The `contrast` Package

Max Kuhn, Steve Weston, Jed Wing, James Forester

May 23, 2008

1 Introduction

The purpose of the `contrast` package is to provide a standardized interface for testing linear combinations of parameters from common regression models. The syntax mimics the `contrast.Design` function from the `Design` library. The contrast class has been extended in this package to linear models produced using the functions `lm`, `glm`, `gls`, `lme` and `geese`. Other `R` functions with similar purposes exist in R, but the interfaces are different and many require the user to specify the contrast in terms of the parameter contrast coefficient vector. This package aims to simplify the process for the user.

2 Contrasts

First, some notation:

$$\begin{aligned} n &= \text{number of samples} \\ p &= \text{number of model parameters associated with fixed effects (excluding the intercept)} \\ q &= \text{number of covariance parameters with random effects or correlations} \\ Y &= n \times 1 \text{ response vector} \\ X &= n \times (p+1) \text{ model matrix} \\ \beta &= \text{model parameters associated with fixed effects} \\ \Sigma &= \text{covariance matrix associated with the fixed effects} \end{aligned} \tag{1}$$

This package uses one degree of freedom Wald tests to calculate p–values for linear combinations of parameters. For example, the basic linear model is of the form $y = X\beta + \epsilon$, where the individual

errors are assumed to be iid $N(0, \sigma^2)$. Ordinary least squares provides us with estimates $\hat{\beta}$, $\hat{\sigma}^2$ and $\hat{\Sigma}$. Given a $(p+1) \times 1$ vector of constants, c, we can estimate a linear combination of parameters $\lambda = c'\beta$ by substituting the estimated parameter vectors: $\hat{\lambda} = c'\hat{\beta}$. Using basic linear algebra, $Var[\lambda] = c'\Sigma c$. The statistic generated for contrasts is

$$S = \frac{c'\lambda}{c'\Sigma c} \tag{2}$$

For linear models with normal errors, $S \sim T_{n-p-1}$ and there is no uncertainty about the distribution of the test statistic and the degrees of freedom. In other cases, this is not true. Asymptotics come into play for several models and there is some ambiguity as to whether a t or normal distribution should be used to compute p–values (See Harrell, 2001, Section 9,2 for a discussion). We follow the conventions of each package: `glm`, `gls` and `lme` models use a t distribution and a normal distribution is used for `gee` models. For models where there are extra covariance or correlation parameters, we again follow the lead of the package. For `gls` model, the degrees of freedom are $n - p$, while in `lme` models, it is $n - p - q$.

The remainder of this document shows two examples and how the `contrast` function can be applied to different models.

2.1 Linear Models

As an example, a gene expression experiment was run to assess the effect of a compound under two different diets: high fat and low fat. The main comparisons of interest are the difference between the treated and untreated groups within a diet. The interaction effect was a secondary hypothesis. For illustration, we only include the expression value of one of the genes.

A summary of the design is given in Table 1.

Table 1: A summary of the diet experimental design

Diet	Group	Freq
high fat	treatment	6
low fat	treatment	6
high fat	vehicle	6
low fat	vehicle	6

The study design was a two–way factorial with $n = 24$. The cell means can be labeled as in Table 2.

The reference cell used by `R` is cell D, the treated samples on a high fat diet.

