One-way Analysis of Variance Models

1 ANOVA Tables and ANOVA Models

In this part of the course we will learn about ANOVA models, and their application to designed experiments. "ANOVA" is one of the most confusing terms in applied statistics. The term "Analysis of variance (ANOVA)" is used to describe both

- the ANOVA table, which summarizes sums-of-squares and degree-of-freedom information for any linear regression; and
- ANOVA models, which are a stylized way of computing and interpreting regressions with dummy variables to learn about differences in group means for grouped data.

1.1 A bit more about ANOVA Tables

You already know about the ANOVA table used to describe a multiple regression.

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon \tag{1}$$

To simplify the writing a bit, we will write

$$\begin{aligned} SS(X_1, X_2, X_3) &= RegSS, \text{ for the regression } \texttt{Y ~ X}_1 \texttt{ + X}_2 \texttt{ + X}_3 \text{ in (1) above;} \\ RSS(X_1, X_2, X_3) &= RSS, \text{ for the same regression;} \end{aligned}$$

and similarly for other expressions such as $SS(X_1) = RegSS$ for the regression `Y ~ X`$_1$, $SS(X_1, X_2) = RegSS$, for the regression `Y ~ X`$_1$ `+ X`$_2$, etc. We will also write

$$\begin{aligned} SS(X_3|X_2, X_1) &= SS(X_1, X_2, X_3) - SS(X_1, X_2) \\ SS(X_2|X_1) &= SS(X_1, X_2) - SS(X_1) \end{aligned}$$

and so forth (for other subscript combinations). Since

$$TSS = SS(X_1, \ldots, X_k) + RSS(X_1, \ldots, X_k) \tag{2}$$

for least-squares regression, and we know for example that $RSS(X_1, X_2, X_3) \leq RSS(X_1, X_2)$, we know that $SS(X_1, X_2, X_3) \geq SS(X_1, X_2)$; thus expressions like $SS(X_3|X_1, X_2)$ above are guaranteed to be nonnegative. By definition, the quantity $SS(X_3|X_1, X_2)$ represents the incremental variation (in Y) explained by regressing on X_3 after having already regressed on X_1 and X_2. Because of (2), of course, we can also express each $SS(\cdot|\cdots)$ expression as a difference in RSS's.

Sometimes it is more convenient to refer to the regression coefficients β_1, β_2, etc. than the regressors X_1, X_2, etc., so we will also denote

$$\begin{aligned} SS(\beta_1, \beta_2, \beta_3) &= SS(X_1, X_2, X_3) \\ SS(\beta_3|\beta_2, \beta_1) &= SS(X_3|X_2, X_1) \end{aligned}$$

and so on.

The standard ANOVA table that summarizes sums of squares for the regression (1) can then be written as follows:

Source	*SS*	*df*	*MS*
X_1	$SS(X_1)$= 55443.5	1	55443.5
$X_2\|X_1$	$SS(X_2\|X_1)$=167927.2	1	167927.2
$X_3\|X_1, X_2$	$SS(X_3\|X_1, X_2)$= 78792.9	1	78792.9
Regression	$SS(X_1, X_2, X_3)$=302163.4	3	100721.1
Error	$RSS(X_1, X_2, X_3)$=339258.2	$n - 4 = 46$	7375.178
Total	TSS=641421.6	$n - 1 = 49$	

which we can read off almost directly from the SPLUS output

```
> fuel _ read.table("../week04/fuel.dat")
> anova(lm(Fuel ~ Tax + Dlic + Inc,data=fuel))
Analysis of Variance Table

Response: Fuel

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value       Pr(F)
      Tax  1   55443.5  55443.5  7.51758 0.008674482
     Dlic  1  167927.2 167927.2 22.76924 0.000018830
      Inc  1   78792.9  78792.9 10.68352 0.002049453
Residuals 46  339258.2   7375.2
```

You can verify that the definitions of $SS(Inc|Tax, Dlic)$, and so forth, work as indicated above, by separately fitting and examining the RSS's from the models

```
Fuel ~ Tax
Fuel ~ Tax + Dlic
Fuel ~ Tax + Dlic + Inc
```

This sort of ANOVA table is useful in all regression analysis, and in particualr it will be useful as we build and assess the stylized regression models called "ANOVA models".

