues <- replicate(100, {
    tmp$epsilon[tmp$SITEID == 1] <- rnorm(nn)
    tmp$epsilon[tmp$SITEID == 2] <- rnorm(nn, sd = sqrt(2))
    tmp$epsilon[tmp$SITEID == 3] <- rt(nn, df = 2)
    tmp$Z <- permute_within_groups(tmp$Z, tmp$SITEID)
    tmp$Y1 <- tmp$Y0 + tmp$site_effect + tmp$gamma_vec * tmp$Z +
        tmp$epsilon
    generate_simulated_pvalues(tmp)
})

design8_pvalues <- t(design8_pvalues)
colnames(design8_pvalues) <- c("ANOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design8_power <- apply(design8_pvalues, 2, compute_power)
```

```
plot_power_curves(design8_power, "Constant Additive Effect of 1, Heteroskedastic Errors")
```



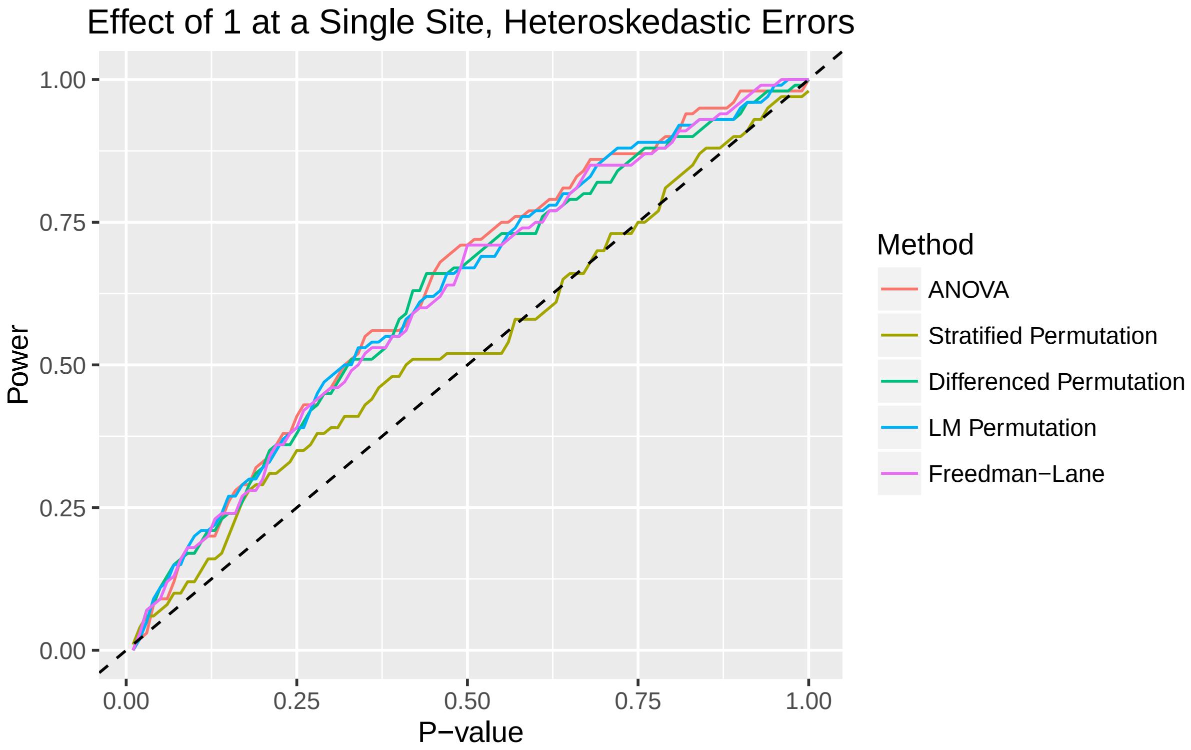
```r
plot_pvalue_hist(design8_pvalues, "Constant Additive Effect of 1, Heteroskedastic Errors")
```



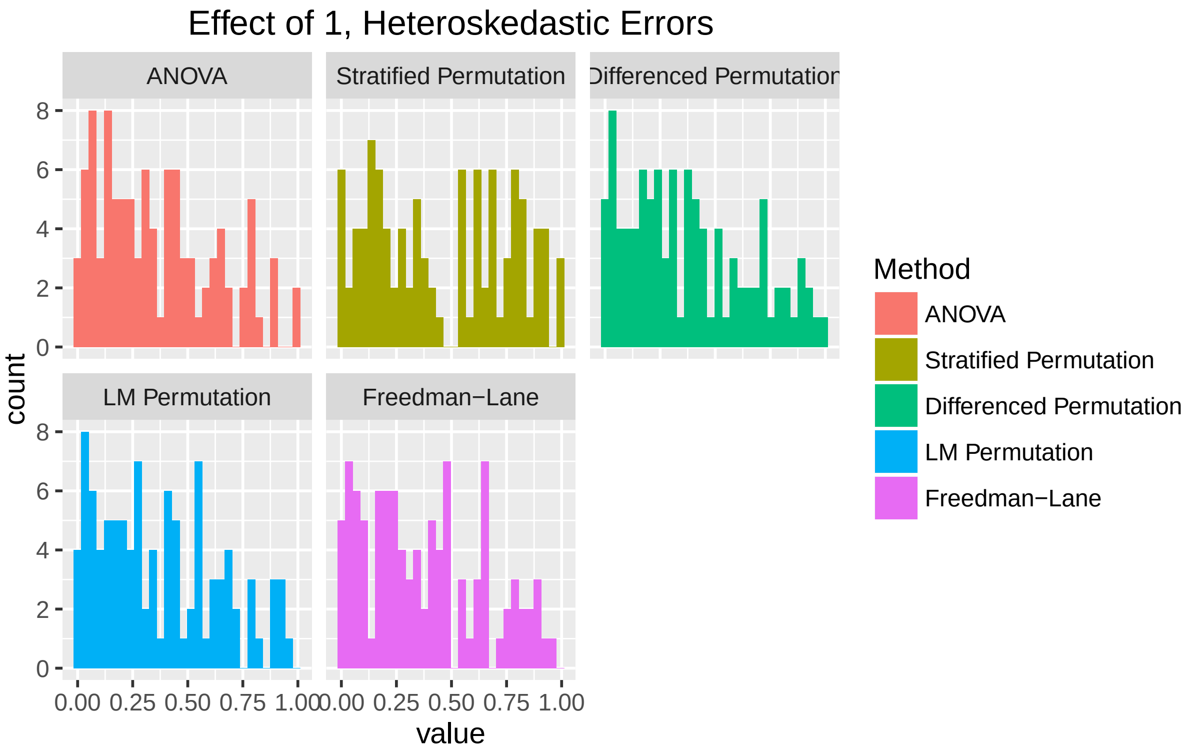
Design 9: heteroskedastic errors, single site effect