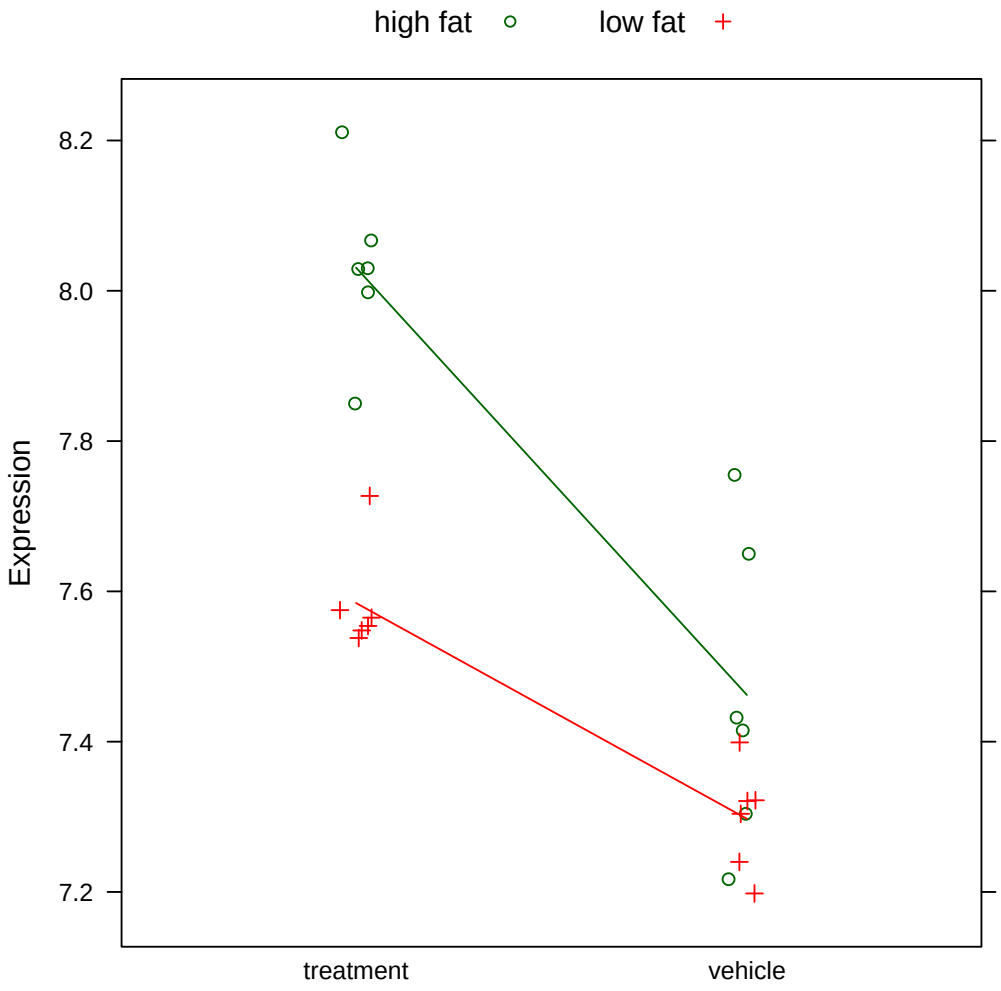

Figure 1: An interaction plot for the diet gene expression experiment

Table 2: The cells of the diet experimental design

	Diet	
	Low Fat	High Fat
Vehicle	A	B
Compound	C	D

The model used is

$$\begin{aligned} \log \text{Expression}_2 = \beta_0 & \\ & + \beta_1 \text{Vehicle Group} \\ & + \beta_2 \text{Low Fat Diet} \\ & + \beta_3 \text{Low Fat Diet and Vehicle Group} \end{aligned} \tag{3}$$

so that $p = 3$. Substituting the appropriate coefficients into each cell produces the parameters in Table 3.

Table 3: The parameter structure of the diet experimental design

	Diet	
	Low Fat	High Fat
Vehicle	$\beta_0 + \beta_1 + \beta_2 + \beta_3$	$\beta_0 + \beta_1$
Compound	$\beta_0 + \beta_2$	β_0

This means that

- β_2 tests for diet effect in the treated samples $(C - D)$
- β_1 tests for a compounds effect in the high fat diet samples $(B - D)$

Fitting the model specified by (3) using `lm`:

```
> lmFit1 <- lm(expression ~ (group + diet)^2, data = example1)
> summary(lmFit1)

Call:
lm(formula = expression ~ (group + diet)^2, data = example1)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.24517 -0.04667 -0.01450  0.02754  0.29283

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                8.03083    0.05218 153.903  < 2e-16 ***
groupvehicle              -0.56867    0.07380  -7.706 2.07e-07 ***
dietlow fat               -0.44633    0.07380  -6.048 6.52e-06 ***
groupvehicle:dietlow fat   0.28150    0.10436   2.697   0.0139 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 0.1278 on 20 degrees of freedom
Multiple R-squared: 0.8447,        Adjusted R-squared: 0.8215
F-statistic: 36.27 on 3 and 20 DF,  p-value: 2.79e-08
```

To test the treatment effect in the high fat diet, $D - B = -\beta_1$. This coefficient and hypothesis test for the difference between treated and un–treated in the high fat diet group is in the row labeled as `groupvehicle` in the output of `summary.lm`.