1.2 ANOVA vs Linear Regression

Someone who prefers thinking about regression models (like me) will often say, *"oh well, ANOVA, that's just multiple regression where all the predictor variables are dummy variables!"*. This is mathematically correct (and we will exploit this idea often!) but ANOVA modeling is more than just running `lm()` with a bunch of dummy variables for $X's$. ANOVA models involve:

- Particular parameterizations of the linear regression model that help us to understand relationships between the categories of discrete regressors; and
- Special arrangements of computing the least-squares estimates and the sums-of-squares in the ANOVA table. These arrangements emphasize the idea that when we are comparing group means in an ANOVA model we are comparing "between" and "within" variance estimates; and they can be much more efficient (for large data sets, e.g.) than the standard linear regression calculations.

1.3 ANOVA

Analysis of variance models (ANOVA models) are used to study the relationship between predictor variables that *are not* necessarily quantitative and response variables that *are* quantitative. They are structured like linear regression models, and have the same basic assumptions, but they do not assume a linear relationship between the predictor and response variables. We've also seen some examples of regressing on dummy variables for categorical predictors. Here I want to introduce the idea that such regressions can also be used to explore nonlinear relationships between *continuous* X's and Y's.

Example.

Suppose we want to study the relationship between $X =$ *the price of a product*, and $Y =$ *the sales volume in one month.*

As a linear regression problem, this is simple. We want to fit the model

$$Y = \alpha + \beta_1 X + \epsilon$$

But this explicitly assumes that the relationship between price and sales is linear, and this need not be the case, as shown in Figure 1 figure (for about 70 months' worth of data).

An alternative to fitting the complex nonparametric regression models we talked about at the beginning of the semester is to choose a few price points, such as \$20.00, \$50.00, and \$90.00, and define dummy variables

$$X_1 = \begin{cases} 1 & \text{if price=\$20.00} \\ 0 & \text{else} \end{cases} \qquad X_2 = \begin{cases} 1 & \text{if price=\$50.00} \\ 0 & \text{else} \end{cases} \qquad X_3 = \begin{cases} 1 & \text{if price=\$90.00} \\ 0 & \text{else} \end{cases}$$

and then fit the regression model

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon \tag{3}$$

As we have seen before, this is an example of a *"one-way ANOVA" model.*

There are three things to note about this approach:

- Even though the model is still the familiar linear regression model, it is possible to detect and explore nonlinearity in the relationship between X and Y, by looking at just these three price levels, or perhaps by adding two or three more price levels to get a better look at the curve. This can be much more efficient than fitting a polynomial or nonlinear regression model.
- It may also be more efficient to collect data for the one-way ANOVA model than for the general regression model; maybe we can get 5 months' worth of data at each of three price points, and have the data collected in a year or so. It is much harder to say how to collect data for this problem if all possible prices are allowed.
- All ANOVA models are overparametrized in their "natural" form, and we often must make a choice about what a sensible "reduced" parametrization is. □

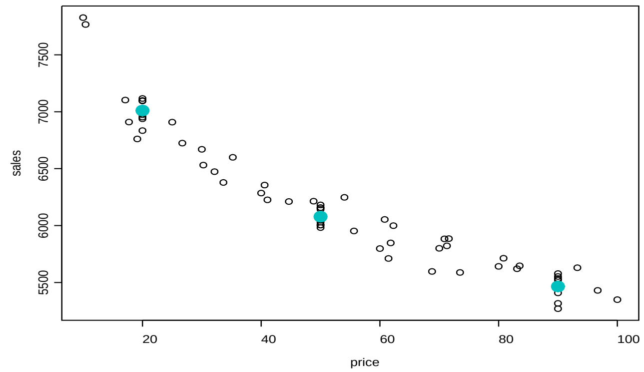

Figure 1: Some fictional sales data. We could fit the linear regression $y = \alpha + \beta X + \epsilon$ to all the data, but instead we will focus on the particular price points \$20.00, \$50.00, and \$90.00. (In real life, if one knew one were only going to focus on these prices, of course one would not bother to collect data at the other price points.)

