hypr: An R package for hypothesis-driven contrast coding

Maximilian M. Rabe[1], **Shravan Vasishth**[1], **Sven Hohenstein**[1], **Reinhold Kliegl**[1], **and Daniel J. Schad**[1, 2]

1 University of Potsdam **2** Tilburg University

DOI: 10.21105/joss.02041

Software

- Review
- Repository
- Archive

Editor: Marcos Vital

Submitted: 24 January 2020
Published: 07 February 2020

Summary

The `hypr` package in R provides the researcher with a straightforward interface to generate contrast matrices from research hypotheses and the reverse. It can be used to derive contrast matrices for custom research hypotheses and as an instructional tool to understand what research hypotheses a given contrast matrix tests. We previously reviewed the statistical methodolgy in a tutorial on contrast coding based on the generalized matrix inverse (Schad, Vasishth, Hohenstein, & Kliegl, 2020).

Contrast coding in R

In many empirical sciences, the linear model and variants are used to investigate the statistical relationship between a continuous *dependent (outcome) variable*, such as response times or percent correct in cognitive research, and one or more *independent (explanatory) variables*. If independent variables are categorical (or, `factor` instances in R), such as the assignment to a baseline and a treatment group, or other categorical experimental conditions, then the researcher must decide how to code the contrasts between the levels of that independent variable, typically termed *contrast coding*.

For a set of standard contrasts (e.g., treatment contrast, sum contrast, Helmert contrast), coding schemes are readily available as part of the base R `stats` package. One such example is the function `contr.treatment()` for treatment contrasts. Each of these functions generates a *contrast matrix*, e.g.:

```
contr.treatment(c("baseline","trt1","trt2"))
```

```
##          trt1 trt2
## baseline    0    0
## trt1        1    0
## trt2        0    1
```

A contrast matrix can be easily assigned to a `factor`:

```
contrasts(some_factor) <- contr.treatment(c("baseline","trt1","trt2"))
```

When fitting a linear model, R will transform the factor to as many contrasts as there are columns in the contrast matrix. For each factor level, it will assign the respective contrast coefficient as a numeric value to the contrast for each observation, and it will use these numeric contrasts as covariates for fitting the linear model.

Contrast coding with hypr

For simple contrast coding schemes, contrast matrices are easy to interpret. However, when custom contrast coding schemes are defined to test particular research hypotheses, contrast matrices become harder to interpret. The hypr package in R provides a set of functions to aid in the process of custom contrast specification. The package provides two functionalities: First, it allows the user to translate research hypotheses to the corresponding contrast matrices. Second, given a contrast matrix, the package translates it to the corresponding hypothesis tests.

Most importantly, the package provides the hypr() function which constructs a hypr object. An arbitrary set of linear research (null) hypotheses can be passed as arguments. For example, to create a hypr object that tests a baseline condition against zero and two treatment conditions against the baseline (i.e., a treatment contrast with one baseline and two treatments), can be created as follows:

```
trtC <- hypr(baseline~0, trt1~baseline, trt2~baseline)
trtC
```

```
## hypr object containing 3 null hypotheses:
## H0.1: 0 = baseline
## H0.2: 0 = trt1 - baseline
## H0.3: 0 = trt2 - baseline
##
## Hypothesis matrix (transposed):
##          [,1] [,2] [,3]
## baseline  1   -1   -1
## trt1      0    1    0
## trt2      0    0    1
##
## Contrast matrix:
##          [,1] [,2] [,3]
## baseline 1    0    0
## trt1     1    1    0
## trt2     1    0    1
```

The term baseline or μ_1 represents the mean response in the baseline condition, while trt1 and trt2, or μ_2 and μ_3, respectively, represent the means of the response in the two treatment conditions. As can be seen above, the object contains three research (null) hypotheses, which are represented as the three columns of the "Hypothesis matrix (transposed)". Hypothesis H_{0_1} (i.e., the first column of the transposed hypothesis matrix) tests whether the baseline condition is significantly different from zero, while H_{0_2} and H_{0_3} (i.e., columns two and three) each test whether one of the two treatment conditions are significantly different from the baseline condition. These hypotheses can be formally specified as:

$$
\begin{aligned}
H_{0_1}: &\quad \mu_1 = 0 \\
H_{0_2}: &\quad \mu_2 = \mu_1 \\
H_{0_3}: &\quad \mu_3 = \mu_1
\end{aligned}
$$

The *hypothesis matrix* contains the coefficient or weight of each of the group means in each of the hypotheses. Groups that are not part of a particular hypothesis have a weight of zero. The generalized inverse of the hypothesis matrix yields the desired *contrast matrix*.

