ANOVA and Regression

Justin Patterson

Contents

ANOVA Tutorial

```
library(cellWise)
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=50),tidy=TRUE)
```

Step 1: Make up Data

```
# dataset1
```

Checking the Assumptions

After running your ANOVA, check that the assumptions about the errors are met so that you can do statistical inference. Those assumptions are:

1. $\mathrm{E}(\epsilon_{ij}) = 0,\ \mathrm{Var}(\epsilon_{ij}) = \sigma_i^2 < \infty,$ for all i, j.
2. The ϵ_{ij} are mutually independent and normally distributed.
3. $\sigma_i^2 = \sigma^2$ for all i.

Checking Assumption 1

Assumption 1 was violated.

Checking Assumption 2

Assumption 2 was violated.

Checking Assumption 3

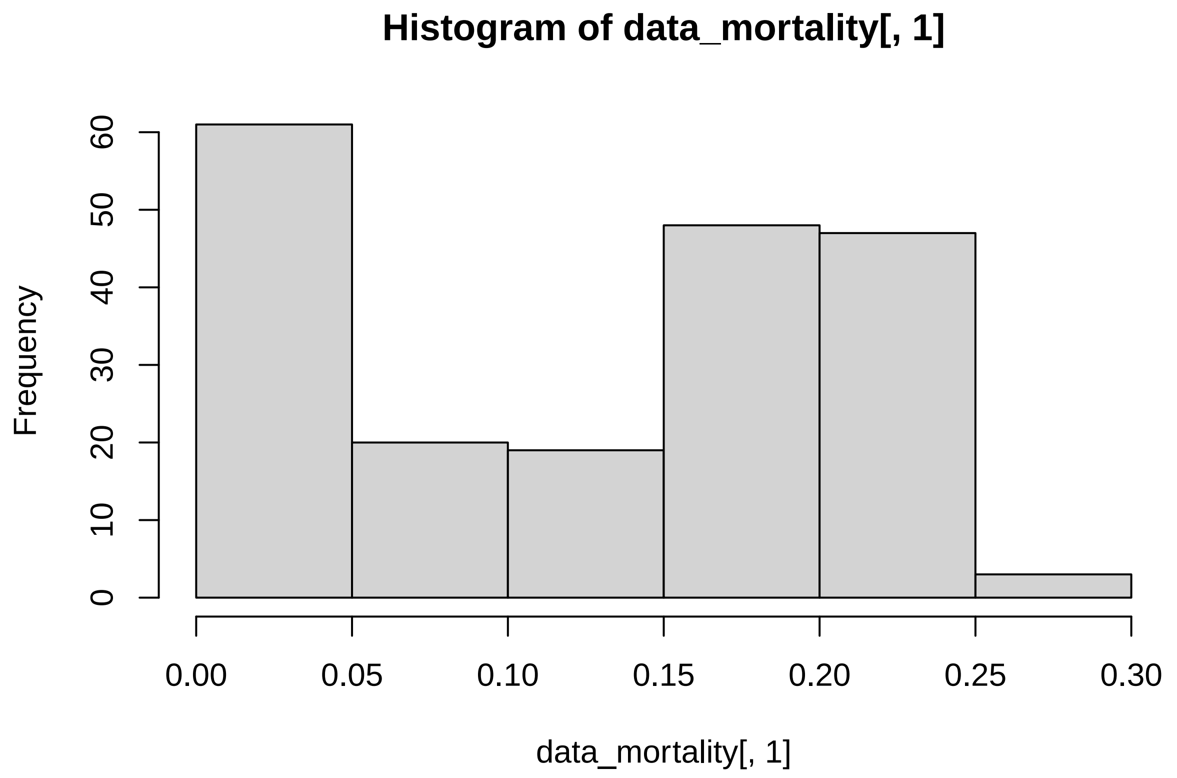
Assumption 3 was violated.

A variance-stabilizing transformation of the response variable may help.