To compare the compound data and the vehicle data in the low fat diet group, Tables 2 and 3 can be used:

$$\begin{aligned} C - A &= \beta_0 + \beta_2 - (\beta_0 + \beta_1 + \beta_2 + \beta_3) \\ &= -\beta_1 - \beta_3 \end{aligned}$$

This hypothesis translates to testing $\beta_1 + \beta_3 = 0$, or a contrast using $c = (0, 1, 0, 1)$. To get the results of the difference between treated and un–treated in the low fat diet group, we (finally) use the contrast function:

```
> highFatDiff <- contrast(lmFit1, list(diet = "low fat", group = "vehicle"),
+     list(diet = "low fat", group = "treatment"))
> print(highFatDiff, X = TRUE)


lm model parameter contrast

   Contrast      S.E.      Lower      Upper     t df Pr(>|t|)
 -0.2871667 0.0737955 -0.4411014 -0.1332320 -3.89 20    9e-04

Contrast coefficients:
 (Intercept) groupvehicle dietlow fat groupvehicle:dietlow fat
           0            1           0                       1
```

While the effect of treatment is significantly different when compared to vehicle for both diets, the difference is more pronounced in the high fat diet.

Alternatively, both test can be done in the same call to contrast:

```
> eachTrmtEffect <- contrast(lmFit1, list(diet = levels(example1$diet),
+     group = "vehicle"), list(diet = levels(example1$diet), group = "treatment"))
> print(eachTrmtEffect, X = TRUE)

lm model parameter contrast

   Contrast      S.E.      Lower      Upper     t df Pr(>|t|)
 -0.5686667 0.0737955 -0.7226014 -0.4147320 -7.71 20    0e+00
 -0.2871667 0.0737955 -0.4411014 -0.1332320 -3.89 20    9e-04

Contrast coefficients:
 (Intercept) groupvehicle dietlow fat groupvehicle:dietlow fat
           0            1          0                        0
           0            1          0                        1
```

Also, we can use the type argument to compute a single treatment effect averaging over the levels of the other factor:

```
> meanTrmtEffect <- contrast(lmFit1, list(diet = levels(example1$diet),
+     group = "vehicle"), list(diet = levels(example1$diet), group = "treatment"),
+     type = "average")
> print(meanTrmtEffect, X = TRUE)

lm model parameter contrast

    Contrast       S.E.      Lower      Upper    t df Pr(>|t|)
1 -0.4279167 0.05218129 -0.5367649 -0.3190684 -8.2 20        0

Contrast coefficients:
  (Intercept) groupvehicle dietlow fat groupvehicle:dietlow fat
1           0            1          0                      0.5
```

Additionally, for ordinary linear regression models, there is an option to use sandwich estimates for the covariance matrix of the parameters. See the sandwich package for more details. Going back to our comparison of treated versus control in low fat samples, we can use the HC3 estimate in the contrast.

```
> highFatDiffSAND <- contrast(lmFit1, list(diet = "low fat", group = "vehicle"),
+     list(diet = "low fat", group = "treatment"), covType = "HC3")
> print(highFatDiffSAND)
```

```
lm model parameter contrast

   Contrast       S.E.     Lower      Upper     t df Pr(>|t|)
 -0.2871667 0.04467878 -0.380365 -0.1939684 -6.43 20        0

The HC3 covariance estimator was used.
```

The t–statistic associated with the sandwich estimate is -6.427 versus -3.891 using the traditional estimate of the covariance matrix.

2.2 Generalized Linear Model

In this class of models, the distributional assumptions are expanded beyond the normal distribution to the general exponential family. Also, these models are linear in the sense that they are linear on a specified scale. The link function, denoted as η, is a function that defines how the linear predictor, $x'\beta$, enters the model. While there are several approaches to testing for statistical differences between models, such as the likelihood ratio or score tests, the Wald test is another method for assessing the statistical significance of linear combinations of model parameters. The basic Wald–type test uses the familiar statistic 2 to evaluate hypotheses. The distributional properties are exact for the normal distribution and asymptotically valid for other distributions in the exponential family. There are some issues with the Wald test (see Hauck and Donner, 1977). Whenever possible, likelihood ratio or score statistics are preferred, but these tests cannot handle some types of hypotheses, in which case the Wald test can be used.

