

## Video 97 : R LAB: LHT FUNCTION

Professor: *Arnab Chakraborty**Semanti Dutta*

Previously we fitted both the Null Model and the Overall Model for hypothesis testing. But we can also perform hypothesis testing by only fitting the full model and then by just specifying the null hypothesis as some kind of linear restriction on the parameters. One way to do this is to take the null hypothesis into account, create a new model and fit it. But most of the time we would like to specify the null hypothesis symbolically into R. To understand how to do that, let us look at an example :

```
library(faraway)
library(car)
names(coagulation)
head.matrix(coagulation)
summary(coagulation)
```

The Output is as follows :

```
> names(coagulation)
[1] "coag" "diet"
> head.matrix(coagulation)
  coag diet
1   62   A
2   60   A
3   63   A
4   59   A
5   63   B
6   67   B

> summary(coagulation)
      coag      diet
Min.   :56.00  A:4
1st Qu.:61.75  B:6
Median :63.50  C:6
Mean   :64.00  D:8
3rd Qu.:67.00
Max.   :71.00
```

The dataset *coagulation* is from package *faraway*. There are two variables, the response variable *coag* which is continuous and the input variable *diet* has 4 categories, where A occurs 4 times, B occurs 6 times, and so on. The first few entries of the dataset is shown above.

Now, let us fit a model :

```
fit=lm(coag~diet,coagulation)
fit
```

The Output is as follows :

```

> fit=lm(coag~diet,coagulation)
> fit

Call:
lm(formula = coag ~ diet, data = coagulation)

Coefficients:
(Intercept)      dietB      dietC      dietD
  6.100e+01   5.000e+00   7.000e+00   2.991e-15

```

Notice, the model consists of an intercept term and *dietB*, *dietC* and *dietD*. R drops *dietA* and forcefully sets it 0. So the GM model is as follows :

$$y_{ij} = \mu + \alpha_i + \epsilon_{ij}$$

where  $\alpha_1$  is the parameter for *dietA* which is forcefully set as 0,  $\alpha_2$  is for *dietB*, and so on. Now suppose we want to test whether  $\alpha$  for *dietB* is same as  $\alpha$  for *dietC*, i.e, we want to check whether the parameters for *dietB* and *dietC* are same or not. For that, we write the following command :

```

lht(fit,'dietB-dietC')
lht(fit,'dietB-dietC',5)

```

**Remark.** Notice inside the command *lht*, we use *dietB*, *dietC* as these are the name of the parameters specified by R.

The first command checks whether  $dietB - dietC = 0$ . If we add another parameter specifying a constant *c*, then we would get the test for checking whether  $dietB - dietC = c$ . Thus, the second command checks if  $dietB - dietC = 5$ . To understand this, look at the two outputs given below :

<pre> &gt; lht(fit,'dietB-dietC') Linear hypothesis test  Hypothesis: dietB - dietC = 0  Model 1: restricted model Model 2: coag ~ diet    Res.Df RSS Df Sum of Sq      F Pr(&gt;F) 1      21 124          12 2.1429 0.1588 2      20 112 </pre>	<pre> &gt; lht(fit,'dietB-dietC',5) Linear hypothesis test  Hypothesis: dietB - dietC = 5  Model 1: restricted model Model 2: coag ~ diet    Res.Df RSS Df Sum of Sq      F      Pr(&gt;F) 1      21 259 2      20 112  1      147 26.25 5.178e-05 *** </pre>
--	---

Model 1 is the restricted model and model 2 is the general model. Here, we get the measure of  $RSS_0 - RSS$  from the sum of squares value. Notice the p-value in the second case is very small and hence it is very significant. This is also shown by the triple stars marks beside the p-value.

We can also perform this same test using matrix notation. Notice there are 4 parameters : *intercept*, *dietB*, *dietC* and *dietD*. To check any linear combination of the parameters, we can use the following command :

```

lht(fit,c(0,1,-1,0))
lht(fit,c(0,1,-1,0),5)

```

The first command checks

$$0 \times intercept + (1) \times dietB + (-1) \times dietC + 0 \times dietD = 0$$

Whereas the second command checks

$$0 \times \text{intercept} + (1) \times \text{dietB} + (-1) \times \text{dietC} + 0 \times \text{dietD} = 5$$

Thus, these set of commands should give the same output as the previous set of commands as they are performing the same tests of hypothesis respectively. This is indeed true as we get the same outputs :

```
> lht(fit,c(0,1,-1,0))
Linear hypothesis test

Hypothesis:
dietB - dietC = 0

Model 1: restricted model
Model 2: coag ~ diet

   Res.Df RSS Df Sum of Sq    F Pr(>F)
1      21 124   0         0.00    NA  NA
2      20 112   1        12 2.1429 0.1588
```

```
> lht(fit,c(0,1,-1,0),5)
Linear hypothesis test

Hypothesis:
dietB - dietC = 5

Model 1: restricted model
Model 2: coag ~ diet

   Res.Df RSS Df Sum of Sq    F    Pr(>F)
1      21 259   0         0.00    NA  NA
2      20 112   1       147 26.25 5.178e-05 ***
```