Supplementary Note:
Contrasts in R are not contrast weights

One common point of confusion among R users pertains to the use of `contrasts` for estimation and testing treatment contrasts in regression and ANOVA.

Consider this simple example of an experiment to compare $k = 4$ drugs, which are indexed by $i = 1, 2, 3, 4$. The data are unbalanced, with sample sizes $n_1 = 5$, $n_2 = 7$, $n_3 = 6$ and $n_4 = 5$.

Drug	*Responses*						
A ($i = 1$)	6	4	7	7	5		
B ($i = 2$)	5	6	6	6	5	6	6
C ($i = 3$)	2	5	3	1	5	4	
D ($i = 4$)	7	8	8	9	7		

Let $\mu_1, \mu_2, \mu_3, \mu_4$ denote the four treatment means. To test the null hypothesis $\mu_1 = \mu_2 = \mu_3 = \mu_4$, we fit a linear model in R. That is, we regress the outcome on the treatment variable after declaring the treatment to be a factor.

```
> y <- c(6,4,7,7,5,5,6,6,6,5,6,6,2,5,3,1,5,4,7,8,8,9,7)
> drug <- c( rep("A",5), rep("B",7), rep("C",6), rep("D",5) )
> drug <- factor(drug)

> result <- lm( y ~ drug )
> anova(result)
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value   Pr(>F)
drug       3 55.290  18.430  14.374 3.98e-05 ***
Residuals 19 24.362   1.282
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

By default, R uses a dummy-coding scheme with the first level as the baseline or reference category. That is, R will fit the model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \text{error},$$

where

$$X_1 = \begin{cases} 1 & \text{if Drug=B,} \\ 0 & \text{otherwise,} \end{cases} \quad X_2 = \begin{cases} 1 & \text{if Drug=C,} \\ 0 & \text{otherwise,} \end{cases} \quad X_3 = \begin{cases} 1 & \text{if Drug=D,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The estimated coefficients are shown below.

```
> summary(result)

Call:
lm(formula = y ~ drug)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3333 -0.7571  0.2000  0.4762  1.6667

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.80000    0.50640  11.453 5.67e-10 ***
drugB       -0.08571    0.66303  -0.129  0.89850
```

```
drugC       -2.46667    0.68567  -3.597  0.00192 **
drugD        2.00000    0.71616   2.793  0.01161 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.132 on 19 degrees of freedom
Multiple R-squared: 0.6941,     Adjusted R-squared: 0.6459
F-statistic: 14.37 on 3 and 19 DF,  p-value: 3.98e-05
```

In this model, the "intercept" is the mean for the reference group,

$$\hat{\beta}_0 = \hat{\mu}_1 = 5.8,$$

and the "slopes" are the means for the other groups minus the reference,

$$\begin{aligned}
\hat{\beta}_1 &= \hat{\mu}_2 - \hat{\mu}_1 = -0.08571, \\
\hat{\beta}_2 &= \hat{\mu}_3 - \hat{\mu}_1 = -2.46667, \\
\hat{\beta}_3 &= \hat{\mu}_4 - \hat{\mu}_1 = 2.00000.
\end{aligned}$$

What is a contrast? In the terminology of ANOVA, a *contrast* is a weighted sum of the treatment means,

$$L = c_1 \mu_1 + c_2 \mu_2 + \cdots + c_k \mu_k, \tag{2}$$

where the weights sum to zero,

$$c_1 + c_2 + \cdots + c_k = 0.$$

By this standard definition, the coefficients β_1, β_2 and β_3 in our linear model are contrasts,

$$\beta_1 = -1\,\mu_1 + 1\,\mu_2 + 0\,\mu_3 + 0\,\mu_4, \tag{3}$$
$$\beta_2 = -1\,\mu_1 + 0\,\mu_2 + 1\,\mu_3 + 0\,\mu_4, \tag{4}$$
$$\beta_3 = -1\,\mu_1 + 0\,\mu_2 + 0\,\mu_3 + 1\,\mu_4. \tag{5}$$

However, users of R often become confused because the R language uses the term *contrasts* in a different way. When R speaks of contrasts, it does not refer to weighted sums of the treatment means (2). Rather, it refers to the coding scheme such as (1) that defines the regressors in the regression model.

When R speaks of contrasts, it is referring to the coding scheme. Every factor in R has an attribute `contrasts` which defines the regressors that are created and included in a regression model whenever the factor appears on the right-hand side of the model formula, as in `lm(y ~ drug)`. In general, the `contrasts` attribute for a k-level factor is is a $k \times (k-1)$ matrix. The rows of this matrix correspond to the levels of the factor, and there is one column for each regressor included in the model. Here is the `contrasts` matrix for our example:

```
> contrasts(drug)
  B C D
A 0 0 0
B 1 0 0
C 0 1 0
D 0 0 1
```

The columns of this matrix are the definitions of the dummy variables (1) But notice that the columns of the `contrasts` matrix are not the same thing as the weights $c_1, \ldots, c_k$ in the contrasts (3)–(5). The relationship between them is this: *The coding scheme defined by the* `contrasts` *attribute in R is the inverse of the matrix of contrast weights.*

To see this, suppose we append a column of constants (1's) to the left-hand side of the `contrasts` matrix and then invert it.

```
> tmp <- contrasts(drug)
> tmp <- cbind(constant=1,tmp)
> tmp
  constant B C D
A        1 0 0 0
B        1 1 0 0
C        1 0 1 0
D        1 0 0 1

> solve(tmp)
          A B C D
constant  1 0 0 0
B        -1 1 0 0
C        -1 0 1 0
D        -1 0 0 1
```

The rows of this inverse tell us that

$$\begin{aligned}
\beta_0 &= 1\,\mu_1 + 0\,\mu_2 + 0\,\mu_3 + 0\,\mu_4, \\
\beta_1 &= -1\,\mu_1 + 1\,\mu_2 + 0\,\mu_3 + 0\,\mu_4, \\
\beta_2 &= -1\,\mu_1 + 0\,\mu_2 + 1\,\mu_3 + 0\,\mu_4, \\
\beta_3 &= -1\,\mu_1 + 0\,\mu_2 + 0\,\mu_3 + 1\,\mu_4.
\end{aligned}$$

With the exception of the intercept term β_0, which is not a contrast, the coeficients in this model are the contrasts.

Built-in coding schemes. R has several built-in coding schemes that are very useful. Dummy coding (the default) is `contr.treatment`.

```
> # apply dummy coding to a factor with 4 levels
> contrasts(drug) <- contr.treatment(4)
> contrasts(drug)
  2 3 4
A 0 0 0
B 1 0 0
C 0 1 0
D 0 0 1
>
> # use the 3rd category as the reference group
> contrasts(drug) <- contr.treatment(4, base=3)
> contrasts(drug)
  1 2 4
A 1 0 0
B 0 1 0
C 0 0 0
D 0 0 1
```

Another common scheme is `contr.sum`, the $(1, 0, -1)$ coding that produces the usual sum-to-zero effects that are familiar to students of ANOVA.

```
> contrasts(drug) <- contr.sum(4)
> contrasts(drug)
  [,1] [,2] [,3]
A    1    0    0
B    0    1    0
C    0    0    1
D   -1   -1   -1

