

RB Data

Kellie Ottoboni

2016-11-10

Data Cleaning

Missing data

There's a bunch of missing data. First, we notice there are 14 subjects who were only observed during the baseline period.

```
numdays <- table(ga$SUBJID)
baseline_only <- names(numdays)[numdays < 14]
unique_numdays <- as.data.frame(table(numdays))
names(unique_numdays) <- NULL
print(xtable(unique_numdays, caption = "Number of subjects observed for 7 or 14 days"),
      include.rownames = FALSE, comment = FALSE)
```

7	14
14	181

Table 1: Number of subjects observed for 7 or 14 days

```
ga$missing <- ifelse(ga$SUBJID %in% baseline_only, 1, 0)
```

Next, we check how many individuals were assigned to each treatment group. 52 subjects have treatment values of .. They appear to never have been surveyed during week 2: although there are records in the dataset, all values are 0. These 52 include the fourteen who were only observed during week 1. We remove all these missing subjects. We are left with a total of 66 subjects in the placebo group and 70 subjects in the active treatment group.

```
num_treated <- table(ga$tr)
baseline_only2 <- as.data.frame(num_treated/14)
colnames(baseline_only2) <- c("Treatment", "Number of subjects")
print(xtable(baseline_only2, digits = 0, caption = "Number of subjects in each treatment group"),
      include.rownames = FALSE, comment = FALSE)
```

Treatment	Number of subjects
.	52
1	66
2	70

Table 2: Number of subjects in each treatment group

```
ga$missing <- ga$missing + ifelse(ga$tr == ".", 1, 0)
```

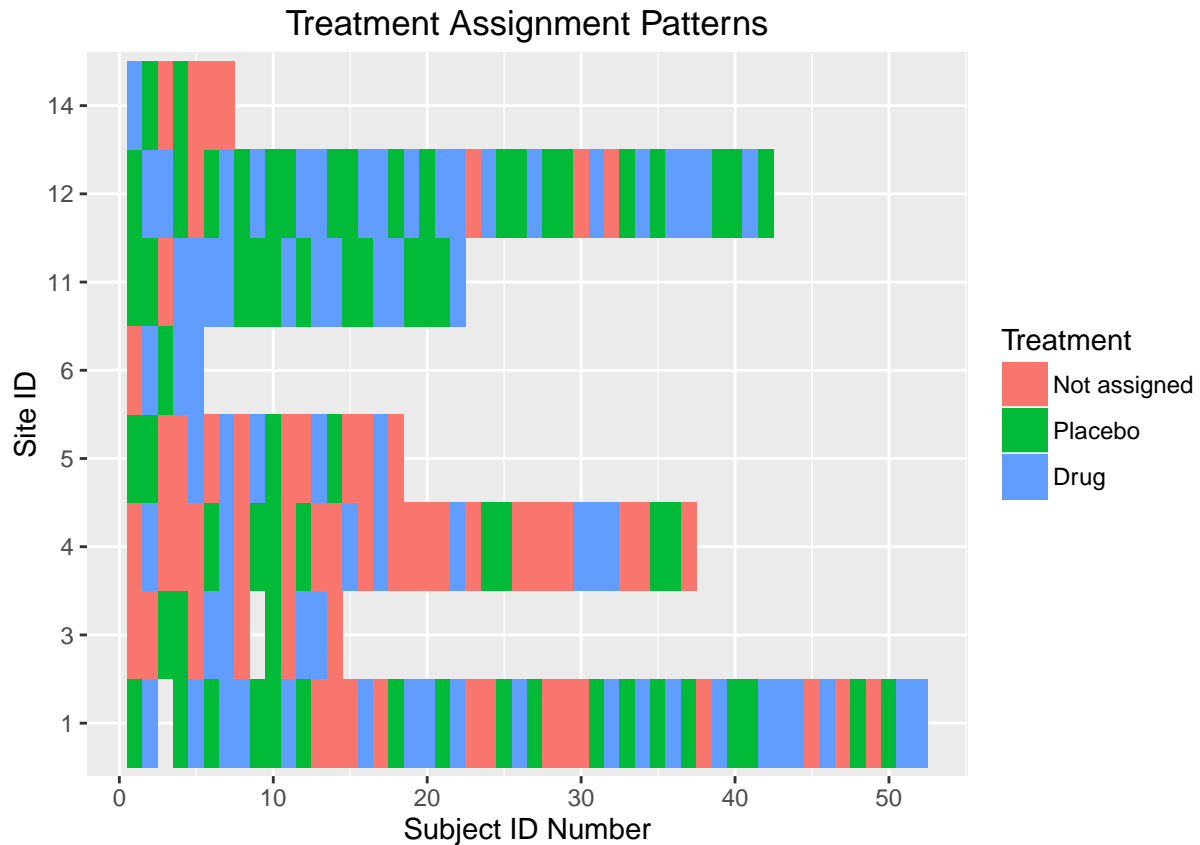


Figure 1: Number of subjects in each treatment group, by site ID.

Exploratory Data Analysis

We notice several things patterns from Figure 1, showing missing data patterns. First, some of the sites have many more subjects than others: these are sites 1, 4, and 12. Second, some of the sites have a large proportion of missing data, such as sites 3, 4, 5, and 14. Finally, we notice that sites 3, 6, and 14 have very few subjects, so we will have low power at these sites. For example, after removing the missing values, site 14 only has one treated subject and two placebos, so the site will only have 3 unique permutation statistics.

```
ga %>% mutate(subjid_by_site = SUBJID%%100) %>% ggplot(aes(x = subjid_by_site,
  y = factor(SITEID))) + geom_tile(aes(fill = tr)) + xlab("Subject ID Number") +
  ylab("Site ID") + scale_fill_discrete(name = "Treatment",
    breaks = c(".", "1", "2"), labels = c("Not assigned", "Placebo",
      "Drug")) + ggtitle("Treatment Assignment Patterns")
```

The data are not in the right shape to analyze a single variable. Below, we include a function to take the raw data and reshape it to analyze one variable of interest.

```
data_by_subjid_visitnum <- ga %>% group_by(SUBJID, VISITNUM)

reshape_ga_continuous <- function(variable, data = data_by_subjid_visitnum) {
  data <- data %>% mutate(myvariable = variable)
  summarized <- data %>% summarise(variable = mean(as.numeric(myvariable)),
    tr = unique(tr), SITEID = unique(SITEID))
}
```

```

cleaned <- dcast(summarized, SUBJID + tr + SITEID ~ VISITNUM,
  value.var = "variable")
colnames(cleaned) <- c("SUBJID", "tr", "SITEID", "Baseline",
  "Outcome")
cleaned <- ungroup(cleaned) %>% mutate(difference = Outcome -
  Baseline)
return(cleaned)
}

