4.2: The Overall Mean Model

Model 1 - The Overall Mean Model

$$Y_{ij} = \mu + \epsilon_{ij} \tag{4.2.1}$$

which simply fits an overall or "grand" mean. This model reflects the situation where H_0 is true, implying that $\mu_1 = \mu_2 = \ldots = \mu_T$.

To understand how various facades of the model relate to each other, let us look at a toy example with 3 treatments (or factor levels) and 2 replicates of each treatment.

We have 6 observations, which means that $\mathbf{Y}$ is a column vector of dimension 6 and so is the error vector $\boldsymbol{\mathcal{E}}$ where its elements are the random error values associated with the 6 observations. In the GLM model of $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\mathcal{E}}$, the design matrix $\mathbf{X}$ for the overall mean model turns out to be a 6-dimensional column vector of ones. The parameter vector, $\boldsymbol{\beta}$, is a scalar equal to μ, the overall population mean.

That is,

$$\mathbf{Y} = \begin{bmatrix} 2 \\ 1 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix}, \ \mathbf{X} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \boldsymbol{\beta} = [\mu], \text{ and } \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \end{bmatrix} \tag{4.2.2}$$

Using the method of least squares, the estimates of the parameters in $\boldsymbol{\beta}$ are obtained as:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \tag{4.2.3}$$

Using the estimate $\hat{\boldsymbol{\beta}}$, the i^{th} predicted response $\hat{\mathbf{y}}_\mathbf{i}$ can be computed as $\hat{\mathbf{y}}_\mathbf{i} = \mathbf{x_i}'$, where $\mathbf{x_i}'$ denotes the i^{th} row vector of the design matrix.

In this simplest of cases, we can see how the matrix algebra works. The term $\mathbf{X}'\mathbf{X}$ would be:

$$[1\ 1\ 1\ 1\ 1\ 1] * \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 1+1+1+1+1+1 = 6 = n \tag{4.2.4}$$

The term $\mathbf{X}'\mathbf{Y}$ would be:

$$[1\ 1\ 1\ 1\ 1\ 1] * \begin{bmatrix} 2 \\ 1 \\ 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} = 2+1+3+4+5+6 = 21 = \sum Y_i \tag{4.2.5}$$

So in this case, the estimate $\mathbf{b}$ as expected is simply the overall mean $= \hat{\mu} = \bar{y}_{..} = 21/6 = 3.5$

Note that the exponent of $\mathbf{X}'\mathbf{X}$ in the formula above indicates arithmetic division as $\mathbf{X}'\mathbf{X}$ is a scalar increase in this case. In the more general setting, the superscript of '-1 ' indicates the inverse operation in matrix algebra.

To perform these matrix operations in SAS IML, we will open a regular SAS editor window, and then copy and paste three components from the file (IML Design Matrices) as shown below.

SAS: Overall Mean Model

Steps in SAS

Step 1

Procedure initiation, and specification of the dependent variable vector, $\mathbf{Y}$.

For our example we have:

```
/* Initiate IML, define response variable */
proc iml;
y={
2,
1,
3,
4,
6,
5};
```

Step 2

We then enter a design matrix $\mathbf{X}$. For the Overall Mean model and our example data, we have:

```
x={
1,
1,
1,
1,
1,
1};
```

Step 3

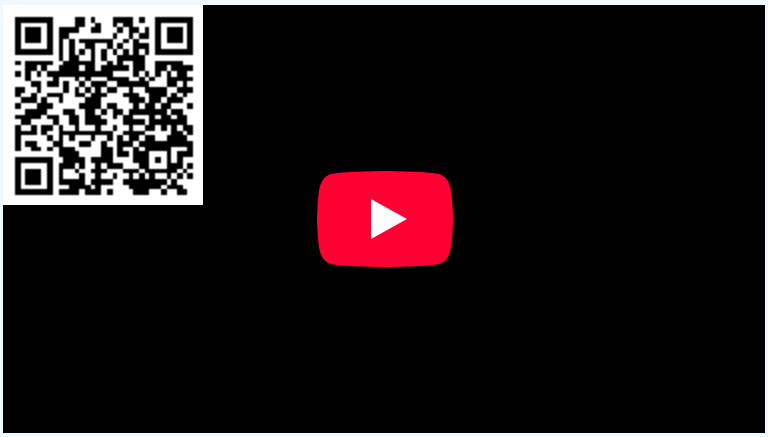
We can now copy and paste a program for the matrix computations to generate results (regression coefficients and ANOVA output):

```
beta=inv(x`*x)*x`*y;
beta_label={"Beta_0","Beta_1","Beta_2","Beta_3"};
print beta [label="Regression Coefficients"
                        rowname=beta_label];
n=nrow(y);
p=ncol(x);
j=j(n,n,1);
ss_tot = (y`*y) - (1/n)*(y`*j)*y;
ss_trt = (beta`*(x`*y)) - (1/n)*(y`*j)*y;
ss_error = ss_tot - ss_trt;
total_df=n-1;
trt_df=p-1;
error_df=n-p;
ms_trt = ss_trt/(p-1);
```

```
ms_error = ss_error / error_df;
F=ms_trt/ms_error;

empty={.};
row_label= {"Treatment", "Error", "Total"};
col_label={"df" "SS" "MS" "F"};
trt_row= trt_df || ss_trt || ms_trt || F;
error_row= error_df || ss_error || ms_error || empty;
tot_row=total_df || ss_tot || empty || empty;
aov = trt_row // error_row // tot_row;
print aov [label="ANOVA"
                        rowname=row_label
                        colname=col_label];
```