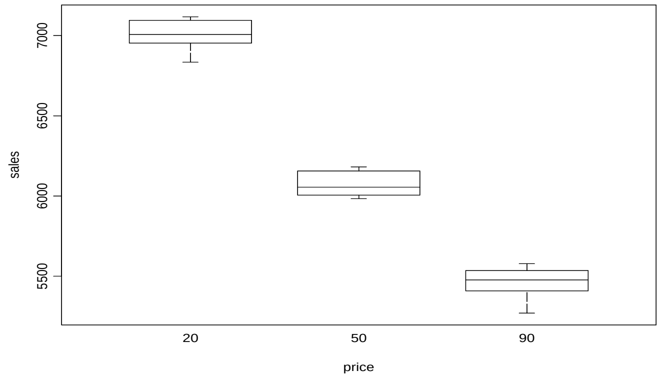

Figure 2: The result of `plotfactor()` is always some sort of parallel boxplot display.

2 One-way ANOVA analysis in SPLUS

We can illustrate the one-way analysis of variance using the sales data:

```
> attach(salesdata)
> sales205090 _ salesdata[((price==20)|(price==50)|(price==90)),]
> sales205090$price _ factor(sales205090$price)
> table(sales205090$price)  # There are 11 months of sales at each price
 20 50 90
 11 11 11

> view(sales205090$price)

      structure(.Data = c(1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3,
      1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3), .Label =
      c("20", "50", "90"), class = "factor")

> plot.factor(sales205090,style.bxp="old")
```

There is a special function, `aov()` for fitting analysis of variance models that Splus can apply to data where the regressors are discrete factors. `aov()` can calculate a regression more efficiently in some cases than `lm()`, and `aov()` retains information that is useful for ANOVA analyses, that `lm()` does not. Let's compare the `aov()` output with the usual `lm()` output; see Table 1.

This tells us that the "price" variable explains much of the variability in the "sales" data, but isn't very informative about details, like what the mean sales was at each price level, etc.

2.1 The cell-means parametrization

As we have seen before, to calculate the mean sales at each price we want to think of a model like this:

$$y_{ij} = \mu_j + \epsilon_{ij}$$

where $j = 1, 2$ or 3, depending on whether the price was 20, 50 or 90, and i indexes the 11 observations at each price level[1]. This is called the *cell means* ANOVA parametrization; and it is equivalent to the regression model

$$y_i = \mu_1 X_1 + \mu_2 X_2 + \mu_3 X_3 + \epsilon_i$$

where now i indexes all the data, and the X_j's indicate whether the price was 20, 50 or 90.

There is a function, `model.tables()`, in Splus, that will calculate the μ_i's directly for us:

[1]In general in ANOVA models we write multiple subscripts like this to help organize our view of the data into observations (j) within cells (i) In more complicated models we may have to change from two to three, or more, subscripts.


```
> sales.aov _ aov(sales ~ price,
+ data=sales205090)
> sales.aov
Call:
   aov(formula = sales ~ price,
      data = sales205090)

Terms:
                   price Residuals
 Sum of Squares 13332276    223371
Deg. of Freedom        2        30

Residual standard error: 86.28854
Estimated effects are balanced
> summary(sales.aov)
         Df Sum of Sq Mean Sq  F Value Pr(F)
   price  2 13332276 6666138 895.2989     0
Residuals 30   223371    7446

> anova(sales.aov)
Analysis of Variance Table

Response: sales

Terms added sequentially (first to last)
         Df Sum of Sq Mean Sq  F Value Pr(F)
   price  2  13332276 6666138 895.2989     0
Residuals 30   223371    7446
>  13332276/( 13332276 +  223371 )
+ # calculate R^2
[1] 0.9835219
```

```
>  sales.lm _ lm(sales ~ price,data=sales205090)
> sales.lm
Call:
lm(formula = sales ~ price, data = sales205090)

Coefficients:
 (Intercept)    price1    price2
    6184.607 -466.4094 -359.8494

Degrees of freedom: 33 total; 30 residual
Residual standard error: 86.28854
> summary(sales.lm)

Call: lm(formula = sales ~ price, data = sales205090)
Residuals:
    Min     1Q Median    3Q   Max
 -194.8 -55.67 -4.224 78.53 113.8

Coefficients:
                Value Std. Error    t value   Pr(>|t|)
(Intercept)  6184.6074    15.0209   411.7333     0.0000
     price1  -466.4094    18.3968   -25.3528     0.0000
     price2  -359.8494    10.6214   -33.8797     0.0000

Residual standard error: 86.29 on 30 degrees of freedom
Multiple R-Squared: 0.9835
F-statistic: 895.3 on 2 and 30 degrees of
freedom, the p-value is 0

Correlation of Coefficients:
       (Intercept) price1
price1 0
price2 0           0
> anova(sales.lm)
Analysis of Variance Table

Response: sales

Terms added sequentially (first to last)
          Df Sum of Sq Mean Sq  F Value Pr(F)
    price  2  13332276 6666138 895.2989     0
Residuals 30    223371    7446
```

Table 1: Comparison of `aov()` and `lm()` output for the fictional sales data.

```
> model.tables(sales.aov,type="means")
Refitting model to allow projection
Tables of means
Grand mean

 6185

 price
   20   50   90
 7011 6078 5465
```

Let's compare these results with fitting the model "by hand" using `lm()`.

```
> lmsales _ sales205090$sales
> lmprice _ sales205090$price
> p20 _ ifelse(lmprice==20,1,0)
> p50 _ ifelse(lmprice==50,1,0)
> p90 _ ifelse(lmprice==90,1,0)

> sales.lm _ lm(lmsales ~ p20 + p50 + p90 - 1)
> summary(sales.lm)

Call: lm(formula = lmsales ~ p20 + p50 + p90 - 1)
Residuals:
    Min     1Q Median    3Q   Max
 -194.8 -55.67 -4.224 78.53 113.8

Coefficients:
        Value Std. Error   t value  Pr(>|t|)
p20 7010.8662   26.0170   269.4728    0.0000
p50 6078.0475   26.0170   233.6185    0.0000
p90 5464.9085   26.0170   210.0517    0.0000

Residual standard error: 86.29 on 30 degrees of freedom
Multiple R-Squared: 0.9998
F-statistic: 57100 on 3 and 30 degrees of freedom, the p-value is 0

Correlation of Coefficients:
    p20 p50
p50 0
p90 0   0
```

Actually, as we know, when `lm()` or `aov()` sees a `factor` variable in a regression, it will construct dummy variables automatically, but by default it produces a coding that is difficult for humans to interpret (this coding also makes the regressors *orthogonal*—so like our by-hand analysis the correlations between the regression coefficients will be zero—but there is more than one way to do this). We will learn more about Splus's coding of dummy variables later.

2.2 The grand-mean-plus-effects parametrization

Often we are not directly interested in the mean of each cell in an ANOVA model, but rather in whether there are substantial differences between the cells. An alternative parametrization focuses the attention away from the overall mean of the data and toward deviations from the overall mean in each cell. In ANOVA notation, we would write the model (3) as

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

where again $j = 1, 2$ or 3, depending on whether the price was 20, 50 or 90, and i indexes the 11 observations at each price level.

Nomenclature:

- μ is called the *overall mean* or *grand mean*
- α_1, α_2 and α_3 are called "effects". Just to be confusing, they are also called "contrasts."