For the previous example, it is customary to log transform gene expression data using a base of 2, we can illustrate contrasts in generalized linear models using the log (base e) link. In this case, the actual model being fit is $\exp(x'\beta)$.

```
> glmFit1 <- glm(2^expression ~ (group + diet)^2, data = example1,
+     family = gaussian(link = "log"))
> summary(glmFit1)

Call:
glm(formula = 2^expression ~ (group + diet)^2, family = gaussian(link = "log"),
    data = example1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-31.518   -6.363   -2.117    3.401   38.181

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              5.56925    0.02766 201.364  < 2e-16 ***
```

```
groupvehicle             -0.38839    0.04928  -7.882 1.47e-07 ***
dietlow fat              -0.31104    0.04680  -6.647 1.80e-06 ***
groupvehicle:dietlow fat  0.18929    0.07730   2.449   0.0237 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

(Dispersion parameter for gaussian family taken to be 315.6268)

    Null deviance: 43574.7  on 23  degrees of freedom
Residual deviance:  6312.5  on 20  degrees of freedom
AIC: 211.84

Number of Fisher Scoring iterations: 4

> highFatDiff <- contrast(glmFit1, list(diet = "low fat", group = "vehicle"),
+     list(diet = "low fat", group = "treatment"))
> print(highFatDiff, X = TRUE)

glm model parameter contrast

   Contrast       S.E.     Lower       Upper     t df Pr(>|t|)
 -0.1990949 0.05955574 -0.323326 -0.07486377 -3.34 20   0.0032

Contrast coefficients:
 (Intercept) groupvehicle dietlow fat groupvehicle:dietlow fat
           0            1           0                        1
```

The coefficients and p–values are not wildly different given that the scale is slightly different (i.e. $\log_2$ versus $\log_e$).

2.3 Generalized Least Squares

In a second gene expression example, stem cells were differentiated using a set of factors (such as media types, cell spreads etc.). These factors were collapsed into a single cell environment configurations variable. The cell lines were assays over three days. Two of the configurations were only run on the first day and the other two were assays at baseline.

To get the materials, three donors provided materials. These donors provided (almost) equal replication across the two experimental factors (day and configuration). Table 4 shows a summary of the design.

The one of the goal of this experiment was to assess pre–specified differences in the configuration at each time point. For example, the differences between configurations A and B at day one is of interest. Also, the differences between configurations C and D at each time points were important.

Table 4: A summary of the stem cell experimental design

Day	Configuration	Number of Donors
1	A	3
2	A	0
4	A	0
1	B	3
2	B	0
4	B	0
1	C	3
2	C	2
4	C	3
1	D	3
2	D	3
4	D	3

Since there are missing cells in the design, it is not a complete two–way factorial. One way to analyze this experiment is to further collapse the time and configuration data into a single variable and then specify each comparison using this factor.

For example:

```
> example2$group <- factor(paste(example2$day, ":", example2$config,
+     sep = ""))
> print(table(example2$group))

1:A 1:B 1:C 1:D 2:C 2:D 4:C 4:D
  3   3   3   3   2   3   3   3
```

Using this new factor, we fit a linear model to this one–way design. We should account for the possible within–donor correlation. A generalized least square fit can do this, where we specify a correlation structure for the residuals. A compound–symmetry (a.k.a. exchangeable) correlation structure assumes that the within–donor correlation is constant.