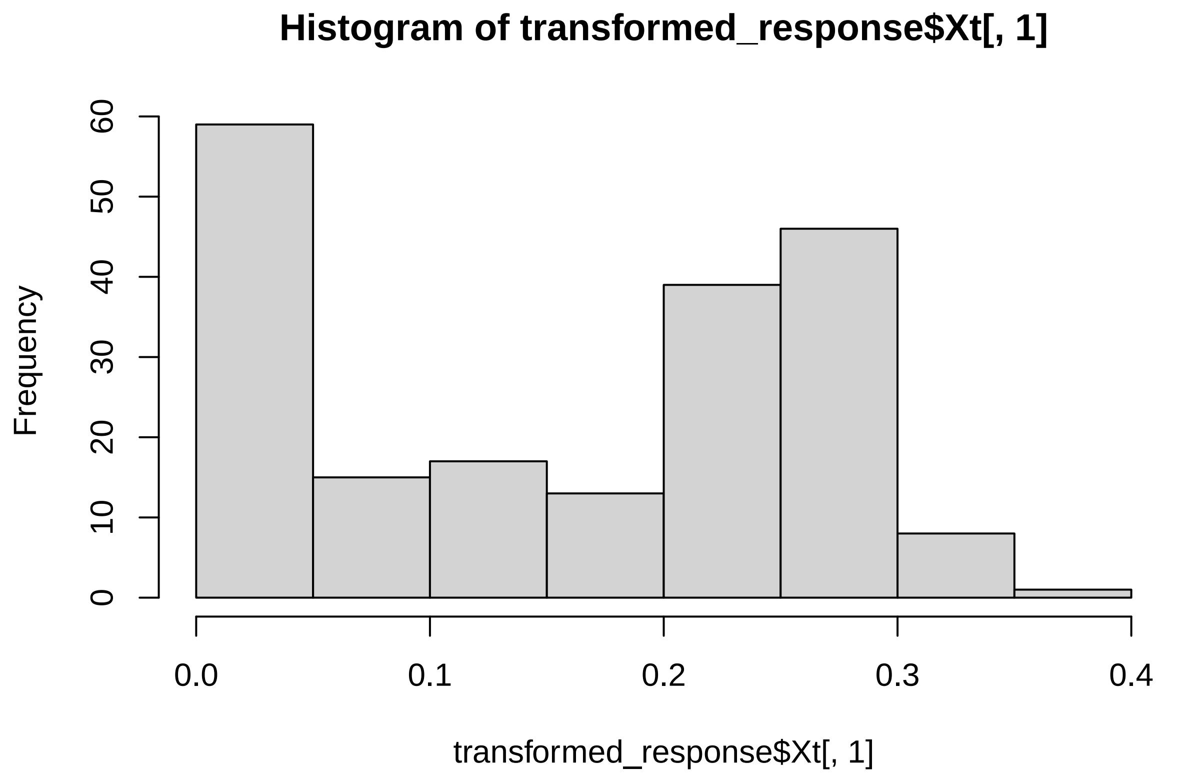
```
data("data_mortality")
transformed_response = transfo(data_mortality, prestandardize = FALSE)
```

```
##
##  The input data has 198 rows and 91 columns.
```

```
hist(data_mortality[, 1])
```



```
hist(transformed_response$Xt[, 1])
```



```
shapiro.test(data_mortality[, 1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_mortality[, 1]
## W = 0.86877, p-value = 4.552e-12
```

```
shapiro.test(transformed_response$Xt[, 1])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  transformed_response$Xt[, 1]
## W = 0.88041, p-value = 1.968e-11
```

Regression and Categorical Variables

```
library(tidymodels)
```

```
## Registered S3 method overwritten by 'tune':
##   method                   from
##   required_pkgs.model_spec parsnip

## -- Attaching packages -------------------------------------- tidymodels 0.1.3 --

## v broom        0.7.6      v recipes      0.1.16
## v dials        0.0.9      v rsample      0.1.0
## v dplyr        1.0.6      v tibble       3.1.2
## v ggplot2      3.3.3      v tidyr        1.1.3
## v infer        0.5.4      v tune         0.1.5
## v modeldata    0.1.0      v workflows    0.2.2
## v parsnip      0.1.6      v workflowsets 0.0.2
## v purrr        0.3.4      v yardstick    0.0.8

## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Use tidymodels_prefer() to resolve common conflicts.
```

```
library(ggplot2)
```

There is a profound connection between linear regression and ANOVA. In order to see this, you have to understand that the categorical variables of an ANOVA can be coded with numbers, which allows them to be used in a linear regression model. Let us recall [@LinearMod] the multiple linear regression model.