```

Figure ?? shows the distribution of the final outcome measures between treatment and placebo groups. Based on these distributions, we would expect to detect a difference in outcomes between the drug and placebo groups for the `daily_heart`, `daily_hrdq`, and `heart_freq` measures.

```

continuous_vars <- c("daily_heart", "daily_regurg", "daily_dysp",
  "daily_hrdq", "heart_freq", "regurg_freq", "dysp_freq")
plot_dists <- lapply(continuous_vars, function(variable) {
  p <- reshape_ga_continuous(variable) %>% mutate(tr = factor(tr)) %>%
    ggplot(aes(Outcome)) + geom_density(alpha = 0.6, aes(fill = tr)) +
    # geom_histogram(binwidth = 0.5, position = 'dodge', aes(fill
    # = tr)) +
    scale_fill_discrete(name = "Experimental\nCondition", breaks = c("0",
      "1"), labels = c("Placebo", "Drug")) + xlab(variable)
  return(p)
})

tmp <- ggplot_gtable(ggplot_build(plot_dists[[1]]))
leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
legend <- tmp$grobs[[leg]]

plot_dists <- lapply(plot_dists, function(x) x + theme(legend.position = "none"))
plot_dists[[length(plot_dists) + 1]] <- legend
do.call(grid.arrange, c(plot_dists, nrow = 2))

```

Ordinal variables

Several variables are measured on an ordinal scale but treated as continuous values. For instance, heartburn frequency is measured on a 7-point scale, with levels 0, 1, 2, 3, 4-5, 6-10, and more than 10. Values in the 4-5 range are coded as 4.5 and in the 6-10 range are coded as 8. We do not necessarily believe that these are accurate representations of the data. We recode the ordinal scales as factors on a scale from 1 to 7.

```

recode_ordinal <- function(variable) {
  levels <- variable <- as.factor(variable)
  levels(variable) <- seq_along(unique(variable))
  return(variable)
}

ga <- ga %>% mutate(heart_sev_f = recode_ordinal(heart_sev),
  regurg_sev_f = recode_ordinal(regurg_sev), dysp_sev_f = recode_ordinal(dysp_sev),
  heart_freq_f = recode_ordinal(heart_freq), regurg_freq_f = recode_ordinal(regurg_freq),
  dysp_freq_f = recode_ordinal(dysp_freq))

```

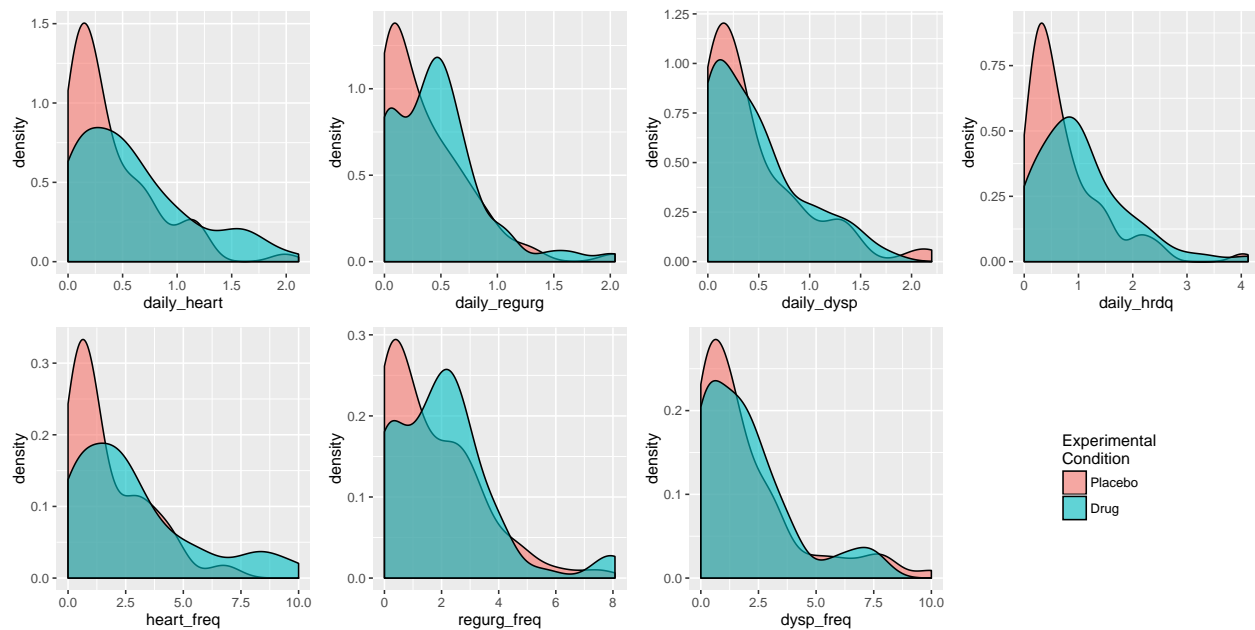


Figure 2: Distribution of continuous outcomes

We must put the data in a usable form so there is only one measurement for each individual. For each person, we take the median response from week 2 and subtract it from their median response from week 1.

```
data_by_subjid_day <- ga %>% mutate(dayofweek = rep(1:7, length.out = nrow(ga))) %>%
  group_by(SUBJID, dayofweek)
```

*# The reshape function below reduces the 14 days into a
 # single measure with the following steps: 1) Take the
 # difference in measurements between the first day of
 # treatment week and the first day of baseline week, the
 # second day of treatment week and the second day of baseline
 # week, and so on. 2) Take the median of these seven
 # numbers. We have decided not to use this method; it
 # assumes that day j in weeks 1 and 2 are paired, which may
 # not be right.*

```
# reshape_ga_ordinal <- function(variable, data =
# data_by_subjid_day){ data <- data %>% mutate(myvariable =
# variable) day_diff <- data %>% summarise('variable' =
# diff(as.numeric(myvariable)), 'tr' = unique(tr), 'SITEID' =
# unique(SITEID)) summarized <- day_diff %>%
# summarise('medoverdays' = median(variable), 'tr' =
# unique(tr), 'SITEID' = unique(SITEID)) summarized <-
# ungroup(summarized) return(summarized) }
reshape_ga_ordinal <- function(variable, data = data_by_subjid_visitnum) {
  data <- data %>% mutate(myvariable = variable)
  summarized <- data %>% summarise(variable = median(as.numeric(myvariable)),
    tr = unique(tr), SITEID = unique(SITEID))
  cleaned <- dcast(summarized, SUBJID + tr + SITEID ~ VISITNUM,
    value.var = "variable")
  colnames(cleaned) <- c("SUBJID", "tr", "SITEID", "Baseline",
```

```

    "Outcome")
  cleaned <- ungroup(cleaned) %>% mutate(difference = Outcome -
    Baseline)
  return(cleaned)
}

```

Figure ?? displays the distribution of these newly-coded ordinal variables in the treatment and control groups. There are certainly differences in distributions, but it is not clear how different they are.

```

ordinal_vars <- c("heart_sev_f", "regurg_sev_f", "dysp_sev_f",
  "heart_freq_f", "regurg_freq_f", "dysp_freq_f")
plot_dists <- lapply(ordinal_vars, function(variable) {
  p <- reshape_ga_ordinal(variable) %>% mutate(tr = factor(tr)) %>%
    ggplot(aes(difference, ..density..)) + geom_bar(aes(y = (..count..)/sum(..count..),
      fill = tr), position = "dodge") + scale_y_continuous(labels = scales::percent) +
    scale_fill_discrete(name = "Experimental\nCondition",
      breaks = c("0", "1"), labels = c("Placebo", "Drug")) +
    xlab(variable)
  return(p)
})

tmp <- ggplot_gtable(ggplot_build(plot_dists[[1]]))
leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
legend <- tmp$grobs[[leg]]

plot_dists <- lapply(plot_dists, function(x) x + theme(legend.position = "none"))
plot_dists[[length(plot_dists) + 1]] <- legend
do.call(grid.arrange, c(plot_dists, nrow = 3))