The mdoel fit is:

```
> glsFit <- gls(expression ~ group, data = example2, corCompSymm(form = ~1 |
+     subject))
> summary(glsFit)

Generalized least squares fit by REML
  Model: expression ~ group
```

```
  Data: example2
       AIC      BIC   logLik
  -6.19145 0.889052 13.09573

Correlation Structure: Compound symmetry
 Formula: ~1 | subject
 Parameter estimate(s):
     Rho
0.882028

Coefficients:
                Value  Std.Error  t-value p-value
(Intercept)  9.300000 0.09811971 94.78218  0.0000
group1:B     0.194667 0.04766072  4.08443  0.0010
group1:C    -0.139333 0.04766072 -2.92344  0.0105
group1:D     0.036667 0.04766072  0.76933  0.4536
group2:C     0.062477 0.05402811  1.15638  0.2656
group2:D    -0.012000 0.04766072 -0.25178  0.8046
group4:C     0.139667 0.04766072  2.93044  0.0103
group4:D     0.032000 0.04766072  0.67141  0.5122

 Correlation:
         (Intr) grp1:B grp1:C grp1:D grp2:C grp2:D grp4:C
group1:B -0.243
group1:C -0.243  0.500
group1:D -0.243  0.500  0.500
group2:C -0.214  0.441  0.441  0.441
group2:D -0.243  0.500  0.500  0.500  0.441
group4:C -0.243  0.500  0.500  0.500  0.441  0.500
group4:D -0.243  0.500  0.500  0.500  0.441  0.500  0.500

Standardized residuals:
       Min         Q1        Med         Q3        Max
-1.5808727 -0.7266915  0.4295423  0.6085851  1.1552531

Residual standard error: 0.1699483
Degrees of freedom: 23 total; 15 residual
```

In this example, $n = 23$ and $p = 8$. This model estimates the residual variance and hte within–subject correlation, so $q = 2$. The default parameter estimates compare each group to the reference cell (day 1, configuration A). The summary table provides one of the p–values that we are interested in (configuration A vs. B at day 1). An example of obtaining the other p–values is shown below:

```
> print(contrast(glsFit, list(group = "4:C"), list(group = "4:D")),
+     X = TRUE)
```

```
gls model parameter contrast

   Contrast       S.E.      Lower     Upper    t df Pr(>|t|)
1 0.1076667 0.04766072 0.01425337 0.2010800 2.26 15   0.0392

Contrast coefficients:
  (Intercept) group1:B group1:C group1:D group2:C group2:D group4:C group4:D
1           0        0        0        0        0        0        1       -1
```

2.4 Linear Mixed Models via `lme`

A similar model can be fit using a linear mixed model via the `lme` function. In this case, we can add a random intercept attributable to the donors. This can produce the above compound symmetry model, but here the within donor–correlation is constrained to be positive.

```
> lmeFit <- lme(expression ~ group, data = example2, random = ~1 |
+     subject)
> summary(lmeFit)


Linear mixed-effects model fit by REML
 Data: example2
       AIC      BIC   logLik
  -6.19145 0.889052 13.09573

Random effects:
 Formula: ~1 | subject
        (Intercept)   Residual
StdDev:   0.1596093 0.05837223

Fixed effects: expression ~ group
                Value  Std.Error DF  t-value p-value
(Intercept)  9.300000 0.09811971 13 94.78218  0.0000
group1:B     0.194667 0.04766072 13  4.08443  0.0013
group1:C    -0.139333 0.04766072 13 -2.92344  0.0119
group1:D     0.036667 0.04766072 13  0.76933  0.4555
group2:C     0.062477 0.05402811 13  1.15638  0.2683
group2:D    -0.012000 0.04766072 13 -0.25178  0.8051
group4:C     0.139667 0.04766072 13  2.93044  0.0117
group4:D     0.032000 0.04766072 13  0.67141  0.5137
 Correlation:
         (Intr) grp1:B grp1:C grp1:D grp2:C grp2:D grp4:C
group1:B -0.243
```