> tmp <- contrasts(drug)
```

```
> tmp <- cbind(const=1,tmp)
> solve(tmp)
           A     B     C     D
const  0.25  0.25  0.25  0.25
       0.75 -0.25 -0.25 -0.25
      -0.25  0.75 -0.25 -0.25
      -0.25 -0.25  0.75 -0.25
```

In this example, the coefficients of the linear model are

$$\begin{aligned}
\beta_0 &= (\mu_1 + \mu_2 + \mu_3 + \mu_4)/4, \\
\beta_1 &= \mu_1 - (\mu_1 + \mu_2 + \mu_3 + \mu_4)/4, \\
\beta_2 &= \mu_2 - (\mu_1 + \mu_2 + \mu_3 + \mu_4)/4, \\
\beta_3 &= \mu_3 - (\mu_1 + \mu_2 + \mu_3 + \mu_4)/4,
\end{aligned}$$

and the effect that is omitted from the model to avoid singularity is

$$\mu_4 - (\mu_1 + \mu_2 + \mu_3 + \mu_4)/4 = -(\beta_1 + \beta_2 + \beta_3).$$

Defining your own coding scheme. Sometimes we need to define our own coding scheme to give us desired contrasts. For example, suppose we want to create a model for a treatment with $k = 4$ levels in which the following contrasts appear as coefficients:

$$\begin{aligned}
\beta_1 &= \mu_2 - \mu_1 \\
\beta_2 &= \mu_3 - \mu_2 \\
\beta_3 &= \mu_4 - \mu_3
\end{aligned}$$

To figure out what the `contrasts` attribute needs to be, create a matrix in which the contrast weights appear as rows. Then append a row at the top with all elements equal to 1/4, so that the intercept is defined as

$$\beta_0 = (\mu_1 + \mu_2 + \mu_3 + \mu_4)/4$$

Then invert the matrix. The inverse, without the first column, is what you want to use for `contrasts`.

```
> tmp <- matrix(NA,4,4)
> tmp[1,] <- 1/4
> tmp[2,] <- c(-1,1,0,0)
> tmp[3,] <- c(0,-1,1,0)
> tmp[4,] <- c(0,0,-1,1)
> tmp
      [,1]  [,2]  [,3] [,4]
[1,]  0.25  0.25  0.25 0.25
[2,] -1.00  1.00  0.00 0.00
[3,]  0.00 -1.00  1.00 0.00
[4,]  0.00  0.00 -1.00 1.00
>
> solve(tmp)
     [,1]  [,2] [,3]  [,4]
[1,]    1 -0.75 -0.5 -0.25
[2,]    1  0.25 -0.5 -0.25
[3,]    1  0.25  0.5 -0.25
[4,]    1  0.25  0.5  0.75
> contrasts(drug) <- tmp[,2:4]
> contrasts(drug)
   [,1]  [,2] [,3]
A  0.25  0.25 0.25
B  1.00  0.00 0.00
C -1.00  1.00 0.00
D  0.00 -1.00 1.00
```

What happens when contrasts are orthogonal? Two contrasts,

$$L_1 = \sum_{i=1}^{k} c_i \mu_i \quad \text{and} \quad L_2 = \sum_{i=1}^{k} d_i \mu_i$$

are said to be orthogonal if the products of the weights sum to zero, $\sum_{i=1}^{k} c_i d_i = 0$.

With a k-level factor, we can define a set of $k-1$ orthogonal contrasts to characterize the differences among the μ_i's. For example, if we have a 3-level factor whose levels are ordered, we may wish to define linear and quadratic contrasts as

$$\begin{aligned} L_1 &= -1\,\mu_1 + 0\,\mu_2 + 1\,\mu_3, \\ L_2 &= 1\,\mu_1 - 2\,\mu_2 + 1\,\mu_3. \end{aligned}$$

What coding scheme can we apply to give us L_1 and L_2 as coefficients in our linear model? To see what the coding scheme should be, create a matrix whose rows contain the contrast weights and invert it.

```
> tmp <- matrix(NA,3,3)
> tmp[1,] <- 1/3
> tmp[2,] <- c(-1,0,1)
> tmp[3,] <- c(1,-2,1)
> tmp
           [,1]       [,2]      [,3]
[1,]  0.3333333  0.3333333 0.3333333
[2,] -1.0000000  0.0000000 1.0000000
[3,]  1.0000000 -2.0000000 1.0000000

> solve(tmp)
     [,1] [,2]       [,3]
[1,]    1 -0.5  0.1666667
[2,]    1  0.0 -0.3333333
[3,]    1  0.5  0.1666667
```

We would set the `contrasts` attribute for this factor equal to a 3×2 matrix whose first column is $(-1/2, 0, 1/2)^T$ and whose second column is $(1/6, -1/3, 1/6)^T$.

Notice that in this special case, the columns of the `contrasts` matrix are rescaled versions of the vectors of weights that define L_1 and L_2. This will happen whenever we have $k-1$ contrasts that are mutually orthogonal. But it does not happen with nonorthogonal contrasts.