```
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC
nn <- 16

tmp <- generate_simulated_data(gamma = 1, effect = "single site effect",
    errors = "normal", n = rep(nn, 3))
design9_pvalues <- replicate(100, {
    tmp$epsilon[tmp$SITEID == 1] <- rnorm(nn)
    tmp$epsilon[tmp$SITEID == 2] <- rnorm(nn, sd = sqrt(2))
    tmp$epsilon[tmp$SITEID == 3] <- rt(nn, df = 2)
    tmp$Z <- permute_within_groups(tmp$Z, tmp$SITEID)
    tmp$Y1 <- tmp$Y0 + tmp$site_effect + tmp$gamma_vec * tmp$Z +
        tmp$epsilon
    generate_simulated_pvalues(tmp)
})
design9_pvalues <- t(design9_pvalues)
colnames(design9_pvalues) <- c("ANOVA", "Stratified Permutation",
    "Differenced Permutation", "LM Permutation", "Freedman-Lane")
design9_power <- apply(design9_pvalues, 2, compute_power)
```

```
plot_power_curves(design9_power, "Effect of 1 at a Single Site, Heteroskedastic Errors")
```



```
plot_pvalue_hist(design9_pvalues, "Effect of 1, Heteroskedastic Errors")
```


Effect of 1, Heteroskedastic Errors
ANOVA
Stratified Permutation
Differenced Permutation
LM Permutation
Freedman−Lane
count
0
2
4
6
8
value
0.00
0.25
0.50
0.75
1.00
Method
ANOVA
Stratified Permutation
Differenced Permutation
LM Permutation
Freedman−Lane