ANOVA Comparison Simulations

Kellie Ottoboni

2016-11-15

The model

We assume an additive linear model:

$$Y_{ij1} = \beta_0 Y_{i0} + \beta_j + \gamma_j Z_{ij} + \varepsilon_{ij}$$

for individuals $i = 1, \ldots, n_j$, $j = 1, \ldots, J$. β_0 is the coefficient for the 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.

set up 1: $\gamma_1 = \gamma_2 = \gamma_3$ set up 2: $\gamma_1 = \gamma > 0$, $\gamma_2 = \gamma_3 = 0$ errors 1: $\varepsilon \sim N(0, \sigma^2)$ errors 2: $\varepsilon \sim t(2)$

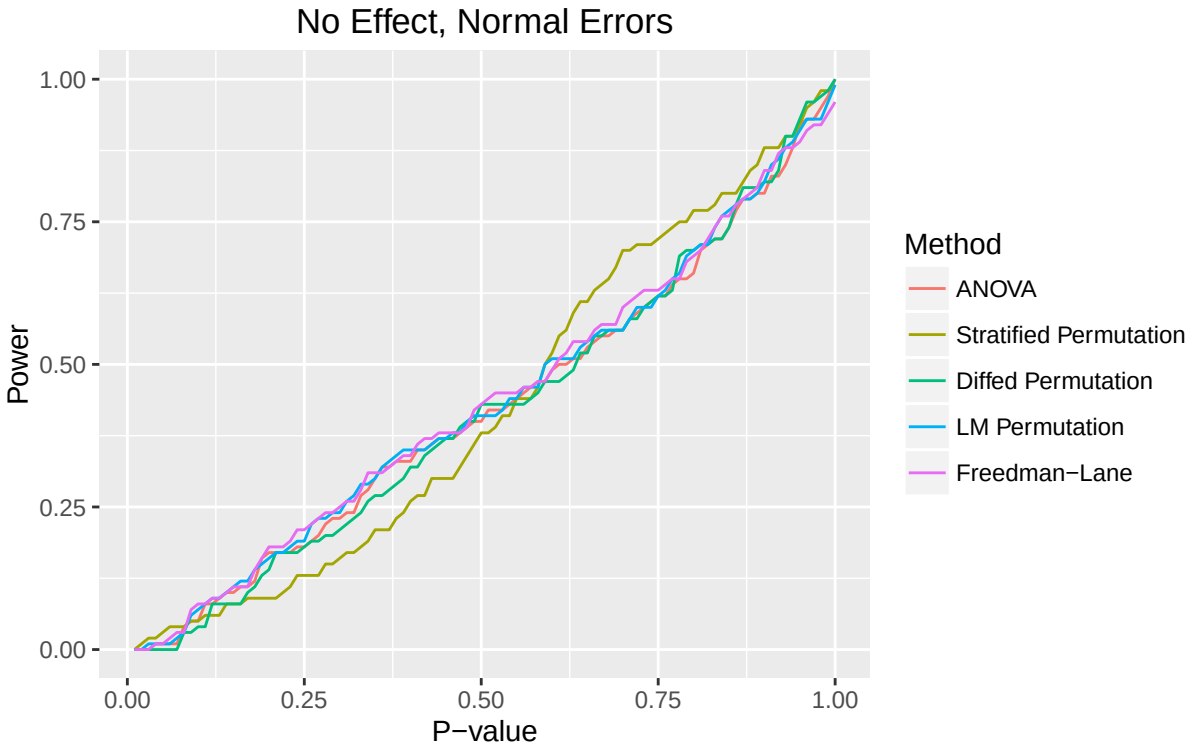
```
generate_simulated_data <- function(gamma, effect, errors, n = c(16,
    16, 16)) {
    # Input: gamma = the magnitude of the treatment effect effect
    # = 'same effect' or 'single site effect' - which sites have
    # a tr effect > 0?  errors = 'normal' or 'heavy' Returns: a
    # dataframe containing 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, SITEID))
```

```
}

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)"]

    # Permutation test
    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$SITEID, reps = reps)
    # diff_means_distr2 <- replicate(reps, { Z_perm <-
    # permute_within_groups(dataset$Z, dataset$SITEID)
    # mean(dataset$Y1[Z_perm == 1]) - mean(dataset$Y1[Z_perm ==
    # 0]) })
    perm_pvalue <- t2p(observed_diff_means, diff_means_distr,
        alternative = "two-sided")

    # Diffed permutation test
    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)
```