Note that there are four parameters but only three groups of data—cells—from which to estimate them.

In the past we have fixed this problem by dropping a regressor, which is equivalent to adding the linear constraint "(that coefficient) = 0". More general linear constraints, of the form

$$w_0\mu + \sum_{j=1}^{\#cells} w_j\alpha_j = 0,$$

where $w_0, w_1, \ldots$ are fixed, known weights, will also fix the problem. The usual linear constraint for the "grand means plus effects" model is to make the effects sum to zero:

$$\sum_{j=1}^{\#cells} \alpha_j = 0$$

(so $w_0 = 0$, and the other w_j's = 1). The $\alpha's$ can again be estimated with the `model.tables()` function:

```
> model.tables(sales.aov,type="effects")
Refitting model to allow projection
Tables of effects

 price
    20     50     90
 826.3 -106.6 -719.7
> sum(scan())        # verify that the alpha's sum to zero
1:  826.3 -106.6 -719.7
4:
[1] -1.136868e-013
```

How can we think about what `model.tables()` is doing for the "grand mean plus effects" parametrization?

It is generally quite useful to start with the "cell means" parametrization expressed as a *linear regression* model, impose the new parameterization on it, and see what linear regression is required to estimate the parameters in the ANOVA model.

We begin by computing the cell means in the grand mean plis effects model, and noting the linear restriction:

$$\begin{aligned} \mu_1 &= \mu + \alpha_1 \\ \mu_2 &= \mu + \alpha_2 \\ \mu_3 &= \mu + \alpha_3, \text{ and} \\ 0 &= \alpha_1 + \alpha_2 + \alpha_3 \end{aligned}$$

Now we write out the regression model for cell means, substitute the values for the cell means above, and use the linear constraint:

$$\begin{aligned} y &= \mu_1 X_1 + \mu_2 X_2 + \mu_3 X_3 \\ &= (\mu + \alpha_1) X_1 + (\mu + \alpha_2) X_2 + (\mu + \alpha_3) X_3 + \epsilon \\ &= (\mu + \alpha_1) X_1 + (\mu + \alpha_2) X_2 + (\mu - \alpha_1 - \alpha_2) X_3 + \epsilon \\ &= \mu + \alpha_1 (X_1 - X_3) + \alpha_2 (X_2 - X_3) + \epsilon \end{aligned}$$

This suggests that to estimate the effects in the grand mean plus effects model we should construct new dummies

$$D_1 = X_1 - X_3 = \begin{cases} 1, & \text{if price} = 20 \\ 0, & \text{if price} = 50 \\ -1, & \text{if price} = 90 \end{cases}$$

$$D_2 = X_2 - X_3 = \begin{cases} 0, & \text{if price} = 20 \\ 1, & \text{if price} = 50 \\ -1, & \text{if price} = 90 \end{cases}$$

and fit the regression

$$y = \mu + \alpha_1 D_1 + \alpha_2 D_2 + \epsilon$$

Let's try it:

```
> d1 _ p20-p90
> d2 _ p50-p90
>  sales.lm _ lm(lmsales ~ d1 + d2)
> summary(sales.lm)

Call: lm(formula = lmsales ~ d1 + d2)
Residuals:
    Min     1Q Median    3Q   Max
 -194.8 -55.67 -4.224 78.53 113.8

Coefficients:
                 Value Std. Error    t value   Pr(>|t|)
(Intercept)  6184.6074    15.0209   411.7333     0.0000
         d1   826.2588    21.2428    38.8960     0.0000
         d2  -106.5599    21.2428    -5.0163     0.0000

Residual standard error: 86.29 on 30 degrees of freedom
```



```
Multiple R-Squared: 0.9835
F-statistic: 895.3 on 2 and 30 degrees of freedom, the p-value is 0

Correlation of Coefficients:
   (Intercept)   d1
d1  0.0
d2  0.0        -0.5
```

what about α_3? Since $\alpha_1 + \alpha_2 + \alpha_3 = 0$ (and since the mle of a function is the function of the mle's), we know it can be calculated as

$$\hat{\alpha}_3 = -\hat{\alpha}_1 - \hat{\alpha}_2 = -826.2588 + 106.5599 = -719.6998$$

The SE of $\hat{\alpha}_3$ can also be found from the variance-covariance matrix of $\hat{\alpha}_1$ and $\hat{\alpha}_2$ using a bit of matrix algebra, but we will leave that for another day.