```

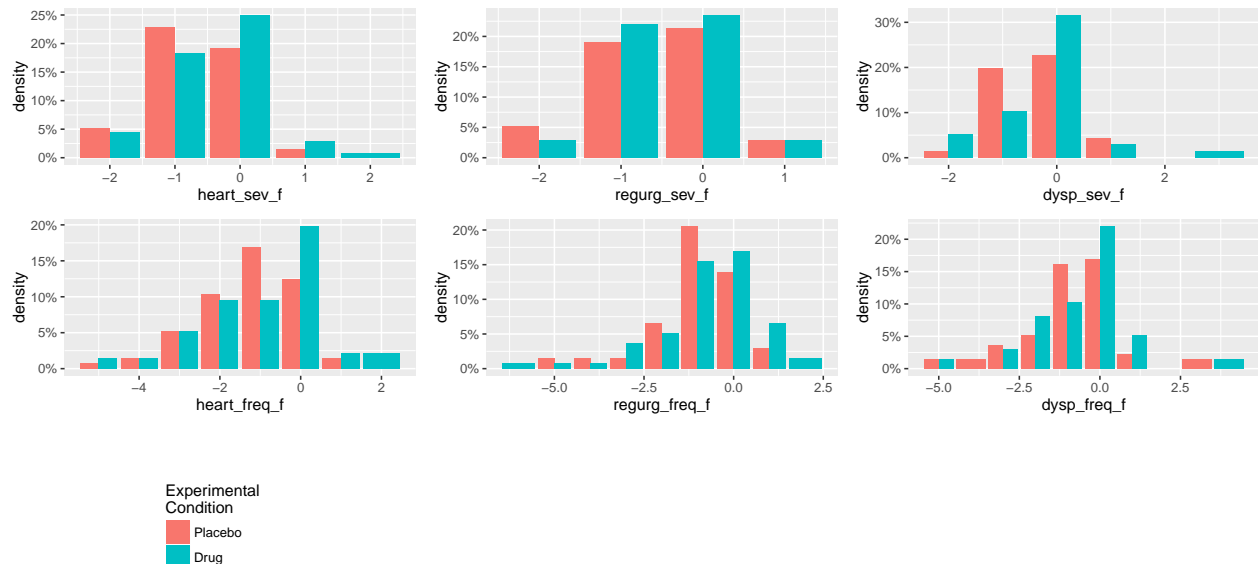


Figure 3: Distribution of ordinal outcomes

Analysis - Primary endpoint

Let's first restrict attention to the primary endpoint, `daily_hrdq`. We must transform the data to contain the variables of interest: the baseline score (averaged over 7 days of no treatment), the outcome score (averaged over 7 days of treatment), an indicator for which treatment was received (1 for the active drug, 0 for placebo), the site ID, and subject ID.

Throughout, we compare two models: the first model considers the final outcome measure, adjusting for site ID and perhaps also the baseline score. The second model considers the difference between outcome and baseline score as its dependent variable, adjusting for site ID.

We compare three testing methods: the usual parametric ANOVA, a permutation test considering the difference in means as test statistic, stratifying by site, and a permutation test in the style of Freedman and Lane (1983), using a linear regression to adjust for covariates.

```
daily_hrdq <- reshape_ga_continuous("daily_hrdq")
summary_stats_hrdq <- cbind(summary(daily_hrdq$difference[daily_hrdq$tr ==
  1]), summary(daily_hrdq$difference[daily_hrdq$tr == 0]))
colnames(summary_stats_hrdq) <- c("Drug", "Placebo")
print(xtable(summary_stats_hrdq, caption = "Summary statistics for change in primary endpoint (daily hr
  include.rownames = TRUE)
```

	Drug	Placebo
Min.	-2.79	-2.81
1st Qu.	-0.90	-1.10
Median	-0.60	-0.69
Mean	-0.57	-0.74
3rd Qu.	-0.16	-0.37
Max.	1.61	0.91

Table 3: Summary statistics for change in primary endpoint (daily hrdq) for drug and placebo groups.

Parametric ANOVA

First, we do a standard parametric ANOVA using the two models. Under model 1, using the outcome as response and adjusting for the baseline value, treatment has a significant effect. Under model 2, the significance goes away.

```
# method 0: parametric ANOVA
lm1 <- lm(Outcome ~ Baseline + tr + factor(SITEID), data = daily_hrdq)
print(xtable(summary(aov(lm1))), include.rownames = TRUE)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Baseline	1	27.00	27.00	65.16	0.0000
tr	1	1.73	1.73	4.17	0.0432
factor(SITEID)	7	4.50	0.64	1.55	0.1563
Residuals	126	52.20	0.41		

```
daily_hrdq$lm1_resid <- residuals(lm1)

lm2 <- lm(difference ~ tr + factor(SITEID), data = daily_hrdq)
print(xtable(summary(aov(lm2))), include.rownames = TRUE)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
tr	1	0.92	0.92	1.71	0.1936
factor(SITEID)	7	5.48	0.78	1.46	0.1876
Residuals	127	68.13	0.54		

```
daily_hrdq$lm2_resid <- residuals(lm2)
```

Stratified permutation test

Suppose each individual, $i = 1, \dots, N$ has two potential outcomes $(Y_i(1), Y_i(0))$ that indicate their response to the active drug and to the placebo, respectively. We randomly assign treatment to individuals; treatment determines which of $Y_i(1)$ and $Y_i(0)$ we observe. We are unable to observe both. Suppose we wish to test the null hypothesis that individual by individual, treatment has no effect. This is referred to as the “sharp null” hypothesis:

$$H_0 : Y_i(1) = Y_i(0), i = 1, \dots, N.$$

Then, whether individual i received the drug or placebo amounts to an arbitrary label. Once we observe their response under treatment T_i , we know what it would have been under treatment $1 - T_i$, for $T_i \in \{0, 1\}$; namely, it would be the same.

Treatment was completely randomized at each site, independently across sites. Assuming that drop-out from the study was independent of treatment assignment (which is true if treatment was blinded and assigned at random), then we may condition on the number of individuals who received the drug and who received the placebo at each site. Any assignment of treatments which preserves the number of treated and controls at each site is valid and has equal probability of occurring. Therefore, by using this principle of equal probabilities and by imputing the unobserved potential outcomes assuming that the null hypothesis is true, we may obtain the permutation distribution of any statistic under the null hypothesis.