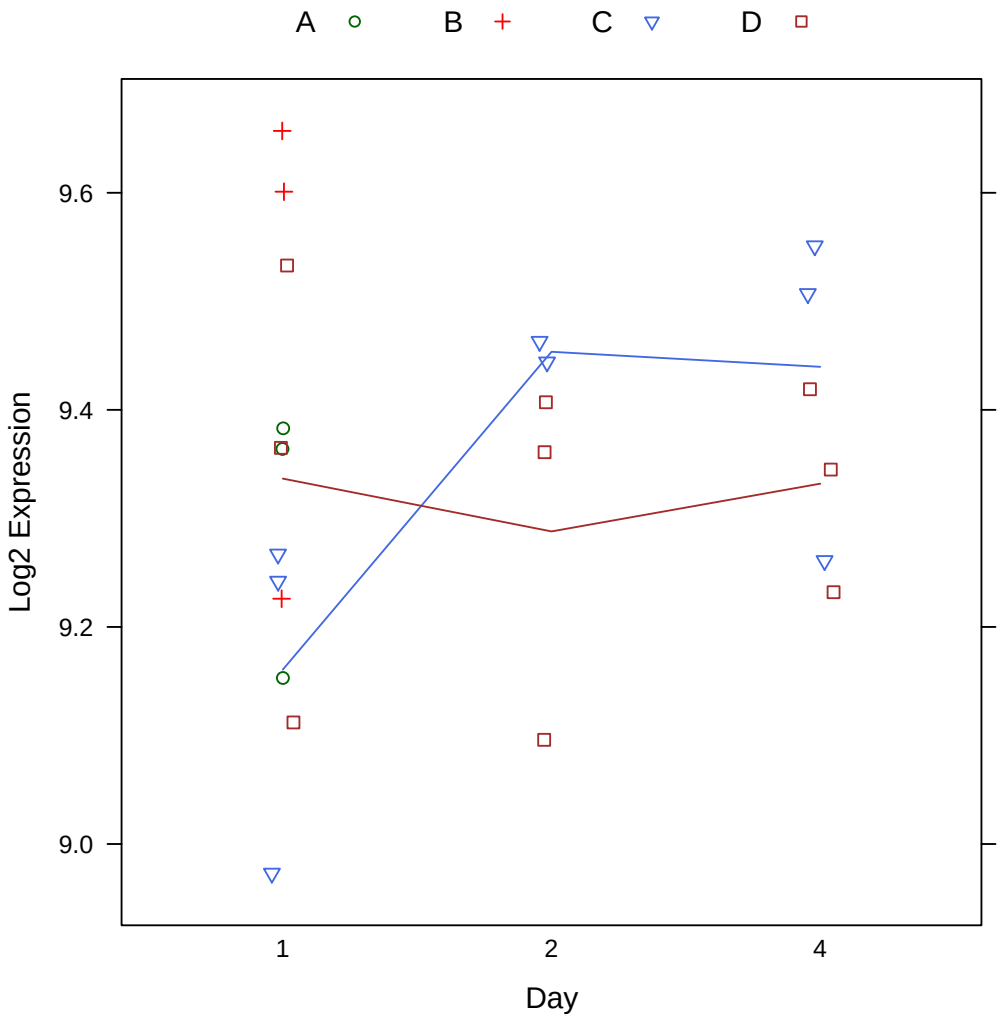

Figure 2: An interaction plot for the stem cell gene expression experiment

```
group1:C -0.243  0.500
group1:D -0.243  0.500  0.500
group2:C -0.214  0.441  0.441  0.441
group2:D -0.243  0.500  0.500  0.500  0.441
group4:C -0.243  0.500  0.500  0.500  0.441  0.500
group4:D -0.243  0.500  0.500  0.500  0.441  0.500  0.500

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-1.48394624 -0.46168591  0.03798095  0.17092529  1.57981011

Number of Observations: 23
Number of Groups: 3
```

```
> print(contrast(lmeFit, list(group = "4:C"), list(group = "4:D")),
+     X = TRUE)
```

```
lme model parameter contrast

   Contrast       S.E.      Lower     Upper    t df Pr(>|t|)
1 0.1076667 0.04766072 0.01425337 0.2010800 2.26 13   0.0417

Contrast coefficients:
  (Intercept) group1:B group1:C group1:D group2:C group2:D group4:C group4:D
1           0        0        0        0        0        0        1       -1
```

Comparing this to the gls model results, the default coefficients have identical parameter estimates, standard errors and test statistics, but their p–values are slightly different. This is due to the difference in how the degrees of freedom are calculated between these models. The same is true for the example contrast for the two models (15 versus 13 degrees of freedom).