Given a random sample of n observations $(Y_i, X_{i1}, ..., X_{ip}),\ i = 1, ..., n$, the basic multiple linear regression model is

$$Y_i = \beta_0 + \beta_1 X_{i1} + ... + \beta_p X_{ip} + \epsilon_i, \quad i = 1, ..., n$$

where each ϵ_i is a random variable with a mean of 0. In matrix form, this can be written as

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & X_{1,1} & X_{1,2} & \dots & X_{1,p} \\ 1 & X_{2,1} & X_{2,2} & \dots & X_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n,1} & X_{n,2} & \dots & X_{n,p} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_0 \\ \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Here, the $X_{i,j}$ represent our coded categorical variables. These categorical variables are coded according to the hypotheses of interest. In many cases, the coding is done so that the newly coded variables are contrasts of the old categorical variables.

A contrast is a linear combination of variables such that the coefficients sum to 0.

$$\sum_i a_i\theta_i \quad \text{such that} \quad \sum_i a_i = 0$$

Unlike in ANOVA, in regression, it is best to use coding schemes based on orthogonal and fractional contrasts. Orthogonal contrasts are a set of contrasts in which, for any distinct pair, the sum of the cross-products of the coefficients is 0.

$$\sum_i a_i b_i = 0$$

I believe that a fractional contrast is such that

$$\sum_i |a_i| = 2$$

Categorical variable coding schemes can be easily expressed in a matrix format. The convention is to have the old categorical variables as the row headers and the newly coded variables as the column headers. In such a matrix, the $[c_{ij}]$ entry indicates the value of the j^{th} level of the new variable for the i^{th} level of the old variable. Here is an example of such a matrix constructed using orthogonal and fractional contrasts.

```r
(contr_mat = matrix(data = c(1, 0, -1, 0.5, -1, 0.5),
    nrow = 3, ncol = 2))
```

```
##      [,1] [,2]
## [1,]    1  0.5
## [2,]    0 -1.0
## [3,]   -1  0.5
```

Interpreting this coding scheme in the context of our linear model, we see that

$$\begin{aligned} E(Y_i|X_{i1} = 1, X_{i2} = \tfrac{1}{2}) &= \beta_0 + \beta_1 + \tfrac{1}{2}\beta_2 &= \mu_1 \\ E(Y_i|X_{i1} = 0, X_{i2} = -1) &= \beta_0 - \beta_2 &= \mu_2 \\ E(Y_i|X_{i1} = -1, X_{i2} = \tfrac{1}{2}) &= \beta_0 - \beta_1 + \tfrac{1}{2}\beta_2 &= \mu_3 \end{aligned}$$

or, in matrix format,

$$\begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 1 & 0 & -1 \\ 1 & -1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}$$

We can solve this for $\boldsymbol{\beta}$ for interpretation's sake.

```r
solve(cbind(rep(1, nrow(contr_mat)), contr_mat))
```

```
##           [,1]       [,2]       [,3]
## [1,] 0.3333333  0.3333333  0.3333333
## [2,] 0.5000000  0.0000000 -0.5000000
## [3,] 0.3333333 -0.6666667  0.3333333
```

$$\begin{aligned}
\beta_0 &= \frac{\mu_1+\mu_2+\mu_3}{3} &&= \text{grand mean response}\\
2\beta_1 &= \mu_1 - \mu_3 &&= \text{difference in the mean response between levels 1 and 3 of the old categorical variable}\\
\tfrac{3}{2}\beta_2 &= \frac{\mu_1+\mu_3}{2} - \mu_2 &&= \text{difference in the mean response between level 2 and the average of levels 1 and 3 of the old categorical variable}
\end{aligned}$$

