ONE-WAY ANOVA
R: Fitting two parametrisation.

RAHUL DEBNATH
Under mentorship of Prof. Arnab Chakraborty

31 AUGUST 2022
Indian Statistical Institute, Kolkata.

> ***"THE FUTURE BELONGS TO THOSE WHO BELIEVE IN THE BEAUTY OF THEIR DREAMS."***
>
> Eleanor Roosevelt

Hello there, readers ! Yes, you. I need your eyes and attention here in 3 ...... 2 .... 1 ......

Today, what we are interested to discuss is about Fitting two prametrisation of the same model using R.

The first thing that we shall try to do is to fit the familiar model that we have here below i.e.,

$$\boxed{y_{ij} = \mu + \alpha_i + \epsilon_{ij}} \tag{1}$$

Now, the dataset that we shall be working with is given below :

DataSet :

variety	yield
1	210.3
2	245.0
2	248.9
3	212.3
3	230.4
2	250.1
1	213.5
1	212.4

So, here we have this agricultural dataset depicting three different varieties of HYV seeds and their corresponding yields in each case. The μ that we used here in the above equation (1) introduces an intercept column i.e., the column which consists of all it's elements being "1" and it is the default behavior of R. So, what we do now in R is that we write the following command :

$$\boxed{\mathbf{fit1 = lm(yield \sim variety, agri)}}$$

Now, we fit the model and see what output it produces in R using the command :

$$\boxed{\mathbf{fit1}}$$

and ....Ahhh, what we get is our first surprise !!!!

Now, what's the surprise ?

Let's first look at the output :

```
Call:
lm(formula = yield ~ variety, data = agri.data)

Coefficients:
(Intercept)     variety2     variety3
    212.067       35.933        9.283
```

So, the surprise is that we can clearly see the Intercept term, variety2 and variety3 terms but unfortunately we are unable to see the term"variety1". We notice that the value of $\hat{\mu}$ is equal to 212.067 ,$\hat{\alpha}_2$ is 35.933 and that of $\hat{\alpha}_3$ is 9.283. But we cannot see the value of $\hat{\alpha}_1$ here.

So, what has actually happened to $\hat{\alpha}_1$?.....Woah, can you think about it for a second?

Well, what R actually does as well as other softwares do is the same thing that whenever it constructs the model matrix i.e., the design matrix, it will never construct a model matrix which is not a full column rank. It means that if the columns are not linearly independent, the software will not construct the design matrix as a result we can say that if it finds some linear dependency then it will cast away one of the columns which has happened precisely in our case. So, it has discarded the column of variety1.

What we do now is that we try to understand this fact by looking at the model matrix constructed by R for us. So, in R what we do now is we run the following command:

model.matrix(fit1)

Now, let us look at what R gives as an output :

```
(Intercept) variety2 variety3
1           1        0         0
2           1        1         0
3           1        1         0
4           1        0         1
5           1        0         1
6           1        1         0
7           1        0         0
8           1        0         0
attr(,"assign")
[1] 0 1 1
attr(,"contrasts")
attr(,"contrasts")$variety
[1] "contr.treatment"
```

So, what do you notice ?

Clearly, we can see that there is a column for the term Intercept i.e, $\hat{\mu}$,one column for variety2 i.e, $\hat{\alpha}_2$, another column for variety3 i.e, $\hat{\alpha}_3$ and the remaining column for variety1 i.e, $\hat{\alpha}_1$ is just thrown away i.e, discarded.

This is natural in R as we notice that all the intercept terms are "1", and variety1 will have "1" in those places where this variety has been plotted, similarly variety2 and variety3 will also have "1" in those places where that variety has been plotted. And if we add all the variety1, varety2 and variety3 columns, i.e., $\hat{\alpha}_1$,$\hat{\alpha}_2$ and $\hat{\alpha}_3$, we will get the same thing as the intercept column μ. i.e.,for each row of the columns what we get is

$$\boxed{\textbf{Intercept} = \textbf{variety1} + \textbf{variety2} + \textbf{variety3}} \tag{2}$$

So, it dropped one column i.e.,in our case it dropped the column $\hat{\alpha}_1$.Since, intercept value is always "1" and we know the values of variety2 and variety3, we can write the value of variety1 in terms of linear combination of other terms.

Now, the question that arises in our mind is that how it selected which column to drop?
It could have dropped the Intercet column, or the variety2 column or even the variety3 column. But it dropped the variety1 column.

Can you guess why it dropped the variety1 column and not the other ones?

......Well, R has this habit that it will take the bunches. The first bunch consists of the Intercept term, the next bunch consists of the variety terms.
In each bunch if there is any problem it will drop the first few columns and hence, in our model it dropped the variety1 column. So what it effectively fits is the model:

$$\boxed{y_{ij} = \mu + \alpha_i + \epsilon_{ij}} \tag{3}$$

$$\boxed{\alpha_1{=}0}$$

So, we can say that the estimate for α_1 has been forced to be "0" which makes the least squared estimator unique.
Now, talking about other softwares which may behave in different ways, let us know about some of them:

1. SPSS drops the last column, so in our case variety3 would have been dropped.

2. SAS has an option by which we can choose which column to be dropped, so in our case we could have dropped any one of the columns.