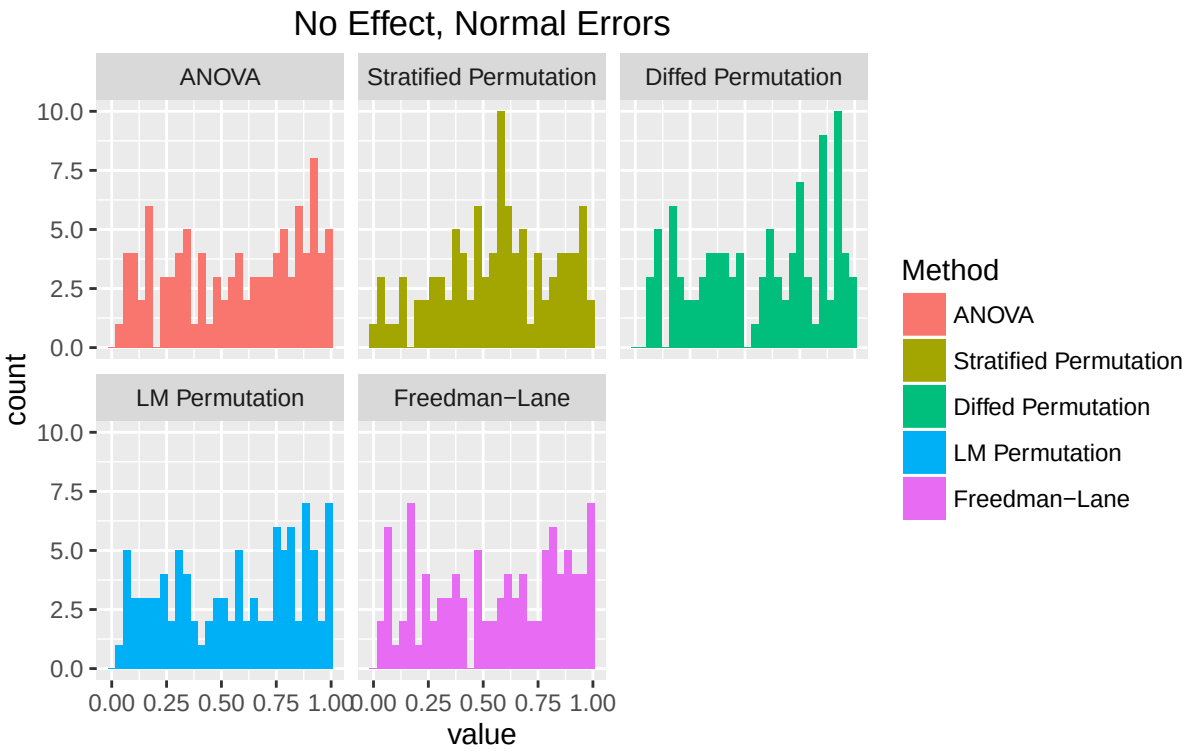
```
        dataset$response_fl <- lm2_yhat + lm2_resid_perm
        lm2_perm <- lm(response_fl ~ Y0 + Z + factor(SITEID),
            data = dataset)
        summary(lm2_perm)[["coefficients"]]["Z", "t value"]
    })
    fl_pvalue <- t2p(observed_t1, lm2_t_distr, alternative = "two-sided")

    return(c(ANOVA = anova_pvalue, `Stratified Permutation` = perm_pvalue,
        `Diffed Stratified 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() + 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)
}
```