We compare two test statistics: the usual difference in means between treatment and placebo groups, and the difference in means within each site, aggregated by taking the sum of their absolute values over sites. The first statistic is more comparable to what one would obtain from an ANOVA and it is readily interpretable. It does not directly account for the stratification by site, so variation between sites may be hidden. The second statistic is useful for testing the two-sided alternative hypothesis that treatment has some effect. It may be more powerful than the simple difference in means if the effect of the drug varies across sites. For instance, if the drug has a positive effect at one site but a negative effect at another site, this test statistic would be large; if we used the simple difference in means without accounting for sites, then the positive effect and negative effect may cancel each other out and appear as no effect.

Figure ?? shows the permutation distributions of the simple difference in means. Figure ?? shows the permutation distribution of the absolute value of the difference of means within strata, summed across strata. There are several points to notice: * The second test is far less powerful than the first. This is likely because some sites had very few patients. These sites have correspondingly few unique permutations, and so they often contribute the same amount to the summed difference in means statistic. It is impossible to detect a difference at small significance levels when there are only 3 patients. * The summed difference of means only makes sense for testing against a two-sided alternative, since taking the absolute value removes the sign of the effect. The permutation distribution is strictly non-negative and skewed to the right. * This statistic is less interpretable than the difference in means because it doesn’t correspond directly to any feature of the distribution of outcomes.

Since the summed difference in means statistic has such lower power, we abandon it for the rest of the analyses.

```

# method 1 : do permutation of differences
observed_diff_means1 <- mean(daily_hrdq[daily_hrdq$tr == 1, ]$Outcome) -
  mean(daily_hrdq[daily_hrdq$tr == 0, ]$Outcome)
diff_means_distr1 <- stratified_two_sample(group = daily_hrdq$tr,
  response = daily_hrdq$Outcome, stratum = daily_hrdq$SITEID,
  reps = 10000)
diff_means_pvalue1 <- t2p(observed_diff_means1, diff_means_distr1,
  alternative = "two-sided")

observed_diff_means2 <- mean(daily_hrdq[daily_hrdq$tr == 1, ]$difference) -
  mean(daily_hrdq[daily_hrdq$tr == 0, ]$difference)
diff_means_distr2 <- stratified_two_sample(group = daily_hrdq$tr,
  response = daily_hrdq$difference, stratum = daily_hrdq$SITEID,
  reps = 10000)
diff_means_pvalue2 <- t2p(observed_diff_means2, diff_means_distr2,
  alternative = "two-sided")

data.frame(perm = c(diff_means_distr1, diff_means_distr2), model = c(rep("Outcome",
  length(diff_means_distr1)), rep("Differenced", length(diff_means_distr2))),
  xintercept = c(rep(observed_diff_means1, length(diff_means_distr1)),
    rep(observed_diff_means2, length(diff_means_distr2)))) %>%
  ggplot(aes(x = perm, fill = model)) + geom_histogram() +
  facet_grid(~model) + geom_vline(aes(xintercept = xintercept))

```

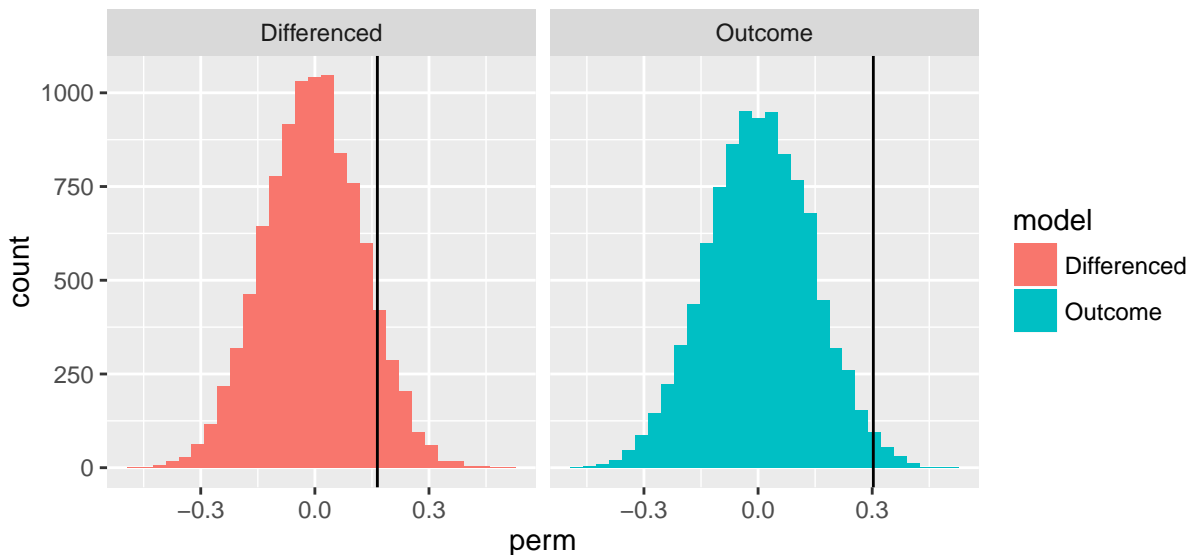


Figure 4: Permutation distribution of the difference of means for daily_hrdq

```

obs_diff_means_bystrata1 <- sum(abs(within_group_mean(group = daily_hrdq$tr,
  response = daily_hrdq$Outcome, stratum = daily_hrdq$SITEID,
  groups = unique(daily_hrdq$tr), strata = unique(daily_hrdq$SITEID))))
diff_means_distr_bystrata1 <- stratified_two_sample(group = daily_hrdq$tr,
  response = daily_hrdq$Outcome, stratum = daily_hrdq$SITEID,
  stat = "mean_within_strata", reps = 10000)

```



```

diff_means_bystrata_pvalue1 <- t2p(obs_diff_means_bystrata1,
  diff_means_distr_bystrata1, alternative = "two-sided")

obs_diff_means_bystrata2 <- sum(abs(within_group_mean(group = daily_hrdq$tr,
  response = daily_hrdq$difference, stratum = daily_hrdq$SITEID,
  groups = unique(daily_hrdq$tr), strata = unique(daily_hrdq$SITEID))))
diff_means_distr_bystrata2 <- stratified_two_sample(group = daily_hrdq$tr,
  response = daily_hrdq$difference, stratum = daily_hrdq$SITEID,
  stat = "mean_within_strata", reps = 10000)
diff_means_bystrata_pvalue2 <- t2p(obs_diff_means_bystrata2,
  diff_means_distr_bystrata2, alternative = "two-sided")

data.frame(perm = c(diff_means_distr_bystrata1, diff_means_distr_bystrata2),
  model = c(rep("Outcome", length(diff_means_distr_bystrata1)),
    rep("Differenced", length(diff_means_distr_bystrata2))),
  xintercept = c(rep(obs_diff_means_bystrata1, length(diff_means_distr_bystrata1)),
    rep(obs_diff_means_bystrata2, length(diff_means_distr_bystrata2)))) %>%
  ggplot(aes(x = perm, fill = model)) + geom_histogram() +
  facet_grid(~model) + geom_vline(aes(xintercept = xintercept))