Now, suppose we want to fit the same model but in a slightly different incarnation.Then, let us work with the very first model which we have discussed earlier i.e.,

$$\boxed{y_{ij} = \alpha_i + \epsilon_{ij}} \tag{4}$$

Notice that here we do not have the Intercept term μ. Now, going back to R we write the following command :

$$\boxed{\mathbf{fit2 = lm(yield \sim variety - 1, agri)}}$$

Can u quickly guess why we used "-1" here in the command ?

Ok, let me clarify this. The "-1" which we introduced here guarantees that the Intercept term is discarded. So, this is an equivalent model.

So, what is meant by equivalent here ?
To understand this let us have a look at what R produces as output by entering the following command in R:

fit2

Let us have a look at the output that we get:

```
Call:
lm(formula = yield ~ variety - 1, data = agri.data)

Coefficients:
variety1  variety2  variety3
   212.1     248.0     221.3
```

woah..... magic , we now see that we have values of $\hat{\alpha_1}$ is equal to 212.1 ,$\hat{\alpha_2}$ is 248.0 and that of $\hat{\alpha_3}$ is 221.3.

That means we have values for the terms variety1, variety2 as well as variety3. So, we can say that as the rank of the model matrix is "3",so it is always going to provide us three estimates.
Now, let us have a look at the model/design matrix for fit2. We thus put the following command in R:

model.matrix(fit2)

As an output we get the following:

```
  variety1 variety2 variety3
1        1        0        0
2        0        1        0
3        0        1        0
4        0        0        1
5        0        0        1
6        0        1        0
7        1        0        0
8        1        0        0
attr(,"assign")
[1] 1 1 1
attr(,"contrasts")
attr(,"contrasts")$variety
[1] "contr.treatment"
```

What difference do you notice here ?

We notice that there is no all plot values "1" in any column. For variety1 we have "1" only in plot1, plot7 and plot8. For variety2 we have "1" only in plot2, plot3 and plot6. The remaining ones have "1" for variety3 ,i.e, plot4 and plot5.
Thus, what we get is a full column matrix which means all the columns here are linearly independent of each other.

That's all for this session,we shall continue with switching parametrisations in R in our next session.

Thank You for reading with patience.

NOTE : The whole R code that we used for this session is given below for reference purpose:

```
#We create the dataset using the command:
agri<-data.frame(
variety=c(1,2,2,3,3,2,1,1),yield=c(210.3,245.0,248.9,212.3,230.4,250.1,213.5,212.4),
stringsAsFactors=FALSE)
#Print the dataset:
print(agri)
#Dimension of dataset:
dim(agri)
#Names of data columns:
names(agri)
#View dataset:
agri
#Factoring variety column :
agri$variety = factor(agri$variety)
#First fit model:
fit1=lm(yield~variety,agri)
#View output of fit1, the estimated coefficients:
fit1
#View model matrix of fit1:
model.matrix(fit1)
#Second fit model:
fit2=lm(yield~variety-1,agri)
#View output of fit2 ,the estimated coefficients:
fit2
#View model matrix of fit2:
model.matrix(fit2)
```

Also, all the outputs that we get in R are as follows:

```
>print(agri)

 variety yield
1       1 210.3
2       2 245.0
3       2 248.9
4       3 212.3
5       3 230.4
6       2 250.1
7       1 213.5
8       1 212.4

> dim(agri)
[1] 8 2
> names(agri)
[1] "variety" "yield"
> agri
  variety yield
1       1 210.3
```

```
2       2 245.0
3       2 248.9
4       3 212.3
5       3 230.4
6       2 250.1
7       1 213.5
8       1 212.4

> fit1

Call:
lm(formula = yield ~ variety, data = agri)

Coefficients:
(Intercept)     variety2     variety3
    212.067       35.933        9.283

> model.matrix(fit1)
  (Intercept) variety2 variety3
1           1        0        0
2           1        1        0
3           1        1        0
4           1        0        1
5           1        0        1
6           1        1        0
7           1        0        0
8           1        0        0
attr(,"assign")
[1] 0 1 1
attr(,"contrasts")
attr(,"contrasts")$variety
[1] "contr.treatment"


> fit2

Call:
lm(formula = yield ~ variety - 1, data = agri)

Coefficients:
variety1  variety2  variety3
   212.1     248.0     221.3

> model.matrix(fit2)
  variety1 variety2 variety3
1        1        0        0
2        0        1        0
3        0        1        0
4        0        0        1
5        0        0        1
```

```
6        0        1        0
7        1        0        0
8        1        0        0
attr(,"assign")
[1] 1 1 1
attr(,"contrasts")
attr(,"contrasts")$variety
[1] "contr.treatment"
```