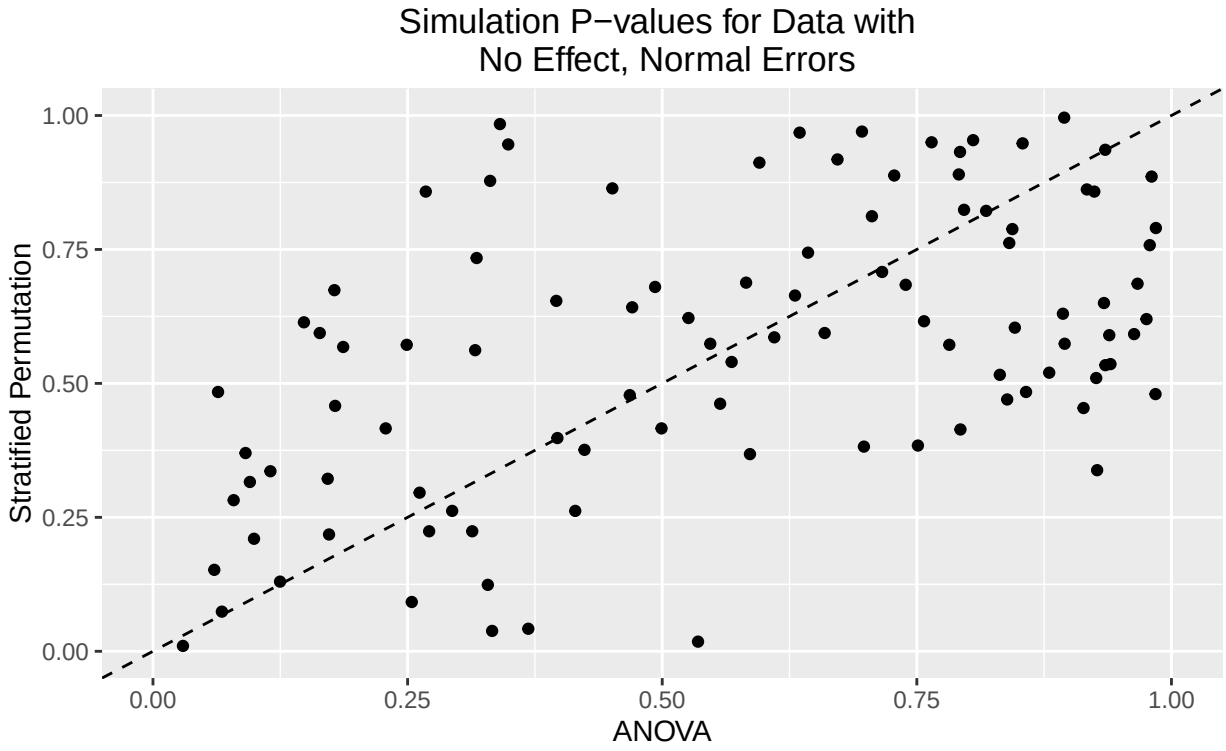
```
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC

design0_pvalues <- replicate(100, {
    tmp <- generate_simulated_data(gamma = 0, effect = "same effect",
        errors = "normal")
    generate_simulated_pvalues(tmp)
})
design0_pvalues <- t(design0_pvalues)
colnames(design0_pvalues) <- c("ANOVA", "Stratified Permutation",
    "Diffed Permutation", "LM Permutation", "Freedman-Lane")
design0_power <- apply(design0_pvalues, 2, compute_power)

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



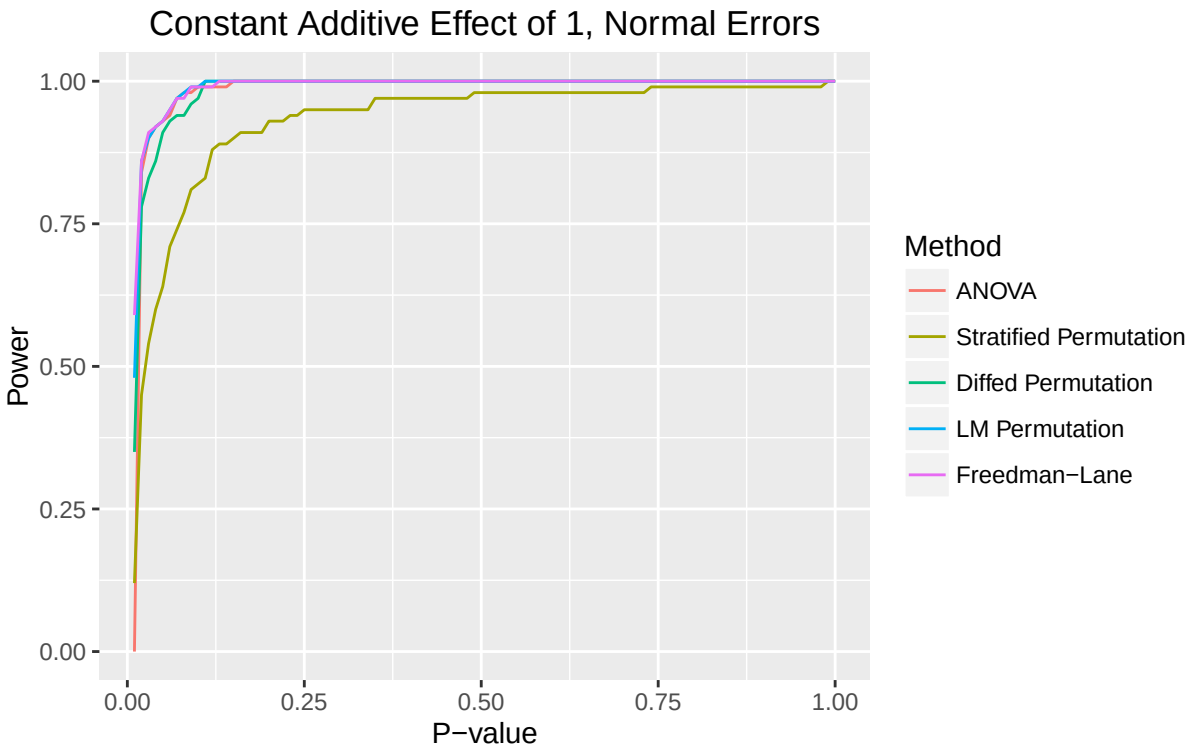

```
plot_pvalue_hist(design0_pvalues, "No Effect, Normal Errors")
```



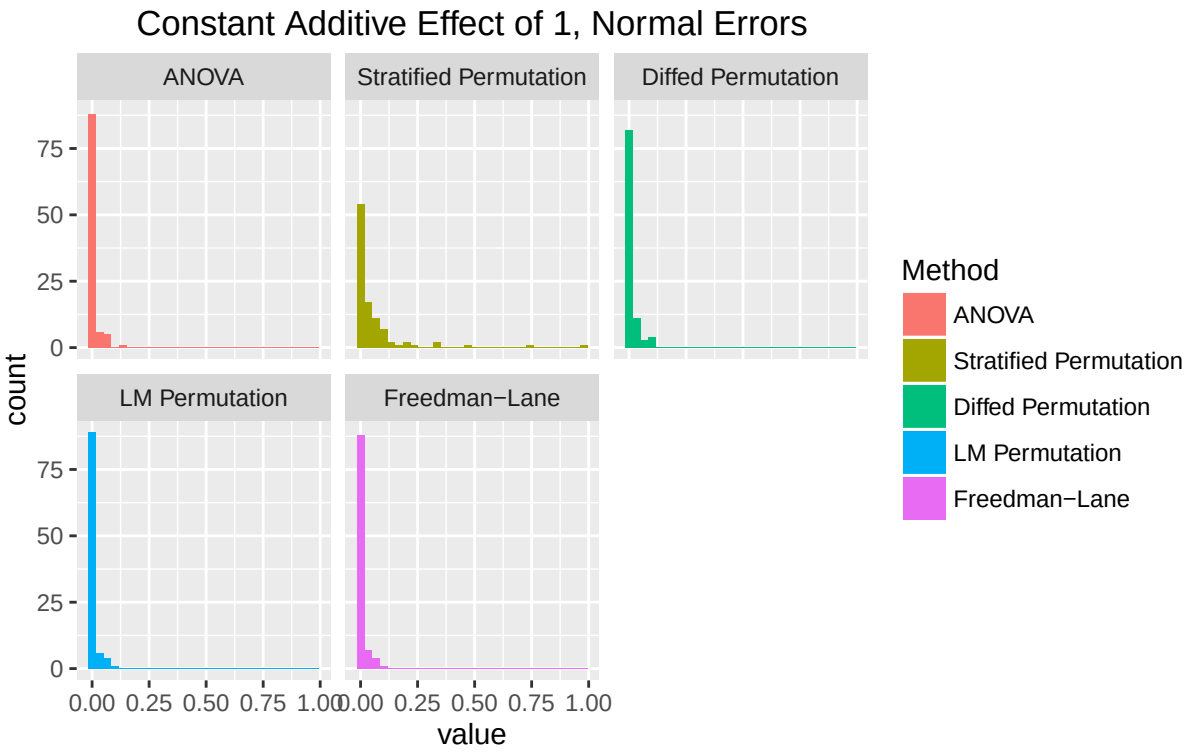

