Linear Models: Summary

Abhinav Mishra, Kristin Köhler, Sanket Gosavi, Utkarsha Kandale

2022-05-11

Abstract

A gist to Linear models, argues for its applications in R, and presents a summary sheet intended for an academic audience.

The Design Matrix

It is also known as *model matrices*, derived from the measurements, we call them experimental units, for the argument that N different entities has two types of variables: **Indicator** and **Continuous**, broadly speaking. The indicator variable indicates if the experimental unit has a certain characteristic or not. If we want to fit a model, definite the design matrix is the first step, and finding the *least square estimates* (LSE) is the second.

$$Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, i = 1, \ldots, N$$

The choice of design is important for two reasons: it encodes which coefficients will fit the linear model, and also responsible for the inter-relationship between the samples. It doesn't follow the straightforward dictum of the samples, and basic information about each sample doesn't implies a ***current*** design matrix. For comparison between different groups, the design that works has at least, in principle, has two columns: Intercept, and the second column that specifies which samples is stored in the second group.

The intercept column represents the population average of the first group, and the second coefficient represents the difference of population of the first and second group. As someone who is doing linear modelling, their interest should be on the second coefficient to know if there is a difference between the two groups, naturally. We have to specify in R that the values in test, and control group should not be interpreted numerically, but as different levels of a factor. If we don't do that, then we will get the matrix that we don't want, and the reason would be we provided a numeric variable instead of an "indicator" to *formula()* and *model.matrix()* functions.

Using linear algebra, we can generalize the above equation like this:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

or

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \\ 1 & x_N \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix}$$

The design matrix is the matrix $\mathbf{X}$. We'll come back to that in the next section.

factor() is an instrument to indicate the membership of the group by assigning 0's and 1's. The function default is the unique set of values taken by *as.character(x)*, sorted into increasing order of x. The factor levels are assumed to be ordered (as a logical flag). Hence, we conclude that names in *factor()* doesn't matter, but order of the names in of utmost importance. Please note that the levels of a factor are by default sorted, but the sort order may well depend on the locale at the time of creation, and should not be assumed to be ASCII. We should always ensure separate coefficients for each group, if we need to model with more groups. If someone is more inclined toward more, group combinations using relational operators (e.g. + or : or *) in *model.matrix()* would be helpful. As far as the *model.matrix()* is concerned, it will grab the variable the variable from the global environment in R, unless data is passed as a data.frame in the data argument.

Generally speaking, we're contrasting the test group against control, so the test group becomes the level. The level which is contrasted against is called reference level. If we want any other group to be the reference level except the default one, use *relevel()* or provide explicit levels to the factor call. In order to be able to do a mathematical transformation of a variable in the design formula, we use *I()* to inhibit interpretation of formula operators so they can be used as arithmetically, in *model.matrix()* as by protecting an object by enclosing it in *I()* in a call to *data.frame* inhibits the conversion of character vectors to factors and the dropping of names, and ensures that matrices are inserted as single columns.

Indicator variables assume a different mean between two groups while continuous variables assume a very specific relationship between outcome and predictor values. **If data doesn't support the model being used, we should avoid using continuous variables to adjust for any parameter.**

The mathematics behind lm() function

Assuming that the population standard deviation is the same for both groups with equally distributed errors, the *lm()* function must consume the formula and data frame to produce a design matrix, and a response vector, and then use that to compute linear regression coefficients ($\boldsymbol{\beta}$).

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$$

where $\hat{\boldsymbol{\beta}}$ is unbiased and minimizes $\boldsymbol{\beta}$.

The design matrix $\mathbf{X}$ gets decomposed into $\mathbf{Q}$ *orthogonal*, and $\mathbf{R}$ *triangular* matrix by *QR decomposition* method using *householder transformations* for the calculation of the *sum of least squares* after solving the *Q*, and *R* system of linear equations.

Now, rewriting the equation for calculating LSE using $\mathbf{QR}$ instead of $\mathbf{X}$ we have:

$$\mathbf{X}^\top \mathbf{X} \boldsymbol{\beta} = \mathbf{X}^\top \mathbf{Y}$$

$$(\mathbf{QR})^\top (\mathbf{QR}) \boldsymbol{\beta} = (\mathbf{QR})^\top \mathbf{Y}$$

$$\mathbf{R}^\top (\mathbf{Q}^\top \mathbf{Q}) \mathbf{R} \boldsymbol{\beta} = \mathbf{R}^\top \mathbf{Q}^\top \mathbf{Y}$$

$$\mathbf{R}^\top \mathbf{R} \boldsymbol{\beta} = \mathbf{R}^\top \mathbf{Q}^\top \mathbf{Y}$$

$$(\mathbf{R}^\top)^{-1}\mathbf{R}^\top\mathbf{R}\boldsymbol{\beta} = (\mathbf{R}^\top)^{-1}\mathbf{R}^\top\mathbf{Q}^\top\mathbf{Y}$$

$$\mathbf{R}\boldsymbol{\beta} = \mathbf{Q}^\top\mathbf{Y}$$

Now we are ready to find LSE using the *QR decomposition* using *householder reflections*. To solve:

$$\mathbf{R}\boldsymbol{\beta} = \mathbf{Q}^\top\mathbf{Y}$$