```

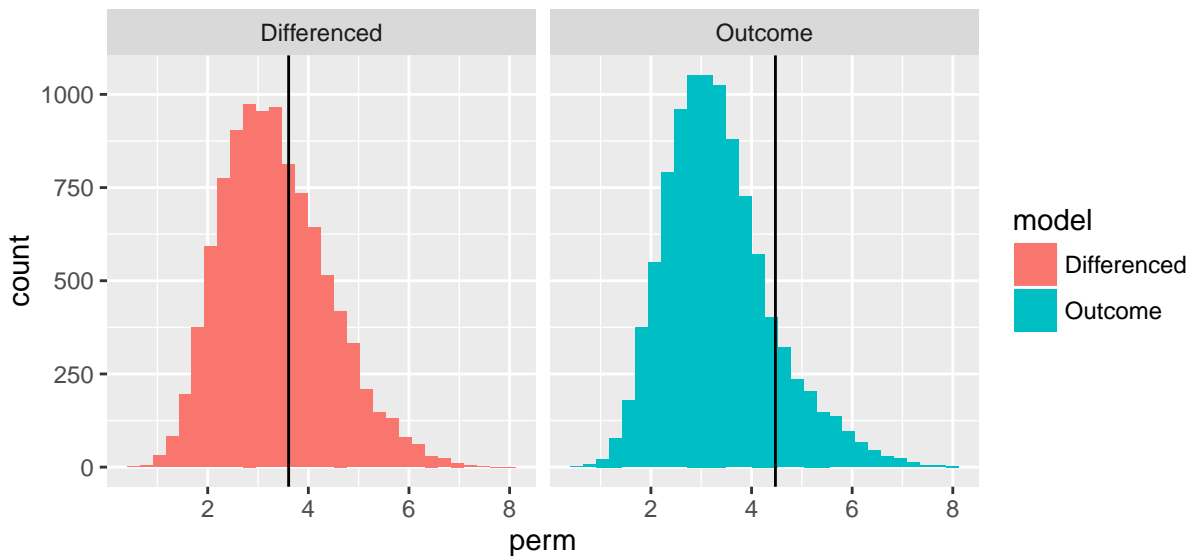


Figure 5: Permutation distribution of the difference of means within strata, summed across strata, for daily_hrdq

Covariate-adjusted permutation test

We would like to test for a difference in outcomes between the active drug and placebo group, but control for other covariates. In particular, when the outcome is the average response during the second week of follow-up, we would like to control for the average response during the first week of follow-up and the location. When the outcome is the difference in responses between the second and first weeks, we would just like to control for location. We will use the approximate permutation test derived by Freedman and Lane (1983) to do so.

Let Y denote the response, Z denote the treatment indicator, and X denote a covariate which may be correlated with Y . We may write the following equation:

$$Y = \beta_0 + \beta_1 X + \beta_2 Z + \varepsilon$$

Under the sharp null H_0 , Z has no effect on Y . In other words, the null hypothesis is that $\beta_2 = 0$. In a randomized experiment such as this, there are two ways we may test this null hypothesis.

Standard linear model

First, we may permute the treatment assignments Z directly. Treatment was assigned at random within each site. This ensures that Z and ε are statistically independent, conditional on site. Therefore, we may conduct a test by permuting treatment assignments Z within site, independently across sites, and calculating a test statistic for each such permutation. We choose to use the t statistic from the linear model as the test statistic.

```
# method 2 : linear model

observed_t1 <- summary(lm1)[["coefficients"]][["tr", "t value"]]
lm1_t_distr <- replicate(10000, {
  daily_hrdq$tr_perm <- permute_within_groups(daily_hrdq$tr,
    daily_hrdq$SITEID)
  lm1_perm <- lm(Outcome ~ Baseline + tr_perm + factor(SITEID),
    data = daily_hrdq)
  summary(lm1_perm)[["coefficients"]][["tr_perm", "t value"]]
})

lm_pvalue_1 <- t2p(observed_t1, lm1_t_distr, alternative = "two-sided")

observed_t2 <- summary(lm2)[["coefficients"]][["tr", "t value"]]
lm2_t_distr <- replicate(10000, {
  daily_hrdq$tr_perm <- permute_within_groups(daily_hrdq$tr,
    daily_hrdq$SITEID)
  lm2_perm <- lm(difference ~ tr_perm + factor(SITEID), data = daily_hrdq)
  summary(lm2_perm)[["coefficients"]][["tr_perm", "t value"]]
})

lm_pvalue_2 <- t2p(observed_t2, lm2_t_distr, alternative = "two-sided")

data.frame(perm = c(lm1_t_distr, lm2_t_distr), model = c(rep("Outcome",
  length(lm1_t_distr)), rep("Differenced", length(lm2_t_distr))),
  xintercept = c(rep(observed_t1, length(lm1_t_distr)), rep(observed_t2,
    length(lm2_t_distr)))) %>% ggplot(aes(x = perm, fill = model)) +
  geom_histogram() + facet_grid(~model) + geom_vline(aes(xintercept = xintercept)) +
  xlab("t Statistic") + ylab("Frequency") + ggtitle("Permutation Distributions")
```

Linear model residuals

Let's take an alternative view of the problem. We still write $Y = \beta_0 + \beta_1 X + \beta_2 Z + \varepsilon$. However, now, we do not treat the ε as random. They are simply defined to be the difference between Y and the data's linear projection onto the plane $\beta_0 + \beta_1 X + \beta_2 Z$.

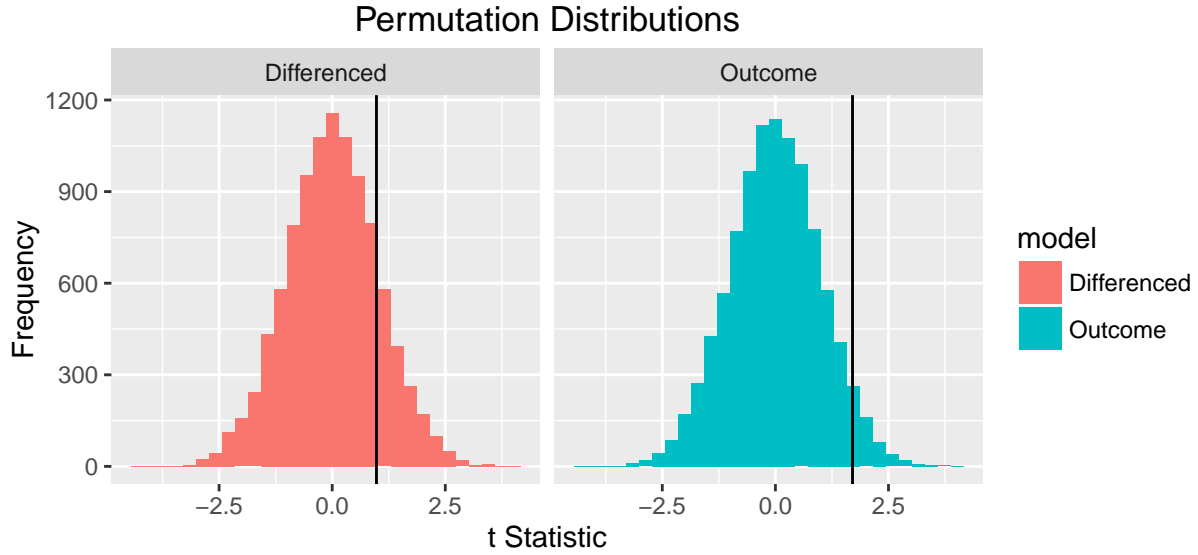


Figure 6: Permutation distribution of the stratified permutation test between `daily_hrdq` and treatment after controlling for covariates in a linear model.