```
plot_pvalue_scatter(design0_pvalues, "Simulation P-values for Data with \n No Effect, Normal Errors")
```




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

design1_pvalues <- replicate(100, {
    tmp <- generate_simulated_data(gamma = 1, effect = "same effect",
        errors = "normal")
    generate_simulated_pvalues(tmp)
})
design1_pvalues <- t(design1_pvalues)
colnames(design1_pvalues) <- c("ANOVA", "Stratified Permutation",
    "Diffed Permutation", "LM Permutation", "Freedman-Lane")
design1_power <- apply(design1_pvalues, 2, compute_power)


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

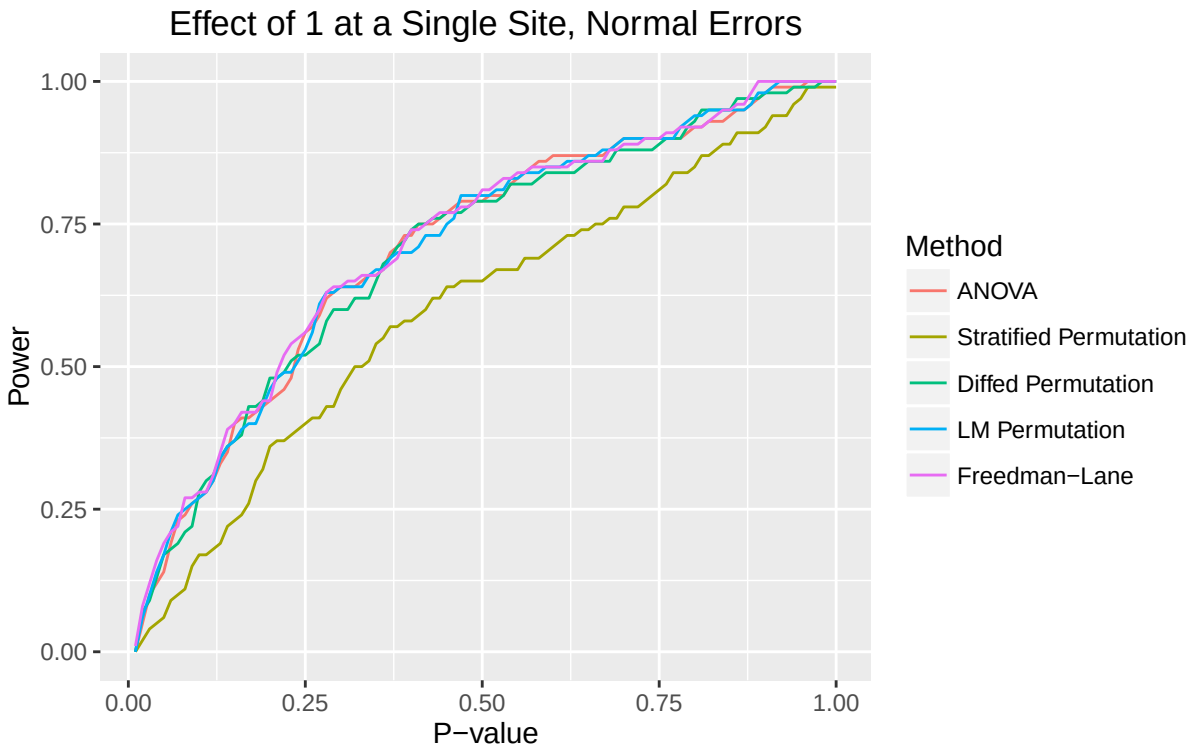