2.5 Generalized Estimating Equations

Yet another way to fit a model to these data would be to use a generalized linear model–type framework using normal errors and a log (base 2) link. To account for the within–donor variability, a generalized estimating equation approach can be used. We use the geese function in the geepack package.

```
> geeFit <- geese(2^expression ~ group, data = example2, id = subject,
+     family = gaussian(link = "log"), corstr = "exchangeable")
> summary(geeFit)
```

```
Call:
geese(formula = 2^expression ~ group, id = subject, data = example2,
    family = gaussian(link = "log"), corstr = "exchangeable")

Mean Model:
 Mean Link:                 log
 Variance to Mean Relation: gaussian

 Coefficients:
               estimate      san.se         wald            p
(Intercept)  6.457722542 0.030377051 4.519262e+04 0.000000e+00
group1:B     0.135794630 0.064029822 4.497805e+00 3.393840e-02
group1:C    -0.108696851 0.020631757 2.775630e+01 1.375997e-07
group1:D     0.029699999 0.038863285 5.840283e-01 4.447377e-01
group2:C     0.046523041 0.009171127 2.573302e+01 3.920608e-07
group2:D    -0.019772151 0.023586890 7.026944e-01 4.018798e-01
group4:C     0.085212961 0.020805608 1.677453e+01 4.209444e-05
group4:D     0.007844728 0.014837328 2.795403e-01 5.970027e-01

Scale Model:
 Scale Link:                identity

 Estimated Scale Parameters:
            estimate   san.se     wald           p
(Intercept) 3633.935 1316.328 7.621251 0.005768444

Correlation Model:
 Correlation Structure:     exchangeable
 Correlation Link:          identity

 Estimated Correlation Parameters:
       estimate    san.se     wald           p
alpha 0.6081796 0.2021482 9.051568 0.002624697

Returned Error Value:    0
Number of clusters:   6   Maximum cluster size: 7

> print(contrast(geeFit, list(group = "4:C"), list(group = "4:D")),
+     X = TRUE)

geese model parameter contrast

    Contrast       S.E.      Lower     Upper    Z df Pr(>|Z|)
1 0.07736823 0.02952906 0.01949233 0.1352441 2.62 NA   0.0088
```

```
Contrast coefficients:
  (Intercept) group1:B group1:C group1:D group2:C group2:D group4:C group4:D
1           0        0        0        0        0        0        1       -1
```

For this model, a simple Wald test is calculated. The contrast shows a more significant p–value than the other models, partly due to the scale and partly due to the distributional assumptions about the test statistic.

3 Fold changes

The `contrast` method also computes fold changes using the follow process:

1. For the two groups defined by the `a` and `b` arguments, the predicted outcomes are computed. When the model objects is generated by either `glm` or `geese`, the linear predictor is calculated.
2. The two predicted values are optionally transformed by the `fcFunc` argument. For our gene expression example, we might use `function(u) 2^u`, while for generalized linear models we might use the inverse link function from the `family` object.
3. The predicted value for the `a` group is divided by the predicted value for the `b` group.
4. If the `fcType` argument is `"simple"`, the ratio is returned as the fold change. If the type is `"signed"`, a different calculation is used. If the simple ratio is less than one, the negative reciprocal of the ratio is returned; otherwise the fold change is equal to the simple ratio

The fold change results are contained in the output as `foldchange`. From the first example:

```
> eachTrmtEffect <- contrast(lmFit1, list(diet = levels(example1$diet),
+     group = "vehicle"), list(diet = levels(example1$diet), group = "treatment"),
+     fcfunc = function(u) 2^u)
> print(eachTrmtEffect, X = TRUE)

lm model parameter contrast

   Contrast      S.E.      Lower      Upper     t df Pr(>|t|)
 -0.5686667 0.0737955 -0.7226014 -0.4147320 -7.71 20    0e+00
 -0.2871667 0.0737955 -0.4411014 -0.1332320 -3.89 20    9e-04

Contrast coefficients:
 (Intercept) groupvehicle dietlow fat groupvehicle:dietlow fat
           0            1          0                       0
           0            1          0                       1
```

```
> eachTrmtEffect$foldChange
```

```
       [,1]
1 0.9291896
2 0.9621377
```

4 References

Harrell, F. E. (2001), *Regression Modeling Strategies*. New York: Springer-Verlag.

Hauck, W. W. and Donner, A. (1977), Wald's Test as Applied to Hypotheses in Logit Analysis. *Journal of the American Statistical Association* **72**, 851-863.