3 A more realistic one-way ANOVA example

3.1 Initial Analysis and ANOVA table: cell means model

The following data is from Box, Hunter and Hunter (1978) and is also analyzed in Chapter 13 of the SPLUS Guide to Statistics. It gives blood coagulation times for each of four diets.

Diet	A	B	C	D
Coag	62	63	68	56
Times	60	67	66	62
	63	71	71	60
	59	64	67	61
		65	68	63
		66	68	64
				63
				59

Blood Coagulation Times

There are many ways to enter this data. In our case I will use NA (Splus's code for "missing data") as a placeholder in order to start with a square array, I'll attach the diet labels, and then delete the NA's since they are not part of this data set (the data isn't really missing, it was never intended to be collected).

```
> coag.times _ scan()
1:  62 63 68 56
5:  60 67 66 62
9:  63 71 71 60
13:  59 64 67 61
17:  NA 65 68 63
21:  NA 66 68 64
25:  NA NA NA 63
29:  NA NA NA 59
```

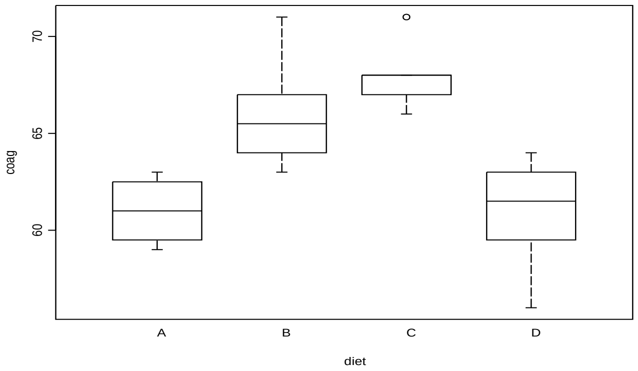

Figure 3: Boxplots of coagulation times by diet for the blood coagulation data.

```
33:
> diet _ factor(rep(c("A","B","C","D"),8))
> split(coag.times,diet)  # check that the factor labels are right
$A: [1] 62 60 63 59 NA NA NA NA
$B: [1] 63 67 71 64 65 66 NA NA
$C: [1] 68 66 71 67 68 68 NA NA
$D: [1] 56 62 60 61 63 64 63 59
> coag _ data.frame(coag=coag.times,diet=diet) # make the data frame
> coag _ coag[!is.na(coag$coag),] # delete rows with NA's
> split(coag$coag,coag$diet) # recheck that factor labels are still OK
$A: [1] 62 60 63 59
$B: [1] 63 67 71 64 65 66
$C: [1] 68 66 71 67 68 68
$D: [1] 56 62 60 61 63 64 63 59
> sapply(.Last.value,mean)
  A  B  C  D
 61 66 68 61
> plot.factor(coag,style.bxp="old")
```

Now we fit the model, look at some diagnostic plots, and consider the analysis of variance table. More details on the SPLUS parts of the problem can be found in Chapter 13 of the SPLUS Guide to Statistics.

```
> coag.aov _ aov(coag ~ diet, data=coag)
> par(mfrow=c(2,3))
```