Here is a quick video walk-through to show you the process for how you can do this in SAS. (Right-click and select "Show All" if your browser does not display the entire screencast window.)



Video 4.2.1: Walkthrough for ANOVA using the SAS overall mean model.

The program can then be run to produce the following output:

Regression Coefficients	
Beta_0	3.5

ANOVA				
Treatment	DF	SS	MS	F
	0	0		
Error	5	17.5	3.5	
Total	5	17.5		

We see the estimate of the regression coefficient for β_0 equals 3.5, which indeed is the overall mean of the response variable, and is also the same value we obtained above using "by-hand" calculations. In this simple case, where the treatment factor has not entered the model, the only item of interest from the ANOVA table would be the SS_{Error} for later use in the General Linear F-test.

If you like to see the internal calculations further, you may optionally add the following few lines, to the end of the calculation code.

```
/* (Optional) Intermediates in the matrix computations */
xprimex=x`*x; print xprimex;
xprimey=x`
*y; print xprimey;
xprimexinv=inv(x`*x); print xprimexinv;
check=xprimexinv*xprimex; print check;
SumY2=beta`
*(x`*y); print SumY2;
CF=(1/n)*(y`
*j)*y; print CF;
```

This additional code produces the following output:

xprimex	xprimey	xprimeinv
6	21	0.1666667

check	SumY2	CF
1	73.5	73.5

From this we can verify the computations for the $SS_{treatment} = \sum Y_i^2 - \frac{(\sum Y_i)^2}{n} = \sum Y_2 - CF = 0$.

The "check" calculation confirms that $(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X} = 1$, which in fact defines the matrix division operation. In this simple case, it amounts to simple division by n, but in other models that we will work with, the matrix division process is more complicated and is explained here. In general, the inverse of a matrix $\mathbf{A}$, denoted $\mathbf{A}^{-1}$, is defined by the matrix identity $\mathbf{A}^{-1}\mathbf{A} = I$, where I is the identity matrix (a diagonal matrix of 1's). In this example, $\mathbf{A}$ is replaced by $\mathbf{X}'\mathbf{X}$, which is a scalar and equals 6.

R: Overall Mean Model

Steps in R

1. Define response variable and design matrix

```
y<-matrix(c(2,1,3,4,6,5), ncol=1)
x<-matrix(c(1,1,1,1,1,1), ncol=1)
```

2. Regression coefficients

```
beta<-solve(t(x)%*%x)%*%(t(x)%*%y) #3.5
```

3. Calculate the entries of the ANOVA Table

```
n<-nrow(y)
p<-ncol(x)
J<-matrix(1,n,n)
ss_tot = (t(y)%*%y) - (1/n)*(t(y)%*%J)%*%y #17.5
```

```
ss_trt = t(beta)%*%(t(x)%*%y) - (1/n)*(t(y)%*%J)%*%y #0
ss_error = ss_tot - ss_trt #17.5
total_df=n-1 #5
trt_df=p-1 #0
error_df=n-p #5
MS_trt = ss_trt/(p-1)
MS_error = ss_error / error_df #3.5
F=MS_trt/MS_error
```

4. Creating the ANOVA table

```
ANOVA <- data.frame(
c ("","Treatment","Error", "Total"),
c("DF", trt_df,error_df,total_df),
c("SS", ss_trt, ss_error, ss_tot),
c("MS", "", MS_error, ""),
c("F","","",""),
stringsAsFactors = FALSE)
names(ANOVA) <- c(" ", "  ", " ","","")
```

5. Print the ANOVA table

```
print(ANOVA)
# 1           DF   SS  MS F
# 2 Treatment  0    0
# 3     Error  5 17.5 3.5
# 4     Total  5 17.5
```

6. Intermediates in the matrix computations

```
xprimex<-t(x)%*%x # 6
xprimey<-t(x)%*%y # 21
xprimexinv<-solve(t(x)%*%x) # 0.1666667
check<-xprimexinv*xprimex # 1
SumY2<-t(beta)%*%(t(x)%*%y) # 73.5
CF<-(1/n)*(t(y)%*%J)%*%y # 73.5
```