Let's look at another contrast matrix and see if we can interpret it.

```r
contr.helmert(n = 3)
```

```
##   [,1] [,2]
## 1   -1   -1
## 2    1   -1
## 3    0    2
```

```r
solve(cbind(rep(1, 3), contr.helmert(n = 3)))
```

```
##               1          2         3
## [1,]  0.3333333  0.3333333 0.3333333
## [2,] -0.5000000  0.5000000 0.0000000
## [3,] -0.1666667 -0.1666667 0.3333333
```

$$\begin{aligned}
\beta_0 &= \frac{\mu_1+\mu_2+\mu_3}{3} &&= \text{grand mean response}\\
2\beta_1 &= \mu_2 - \mu_1 &&= \text{difference in the mean response between levels 2 \& 1 of the old categorical variable}\\
3\beta_2 &= \mu_3 - \frac{\mu_1+\mu_2}{2} &&= \text{difference in the mean response between level 3 and the average of levels 1 and 2 of the old categorical variable}
\end{aligned}$$

Perhaps you have heard of polynomial regression? Polynomial regression is just a special case of linear regression in a different basis. In polynomial regression, (just like multiple linear regression) if you use all of your explanatory variables, then you will likely get multi-collinearity problems.

```r
contr.poly(n = 3)
```

```
##                 .L         .Q
## [1,] -7.071068e-01  0.4082483
## [2,] -7.850462e-17 -0.8164966
## [3,]  7.071068e-01  0.4082483
```

```r
(A = solve(cbind(rep(1, 3), contr.poly(n = 3))))
```

```
##          [,1]       [,2]      [,3]
##     0.3333333  0.3333333 0.3333333
## .L -0.7071068  0.0000000 0.7071068
## .Q  0.4082483 -0.8164966 0.4082483
```

The first matrix shows how to code the levels of your categorical variable and the second matrix is used for interpretation.

$$\begin{aligned}
\beta_0 &= \frac{\mu_1+\mu_2+\mu_3}{3} &&= \text{grand mean response}\\
\beta_1 &= -0.707\mu_1 + 0.707\mu_3 &&= \text{measure of a linear trend in the mean response}\\
\beta_2 &= 0.408\mu_3 - 0.816\mu_2 + 0.408\mu_3 &&= \text{measure of a quadratic trend in the mean response}
\end{aligned}$$

For example, we can test whether the difference between the means from two populations are equal by doing a linear regression or an ANOVA.

Let's make up some data and try it!

```r
source(file.path("src", "fabricate.R"))

design = data.frame(group = c(0, 1), n = c(10, 10))
```