If the null hypothesis is true, then $\varepsilon = Y - \beta_0 - \beta_1 X$. Therefore, we may estimate the errors $\hat{\varepsilon}$ by $Y - \hat{Y}$, where \hat{Y} is obtained by regressing Y on X but not Z . They approximate the true errors ε from the equation $Y = \beta_0 + \beta_1 X + \beta_2 Z + \varepsilon$. The ε are orthogonal to X and Z . Therefore, within sites, these estimated $\hat{\varepsilon}$ are approximately exchangeable.

We construct a permutation distribution using several steps:

1. Estimate $\hat{\varepsilon}$ by $Y - \hat{\beta}_0 - \hat{\beta}_1 X$, where $\hat{\beta}_0$ and $\hat{\beta}_1$ are obtained by regressing Y on X .
2. Construct permuted errors $\hat{\varepsilon}^\pi$ by permuting the $\hat{\varepsilon}$ within sites.
3. Construct permuted responses $Y^\pi = \hat{\beta}_0 + \hat{\beta}_1 X + \hat{\varepsilon}^\pi$.
4. Regress Y^π on X and Z . The test statistic is the t -statistic for the coefficient of Z .

For this dataset, we are guaranteed that treatment Z is independent of ε , as it is randomized within site. We should also check that site ID is uncorrelated with *varepsilon*. If not, then the $\hat{\varepsilon}$ are not exchangeable across sites. However, our permutations are done within sites, so even if this is violated, there is not a big issue. We check these associations using residual plots in Figure ???. Indeed, the distribution of residuals looks nearly equal between treatment groups in both models. The distribution of residuals varies a bit across sites, but they all appear roughly centered around 0. Sites 4 and 5 have larger variance in the differenced model. As mentioned, this should not be an issue since we permute treatments within sites, independently across sites. (In general, when treatment is not randomly assigned, this condition is necessary for the permutation test to be valid.)

```
# method 3 : freedman lane perm residuals

lm1_no_tr <- lm(Outcome ~ Baseline + factor(SITEID), data = daily_hrdq)
lm1_resid <- residuals(lm1_no_tr)
lm1_yhat <- fitted(lm1_no_tr)
observed_t1 <- summary(lm1)[["coefficients"]][["tr", "t value"]]
lm1_t_distr <- replicate(10000, {
  lm1_resid_perm <- permute_within_groups(lm1_resid, daily_hrdq$SITEID)
  daily_hrdq$response_fl <- lm1_yhat + lm1_resid_perm
  lm1_perm <- lm(response_fl ~ Baseline + tr + factor(SITEID),
```