```
plot_pvalue_hist(design1_pvalues, "Constant Additive Effect of 1, Normal Errors")
```

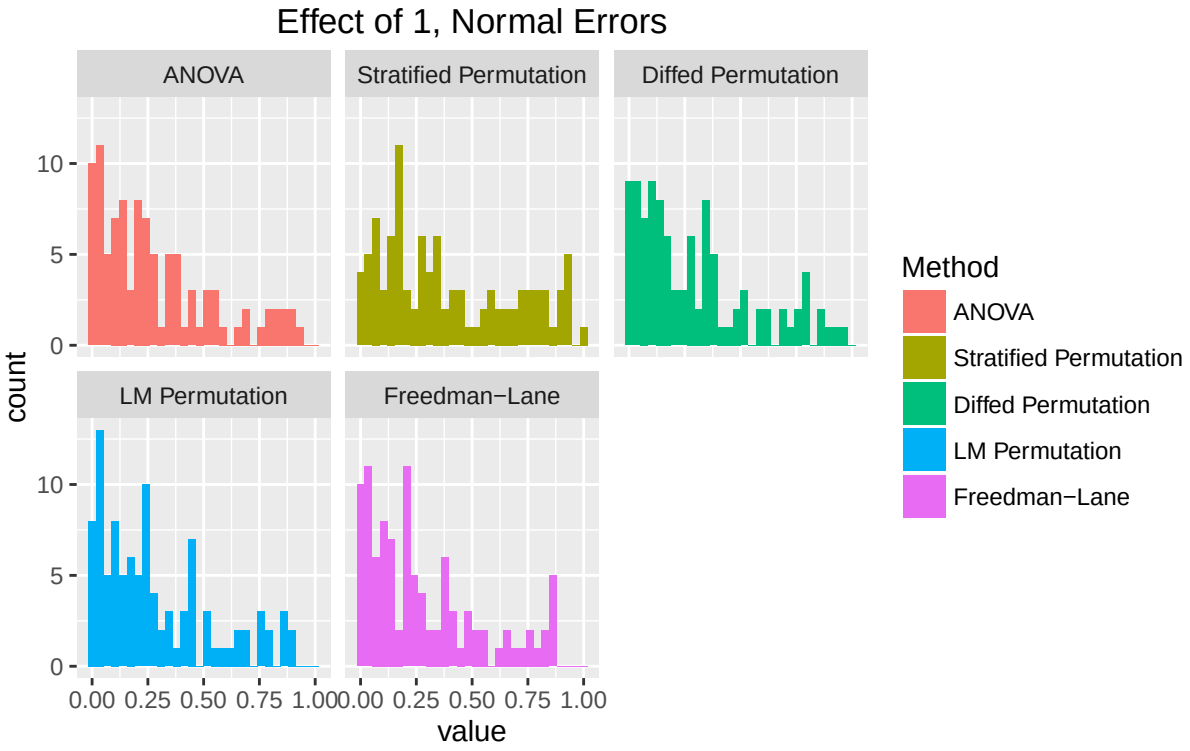



```
# plot_pvalue_scatter(design1_pvalues, 'Simulation P-values
# for Data with \n Constant Additive Effect of 1, Normal
# Errors')
```

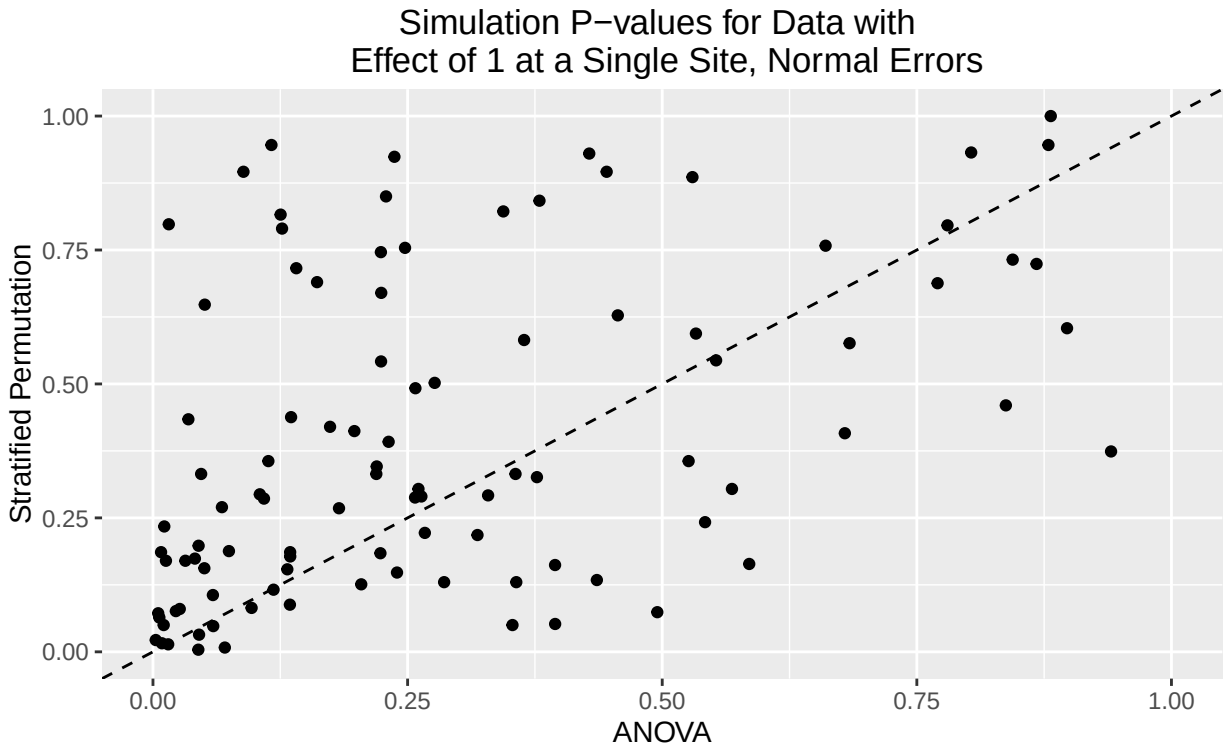
```r
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC

design2_pvalues <- replicate(100, {
    tmp <- generate_simulated_data(gamma = 1, effect = "single site effect",
        errors = "normal")
    generate_simulated_pvalues(tmp)
})
design2_pvalues <- t(design2_pvalues)
colnames(design2_pvalues) <- c("ANOVA", "Stratified Permutation",
    "Diffed 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")
```




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

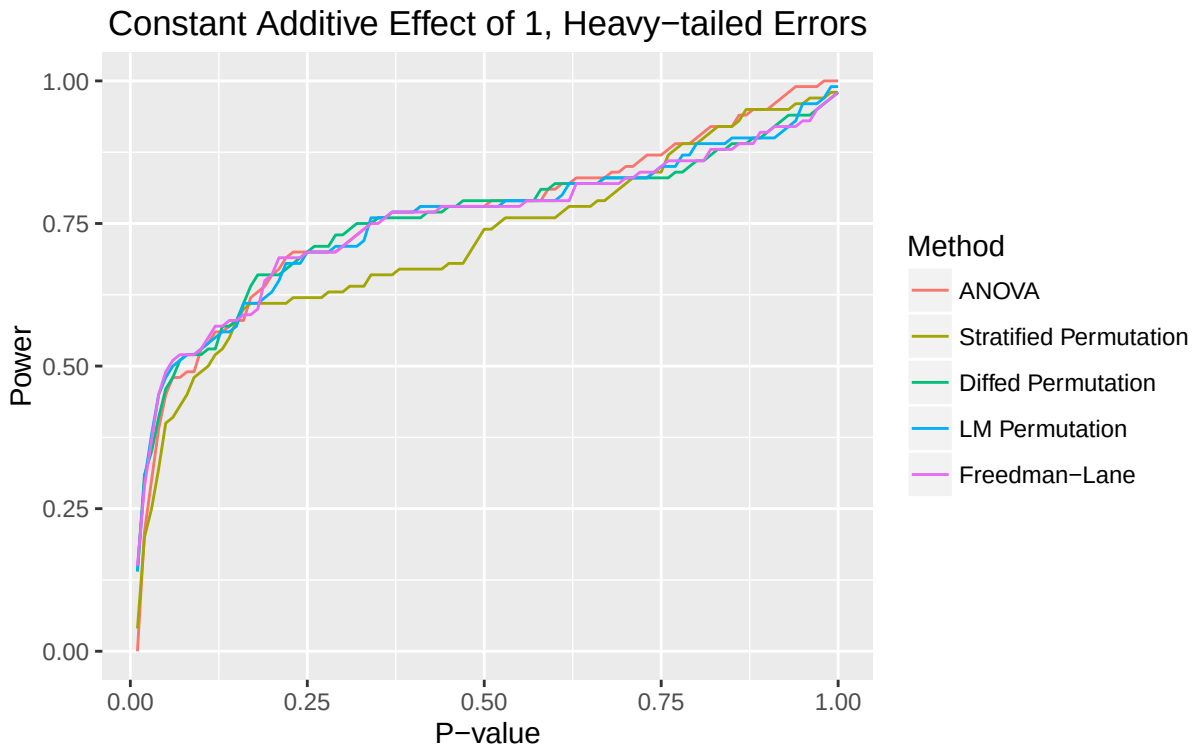



```
plot_pvalue_scatter(design2_pvalues, "Simulation P-values for Data with \n Effect of 1 at a Single Site
```

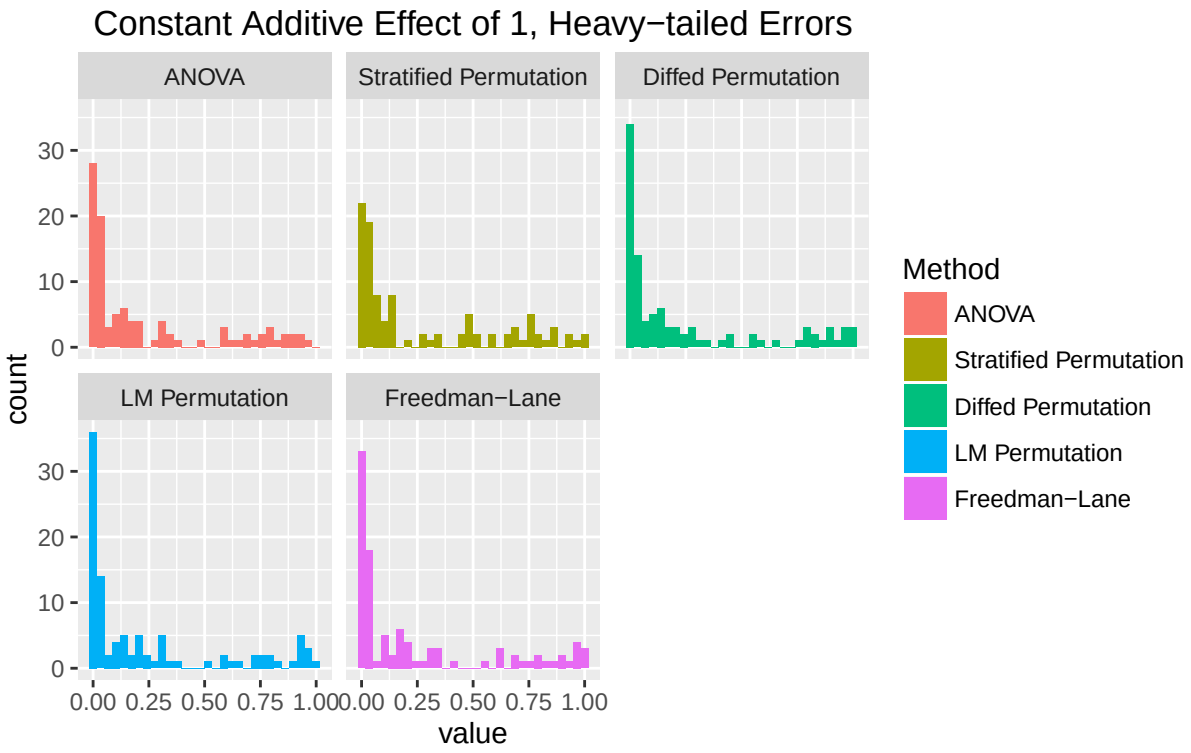