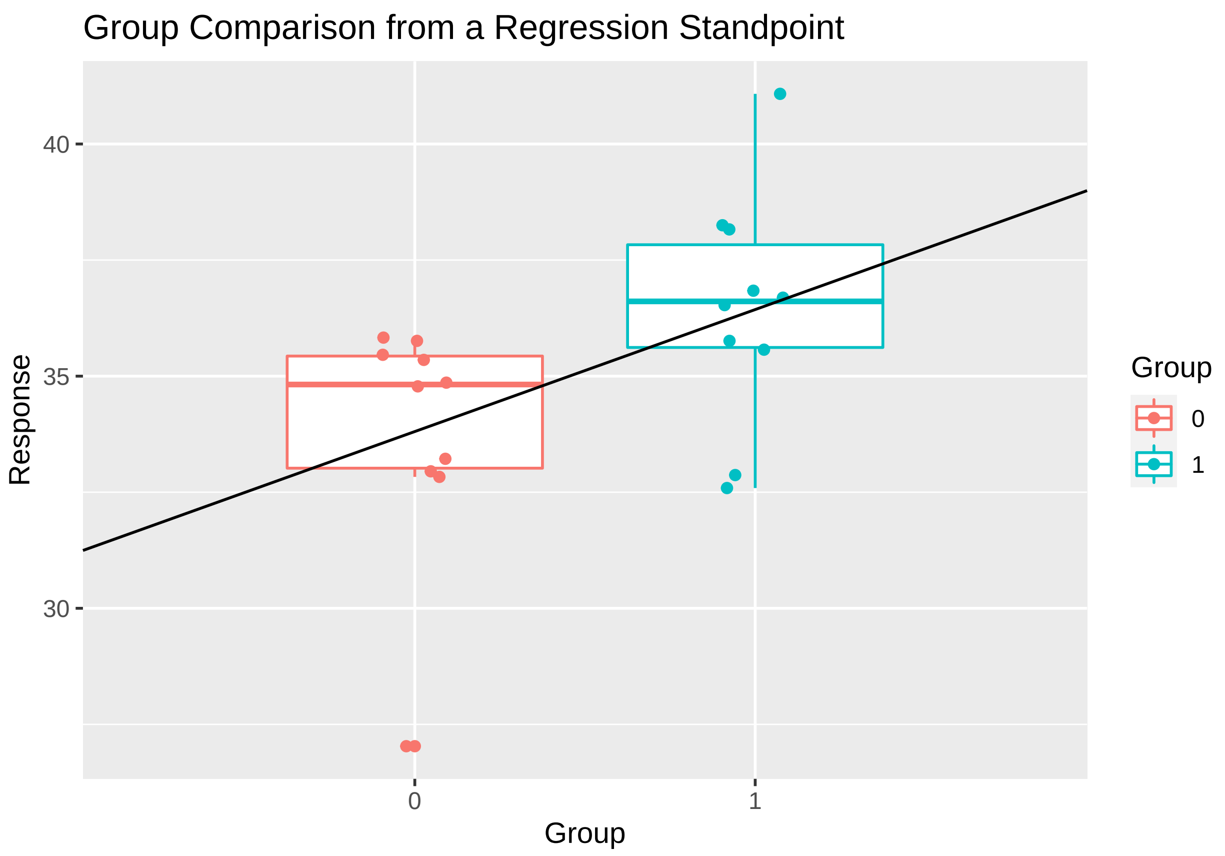
```
data1 = fabricate(flr = design)
```

Let's check out our data.

```
# Make a linear model
data1_lm_independent_samples = lm(response ~ group,
    data = data1)

# plot
ggplot(data = data1, aes(x = group, y = response, color = factor(group))) +
    geom_boxplot() + geom_jitter(height = 0, width = 0.1) +
    geom_abline(intercept = data1_lm_independent_samples$coefficients[1],
        slope = data1_lm_independent_samples$coefficients[2]) +
    labs(title = "Group Comparison from a Regression Standpoint",
        color = "Group", x = "Group", y = "Response") +
    scale_x_discrete(limits = c(0, 1))
```

```
## Warning: Continuous limits supplied to discrete scale.
## Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```



The way you code your categorical variables in a linear model is extremely important. Different codings lead to different interpretations of the parameters (betas) in your model. For us, our model is

$$Y_i = \beta_0 + \beta_{i1} X_{i1} + \epsilon_i$$

From this, we have

$$E(Y_i | X_{i1} = 0) = \beta_0$$
$$E(Y_i | X_{i1} = 1) = \beta_0 + \beta_1$$

From which we can derive,

$$\beta_1 = E(Y_i|X_{i1} = 1) - E(Y_i|X_{i1} = 0)$$

So, our slope estimate is the estimated amount by which the mean of group1 is above that of the mean of group0.

Run linear regression

```
summary(data1_lm_independent_samples)
```

```
##
## Call:
## lm(formula = response ~ group, data = data1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7770 -0.8587  0.3310  1.6712  4.6460
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.8070     0.8159  41.436   <2e-16 ***
## group         2.6270     1.1538   2.277   0.0352 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.58 on 18 degrees of freedom
## Multiple R-squared:  0.2236, Adjusted R-squared:  0.1805
## F-statistic: 5.184 on 1 and 18 DF,  p-value: 0.03524
```

Run ANOVA