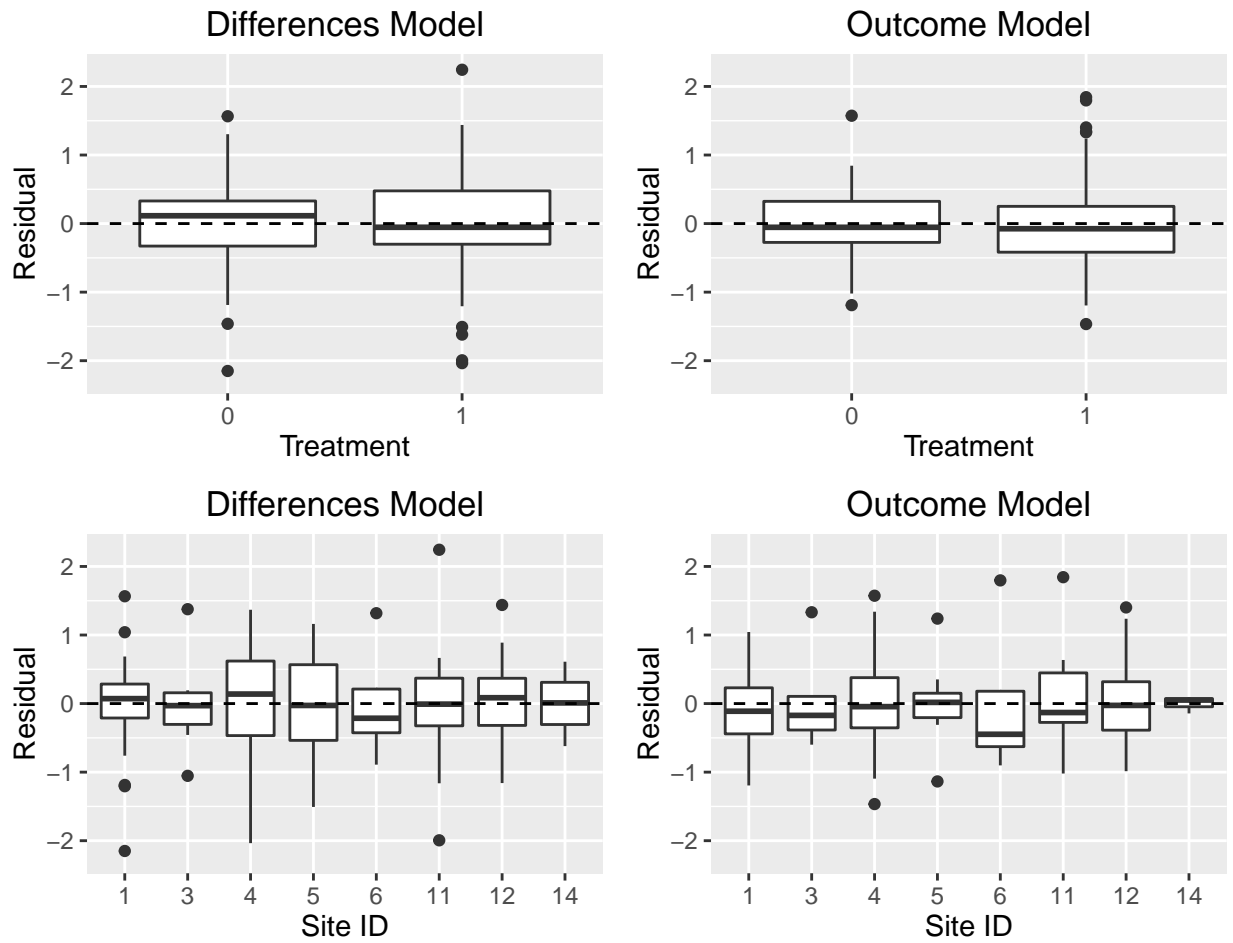


Figure 7: Residual plots of the linear regression of daily_hrdq on treatment and covariates.

```

    data = daily_hrdq)
    summary(lm1_perm)[["coefficients"]][["tr", "t value"]
  })

fl_pvalue_1 <- t2p(observed_t1, lm1_t_distr, alternative = "two-sided")

lm2_no_tr <- lm(difference ~ factor(SITEID), data = daily_hrdq)
lm2_resid <- residuals(lm2_no_tr)
lm2_yhat <- fitted(lm2_no_tr)
observed_t2 <- summary(lm2)[["coefficients"]][["tr", "t value"]]
lm2_t_distr <- replicate(10000, {
  lm2_resid_perm <- permute_within_groups(lm2_resid, daily_hrdq$SITEID)
  daily_hrdq$response_fl <- lm2_yhat + lm2_resid_perm
  lm2_perm <- lm(response_fl ~ tr + factor(SITEID), data = daily_hrdq)
  summary(lm2_perm)[["coefficients"]][["tr", "t value"]]
})

fl_pvalue_2 <- t2p(observed_t2, lm2_t_distr, alternative = "two-sided")

data.frame(perm = c(lm1_t_distr, lm2_t_distr), model = c(rep("Outcome",
  length(lm1_t_distr)), rep("Differenced", length(lm2_t_distr))),
  xintercept = c(rep(observed_t1, length(lm1_t_distr)), rep(observed_t2,
    length(lm2_t_distr)))) %>% ggplot(aes(x = perm, fill = model)) +
  geom_histogram() + facet_grid(~model) + geom_vline(aes(xintercept = xintercept))

```

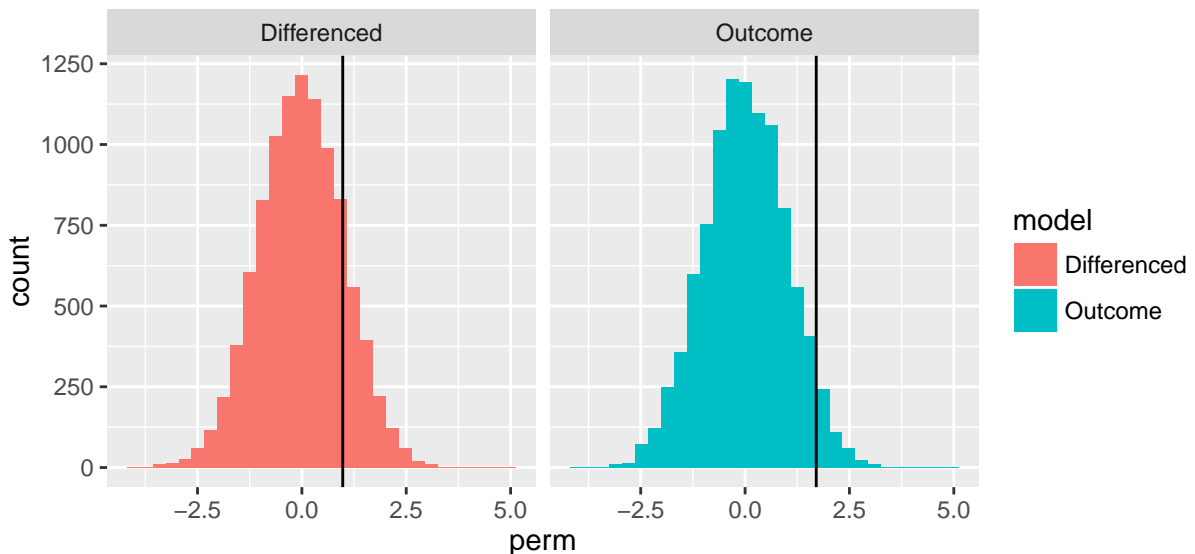


Figure 8: Permutation distribution of the Freedman-Lane tests of correlation between `daily_hrdq` and treatment after controlling for covariates.

Table ?? shows the p-values for each of the four methods, using the two models (treatment controlling for baseline, and the difference of treatment minus baseline). For the models using the treatment measure as the dependent variable, the effect of `daily_hrdq` is significant at the 0.05 level in two of the five models and is significant at the 0.1 level in four of the five models. It is never significant in the models using the difference

from baseline to treatment. As expected, the stratified permutation test using the sum of differences in means across sites has low power. We restrict our attention to the model using treatment measurement and controlling for the baseline, as this is what RB's original analysis does. Now, we run this procedure for all continuous variables. Table ?? shows the results.

```
outcome_pvalues <- c(summary(aov(lm1))[[1]]["tr", "Pr(>F)"],
  diff_means_pvalue1, diff_means_bystrata_pvalue1, lm_pvalue_1,
  fl_pvalue_1)
differenced_pvalues <- c(summary(aov(lm2))[[1]]["tr", "Pr(>F)"],
  diff_means_pvalue2, diff_means_bystrata_pvalue2, lm_pvalue_2,
  fl_pvalue_2)
pvalues_table <- cbind(differenced_pvalues, outcome_pvalues)

tests <- c("Parametric ANOVA", "Unadjusted permutation", "Unadjusted permutation (summed across strata)",
  "Linear regression permutation", "Residual permutation")
rownames(pvalues_table) <- tests
colnames(pvalues_table) <- c("Differences", "Outcome")

print(xtable(pvalues_table, digits = 3, caption = "Comparison of p-values for two measures (average outcome and difference of average outcome)",
  include.rownames = TRUE))
```

	Differences	Outcome
Parametric ANOVA	0.194	0.043
Unadjusted permutation	0.191	0.031
Unadjusted permutation (summed across strata)	0.750	0.279
Linear regression permutation	0.330	0.090
Residual permutation	0.334	0.090

Table 4: Comparison of p-values for two measures (average outcome during treatment vs. difference of average outcome and average baseline) of the primary endpoint.

```
set.seed(919547773) # Generated on Random.org Timestamp: 2016-11-09 15:39:29 UTC

continuous_vars <- c("heart_freq", "regurg_freq", "dysp_freq",
  "daily_heart", "daily_regurg", "daily_hrdq", "daily_dysp")

tests <- c("Parametric ANOVA", "Unadjusted permutation", "Linear regression permutation",
  "Residual permutation")
pvalues_table_contin <- as.data.frame(matrix(NA, nrow = length(continuous_vars),
  ncol = 4))
i <- 0

for (col in continuous_vars) {
  i <- i + 1
  tmpdata <- reshape_ga_continuous(col)
  lm1 <- lm(Outcome ~ Baseline + tr + factor(SITEID), data = tmpdata)
  pvalues_table_contin[i, 1] <- summary(aov(lm1))[[1]]["tr",
    "Pr(>F)"]

  observed_diff_means <- mean(tmpdata[tmpdata$tr == 1, ]$Outcome) -
    mean(tmpdata[tmpdata$tr == 0, ]$Outcome)
  diff_means_distr <- stratified_two_sample(group = tmpdata$tr,
    response = tmpdata$Outcome, stratum = tmpdata$SITEID,
```

```

    reps = 1000)
pvalues_table_contin[i, 2] <- t2p(observed_diff_means, diff_means_distr,
  alternative = "two-sided")

observed_t <- summary(lm1)[["coefficients"]][["tr", "t value"]]
lm1_t_distr <- replicate(1000, {
  tmpdata$tr_perm <- permute_within_groups(tmpdata$tr,
    tmpdata$SITEID)
  lm1_perm <- lm(Outcome ~ Baseline + tr_perm + factor(SITEID),
    data = tmpdata)
  summary(lm1_perm)[["coefficients"]][["tr_perm", "t value"]]
})
pvalues_table_contin[i, 3] <- t2p(observed_t, lm1_t_distr,
  alternative = "two-sided")

lm_no_tr <- lm(Outcome ~ Baseline + factor(SITEID), data = tmpdata)
lm_resid <- residuals(lm_no_tr)
lm_yhat <- fitted(lm_no_tr)
observed_t <- summary(lm1)[["coefficients"]][["tr", "t value"]]
lm_t_distr <- replicate(1000, {
  lm_resid_perm <- permute_within_groups(lm_resid, tmpdata$SITEID)
  tmpdata$response_fl <- lm_yhat + lm_resid_perm
  lm_perm <- lm(response_fl ~ Baseline + tr + factor(SITEID),
    data = tmpdata)
  summary(lm_perm)[["coefficients"]][["tr", "t value"]]
})

pvalues_table_contin[i, 4] <- t2p(observed_t, lm_t_distr,
  alternative = "two-sided")
}

rownames(pvalues_table_contin) <- continuous_vars
colnames(pvalues_table_contin) <- tests

print(xtable(pvalues_table_contin, digits = 3, align = paste0(c("r|",
  rep("p{1in}", ncol(pvalues_table_contin))), collapse = ""),
  caption = "Comparison of p-values from four tests, for each continuous endpoint."),
  include.rownames = TRUE)