```
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC

design3_pvalues <- replicate(100, {
```

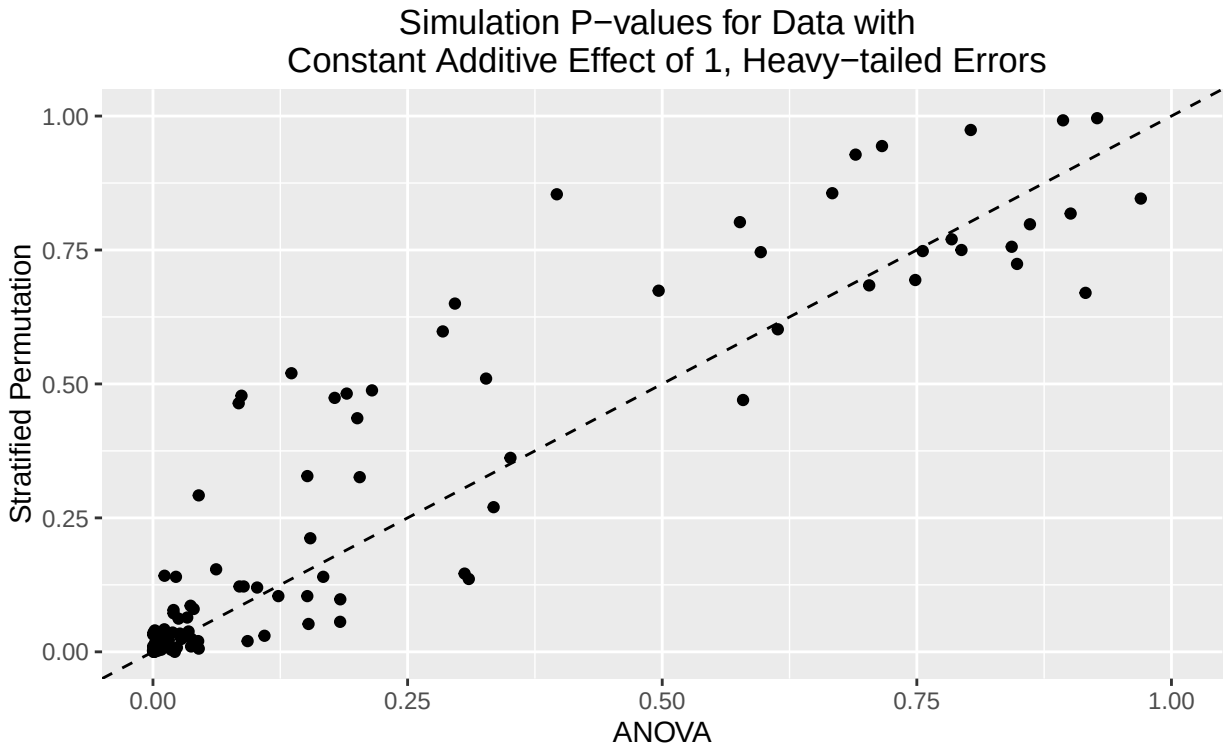
```
    tmp <- generate_simulated_data(gamma = 1, effect = "same effect",
        errors = "heavy")
    generate_simulated_pvalues(tmp)
})
design3_pvalues <- t(design3_pvalues)
colnames(design3_pvalues) <- c("ANOVA", "Stratified Permutation",
    "Diffed Permutation", "LM Permutation", "Freedman-Lane")
design3_power <- apply(design3_pvalues, 2, compute_power)