```
data1$group = as.factor(data1$group)
data1_ANOVA_independent_samples = aov(response ~ group,
    data = data1)
summary(data1_ANOVA_independent_samples)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## group        1  34.51   34.51   5.184 0.0352 *
## Residuals   18 119.82    6.66
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Run t-Test

```
(data1_t_test_independent_samples = t.test(x = data1[data1$group ==
    1, "response"], y = data1[data1$group == 0, "response"],
    paired = FALSE, var.equal = TRUE))
```

```
##
##  Two Sample t-test
##
## data:  data1[data1$group == 1, "response"] and data1[data1$group == 0, "response"]
## t = 2.2768, df = 18, p-value = 0.03524
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2029061 5.0510939
## sample estimates:
## mean of x mean of y
##    36.434    33.807
```

Notice the similarities.

```
# Confidence interval for the difference in the
# means
confint(data1_lm_independent_samples, "group", level = 0.95)
```

```
##           2.5 %   97.5 %
## group 0.2029061 5.051094
```

```
data1_t_test_independent_samples$conf.int
```

```
## [1] 0.2029061 5.0510939
## attr(,"conf.level")
## [1] 0.95
```

```
# p-values
with(summary(data1_lm_independent_samples), unname(pf(fstatistic[1],
    fstatistic[2], fstatistic[3], lower.tail = F)))
```

```
## [1] 0.03524354
```

```
summary(data1_ANOVA_independent_samples)[[1]][[1, 5]]
```

```
## [1] 0.03524354
```

```
data1_t_test_independent_samples$p.value
```

```
## [1] 0.03524354
```

Now, let's look at something else. The CO2 data frame has 84 rows and 5 columns of data from an experiment on the cold tolerance of the grass species Echinochloa crus-galli.

```
data("CO2")

CO2[sample(nrow(CO2), size = 5), ]
```

```
##    Plant        Type  Treatment conc uptake
## 70   Mc1 Mississippi    chilled 1000   21.9
## 84   Mc3 Mississippi    chilled 1000   19.9
## 75   Mc2 Mississippi    chilled  500   12.5
## 81   Mc3 Mississippi    chilled  350   17.9
## 13   Qn2      Quebec nonchilled  675   41.4
```

What is a linear model? In the context of linear regression, a linear model is a relationship between the responses and the explanatory variables that is linear in the parameters.

```
CO2_recipe = recipe(uptake ~ ., data = CO2) %>% step_dummy(c("Type",
    "Treatment"))
# see contrasts() function

CO2_linear_model = linear_reg() %>% set_engine("lm",
    contrasts = list(Plant = "contr.poly"))

CO2_workflow = workflow() %>% add_model(CO2_linear_model) %>%
    add_recipe(CO2_recipe)
```

```
CO2_fit = CO2_workflow %>% fit(data = CO2)
```

```
CO2_fit %>% pull_workflow_fit() %>% tidy()
```

```
## # A tibble: 15 x 5
##    term              estimate std.error statistic   p.value
##    <chr>                <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)        19.5      1.17      16.7     2.96e-26
##  2 Plant.L           -22.9      2.27     -10.1     2.17e-15
##  3 Plant.Q            -4.62     2.27      -2.03    4.57e- 2
##  4 Plant.C             4.67     2.27       2.06    4.34e- 2
##  5 Plant^4             2.34     2.27       1.03    3.06e- 1
##  6 Plant^5             4.31     2.27       1.90    6.13e- 2
##  7 Plant^6            -0.0390   2.27      -0.0172  9.86e- 1
##  8 Plant^7            -2.04     2.27      -0.897   3.73e- 1
##  9 Plant^8            -3.28     2.27      -1.44    1.53e- 1
## 10 Plant^9            -9.07     2.27      -4.00    1.56e- 4
## 11 Plant^10            0.546    2.27       0.241   8.10e- 1
## 12 Plant^11            1.91     2.27       0.843   4.02e- 1
## 13 conc                0.0177   0.00223    7.96    1.97e-11
## 14 Type_Mississippi   NA       NA         NA      NA
## 15 Treatment_chilled  NA       NA         NA      NA
```