```

	Parametric ANOVA	Unadjusted per- mutation	Linear regression permutation	Residual permu- tation
heart_freq	0.035	0.006	0.080	0.082
regurg_freq	0.136	0.118	0.280	0.220
dysp_freq	0.565	0.948	0.616	0.592
daily_heart	0.032	0.004	0.056	0.068
daily_regurg	0.142	0.174	0.286	0.246
daily_hrdq	0.043	0.012	0.088	0.098
daily_dysp	0.582	0.810	0.756	0.722

Table 5: Comparison of p-values from four tests, for each continuous endpoint.

Analysis - Ordinal variables

For ordinal variables, it doesn't make sense to do ANOVA. Instead, we do a multi-aspect analysis to test whether there is any difference in the outcome distributions between the drug and placebo groups. For this, we use the same stratified permutation test approach as above. However, we use three different test statistics and combine them into a single, global test using the nonparametric combination (NPC) method. The test statistics are the difference in mean outcomes between the drug and placebo group, the Anderson-Darling statistic, and the Kolmogorov-Smirnov statistic. We use the Tippett combining function.

The outcome measures used are the difference in median measures during the treatment week minus the median measures during the baseline week. Recall from Figure ?? that it is not clear whether the distributions are different. Table ?? shows the p-values for the three partial tests plus the global p-value. None is significant. It appears that treating the variables as ordinal results in a loss of information and, consequently, power.

```
# Compute difference in mean value (median outcome) between
# treatment and placebo
mean_median <- function(outcome, treatment) {
  tr_levels <- sort(unique(treatment))
  mean(outcome[treatment == tr_levels[2]], na.rm = TRUE) -
    mean(outcome[treatment == tr_levels[1]], na.rm = TRUE)
}

# Anderson-Darling
ad <- function(outcome, treatment) {
  n <- length(treatment)
  W <- table(outcome, treatment)
  ecdf <- apply(W, 2, cumsum)
  ecdf_comb <- apply(ecdf, 1, sum)

  ratio <- ecdf[, 2]/(ecdf_comb * (n - ecdf_comb))
  j <- length(ratio)
  T_ad <- sum(ratio[seq_len(j - 1)])
  return(T_ad)
}

# Kolmogorov-Smirnov
ks <- function(outcome, treatment) {
  tr_levels <- sort(unique(treatment))
  # Assume that larger treatment level is active treatment,
  # smaller other is control
  g1 <- outcome[treatment == tr_levels[2]]
  g0 <- outcome[treatment == tr_levels[1]]
  suppressWarnings(ks.test(g1, g0)$statistic)
}

multiaspect_test <- function(outcome, treatment, strata, reps = 1000) {
  obs_ts <- c(mean_median(outcome, treatment), ad(outcome,
    treatment), ks(outcome, treatment))
  names(obs_ts) <- c("Difference in Means", "Anderson-Darling",
    "Kolmogorov-Smirnov")
  perm_distr <- t(replicate(reps, {
    perm_treatment <- permute_within_groups(treatment, strata)
    c(mean_median(outcome, perm_treatment), ad(outcome, perm_treatment),
```



```

      ks(outcome, perm_treatment))
    })

    partial_pvalues <- sapply(seq_along(obs_ts), function(i) {
      t2p(obs_ts[i], perm_distr[, i], "two-sided")
    })
    names(partial_pvalues) <- c("Difference in Means", "Anderson-Darling",
      "Kolmogorov-Smirnov")
    global_pvalue <- npc(obs_ts, perm_distr, alternatives = "two-sided",
      combine = "tippett")

    return(list(`Partial test statistics` = obs_ts, `Partial p-values` = partial_pvalues,
      `Global p-value` = global_pvalue))
  }

```

```

set.seed(609726531) # Generated from Random.org Timestamp: 2016-11-09 15:38:50 UTC
ordinal_vars <- c("heart_sev_f", "regurg_sev_f", "dysp_sev_f",
  "heart_freq_f", "regurg_freq_f", "dysp_freq_f")

pvalues_table_ord <- as.data.frame(matrix(NA, nrow = length(ordinal_vars),
  ncol = 4))
i <- 0

for (col in ordinal_vars) {
  i <- i + 1
  tmpdata <- reshape_ga_ordinal(col)
  res <- multiaspect_test(outcome = tmpdata$difference, treatment = tmpdata$str,
    strata = tmpdata$SITEID, reps = 1000)
  pvalues_table_ord[i, 1] <- res$`Partial p-values`[1]
  pvalues_table_ord[i, 2] <- res$`Partial p-values`[2]
  pvalues_table_ord[i, 3] <- res$`Partial p-values`[3]
  pvalues_table_ord[i, 4] <- res$`Global p-value`
}

```

```

rownames(pvalues_table_ord) <- ordinal_vars
tests <- c("Difference in Means", "Anderson-Darling", "Kolmogorov-Smirnov",
  "Global P-value")
colnames(pvalues_table_ord) <- tests

print(xtable(pvalues_table_ord, digits = 3, align = paste0(c("r|",
  rep("p{0.75in}", ncol(pvalues_table_ord))), collapse = ""),
  caption = "Comparison of p-values from three tests plus a global p-value, for each ordinal endpoint",
  include.rownames = TRUE))

```

	Difference in Means	Anderson- Darling	Kolmogorov- Smirnov	Global P-value
heart_sev_f	0.180	0.188	0.302	0.340
regurg_sev_f	0.952	0.840	0.596	0.859
dysp_sev_f	0.612	0.746	0.252	0.436
heart_freq_f	0.424	0.466	0.124	0.237
regurg_freq_f	0.432	0.756	0.302	0.563
dysp_freq_f	0.316	0.314	0.378	0.537

Table 6: Comparison of p-values from three tests plus a global p-value, for each ordinal endpoint.