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




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

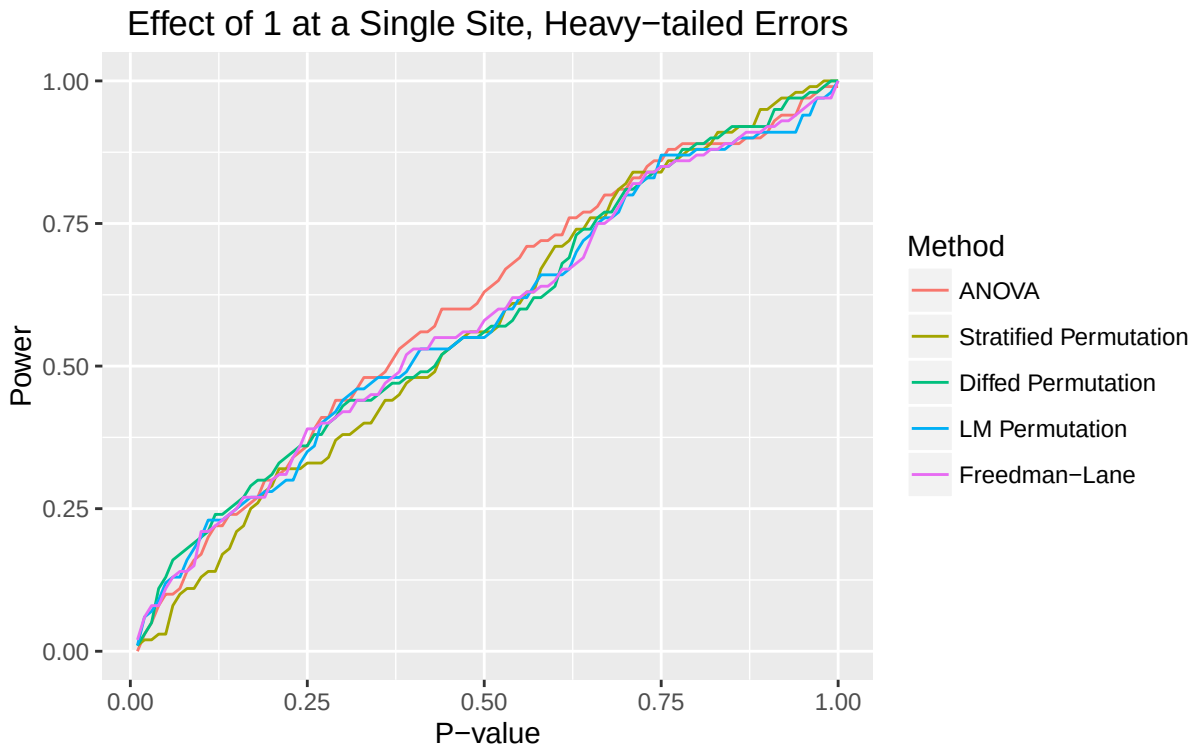



```
plot_pvalue_scatter(design3_pvalues, "Simulation P-values for Data with \n Constant Additive Effect of
```

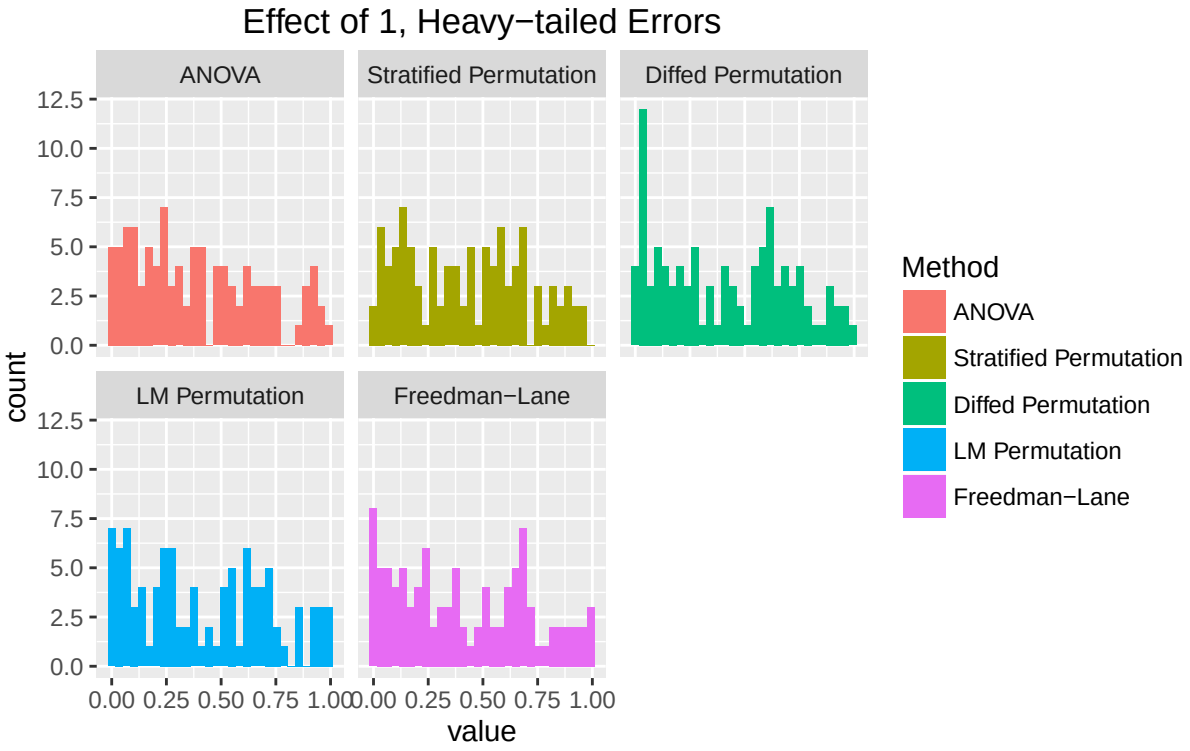



```
set.seed(760682460)  # Generated from random.org Timestamp: 2016-11-14 10:21:12 UTC

design4_pvalues <- replicate(100, {
```

```
    tmp <- generate_simulated_data(gamma = 1, effect = "single site effect",
        errors = "heavy")
    generate_simulated_pvalues(tmp)
})
design4_pvalues <- t(design4_pvalues)
colnames(design4_pvalues) <- c("ANOVA", "Stratified Permutation",
    "Diffed 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")
```




```
plot_pvalue_scatter(design4_pvalues, "Simulation P-values for Data with \n Effect of 1 at a Single Site
```