With the provided `contr.hypothesis()` function, the resulting contrast matrix (as seen in the lower third of the output above) can be derived and – as expected – it equals the output of the `contr.treatment(3)` function call:

```
contr.hypothesis(trtC)
```

```
##          [,1] [,2]
## baseline    0    0
## trt1        1    0
## trt2        0    1
```

This matrix can be assigned to a factor in the same way as the matrices generated by the `contr.*()` functions in R:

```
contrasts(some_factor) <- contr.hypothesis(trtC)
```

The package provides more useful functions to retrieve and modify a `hypr` object's contrast matrix or (transposed) hypothesis matrix, such as `cmat()`, `thmat()`, and `hmat()`

Contrast validation with hypr

As previously mentioned, `hypr` can also be used to derive a set of tested research (null) hypotheses from a given contrast matrix. This may help to verify what a given contrast matrix is testing. To do so, one can create an empty `hypr` object and set its contrast matrix to whatever matrix is to be inspected.

We here consider a successive difference contrast, which is designed to test the differences between each successive pair of ordered groups (Venables & Ripley, 2002). For example, with four levels, the contrast should test three hypotheses: the difference between the second and the first factor level, between the third and second factor level, as well as between the fourth and third level. This is not immediately clear from looking at the contrast matrix alone:

```
MASS::contr.sdif(4)
```

```
##     2-1  3-2   4-3
## 1 -0.75 -0.5 -0.25
## 2  0.25 -0.5 -0.25
## 3  0.25  0.5 -0.25
## 4  0.25  0.5  0.75
```

After initializing the `hypr` object, its contrast matrix can be set to a successive difference contrast as follows:

```
sdifC <- hypr()
cmat(sdifC) <- MASS::contr.sdif(4)
sdifC
```

```
## hypr object containing 3 null hypotheses:
## H0.2-1: 0 = -X1 + X2
## H0.3-2: 0 = -X2 + X3
## H0.4-3: 0 = -X3 + X4
```

```
##
## Hypothesis matrix (transposed):
##    2-1 3-2 4-3
## X1 -1    0   0
## X2  1   -1   0
## X3  0    1  -1
## X4  0    0   1
##
## Contrast matrix:
##    2-1  3-2  4-3
## X1 -3/4 -1/2 -1/4
## X2  1/4 -1/2 -1/4
## X3  1/4  1/2 -1/4
## X4  1/4  1/2  3/4
```

When evaluating the output, it can be seen that the “hypothesis matrix (transposed)” (i.e., the transposed generalized inverse of the contrast matrix) now shows much more meaningful information (compared to the contrast matrix): it shows the hypotheses that the contrast matrix tests, and can thus be used to interpret the results.

The derived research hypotheses can be rewritten as:

$$\begin{aligned}
H_{0_1}: &\quad \mu_2 = \mu_1 \\
H_{0_2}: &\quad \mu_3 = \mu_2 \\
H_{0_3}: &\quad \mu_4 = \mu_3
\end{aligned}$$

As expected, the contrasts test each of the successive differences between μ_1, μ_2, μ_3, and μ_4 as outlined above. In a case, where we would be unsure about what a given contrast matrix tests, this hypothesis matrix could thus show us the hypotheses that are tested by the contrast matrix. In order to learn what the intercept term tests in this example case, we can add an intercept to the contrast matrix:

```
cmat(sdifC) <- cbind(int = 1, MASS::contr.sdif(4))
sdifC
```

```
## hypr object containing 4 null hypotheses:
## H0.int: 0 = 1/4*X1 + 1/4*X2 + 1/4*X3 + 1/4*X4
## H0.2-1: 0 = -X1 + X2
## H0.3-2: 0 = -X2 + X3
## H0.4-3: 0 = -X3 + X4
##
## Hypothesis matrix (transposed):
##    int 2-1 3-2 4-3
## X1 1/4  -1   0   0
## X2 1/4   1  -1   0
## X3 1/4   0   1  -1
## X4 1/4   0   0   1
##
## Contrast matrix:
##    int  2-1  3-2  4-3
## X1   1 -3/4 -1/2 -1/4
## X2   1  1/4 -1/2 -1/4
## X3   1  1/4  1/2 -1/4
## X4   1  1/4  1/2  3/4
```

In the first column of the transposed hypothesis matrix, we can see that the intercept term – as is typically the case for centered contrasts – is the grand mean, i.e. the mean of means:

$$H_{0_{Int}}: \qquad 0 = \frac{\mu_1 + \mu_2 + \mu_3 + \mu_4}{4}$$

Acknowledgements

The development of this package was supported by German Research Foundation (DFG), SFB 1287 *Limits of Variability in Language*, project number 317633480, and Center for Interdisciplinary Research, Bielefeld (ZiF), Cooperation Group *Statistical models for psychological and linguistic data*.

References

Schad, D. J., Vasishth, S., Hohenstein, S., & Kliegl, R. (2020). How to capitalize on a priori contrasts in linear (mixed) models: A tutorial. *Journal of Memory and Language*, *110*. doi:10.1016/j.jml.2019.104038

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S* (Fourth ed.). Springer.