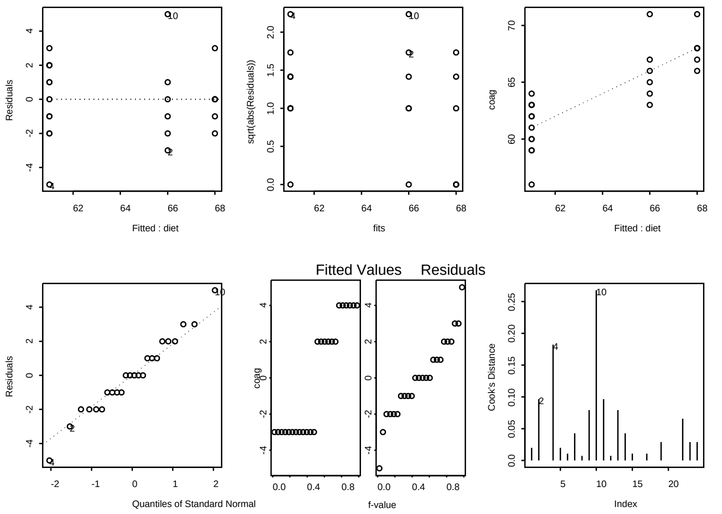

Figure 4: Standard diagnostic plots for one-way ANOVA for the blood coagulation data.

```
> plot(coag.aov)

> model.tables(coag.aov,type="means")
Refitting model to allow projection
Tables of means
Grand mean

 64

 diet
     A  B  C  D
    61 66 68 61
rep  4  6  6  8
```

The cell means $Y_{i.}$ and the grand mean $Y_{..}$ are illustrated in the figure below; the cell means $\bar{Y}_{i.}$ are the estimates of the μ_i's in the *cell means model*

$$Y_{ij} = \mu_i + \epsilon_{ij}, \quad i = 1, \ldots, \#cells, \quad J = 1, \ldots, n_i$$

where k is the number of cells, and there are n_i observations in the i^{th} cell. As usual in regression, the error terms (ϵ_{ij} in this case) are distributed $N(0, \sigma^2)$ for some unknown error variance σ^2.

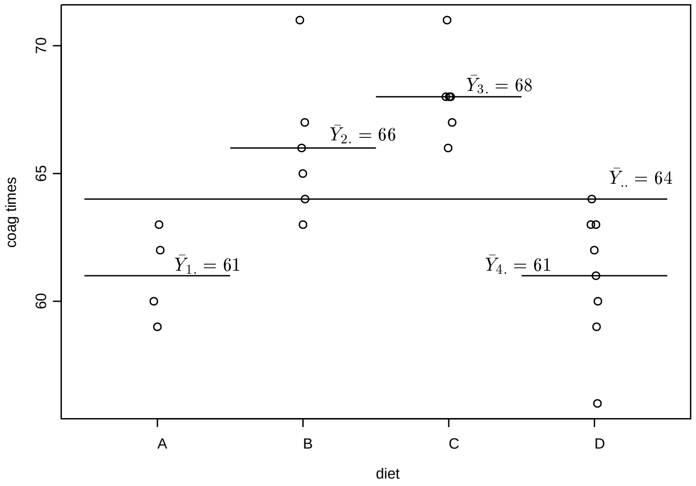

Figure 5: Illustration of the fitted cell means model for the blood coagulation data.

> Note: The roles of i are j are reversed in this example: i now stands for the group or cell, and j stands for individual observations within a group or cell.

Now recall that we can write the deviation of a single observation from the grand mean as

$$(Y_{ij} - Y_{..}) = (Y_{ij} - Y_{i.}) + (Y_{i.} - Y_{..})$$

If we square and sum these terms, the magic of orthogonal sums of squares tells us

$$\sum_i \sum_j (Y_{ij} - Y_{..})^2 = \sum_i \sum_j (Y_{ij} - Y_{i.})^2 + \sum_i n_i \cdot (Y_{i.} - Y_{..})^2$$

with degrees of freedom $n-1$, $n-k$ and $k-1$ respectively. This gives rise to the ANOVA table

Source	*SS*	*df*	*MS*
Factor	RegSS=$\sum_i n_i \cdot (Y_{i.} - Y_{..})^2$	$k-1$	RegMS=$RegSS/(k-1)$
Error	RSS=$\sum_i \sum_j (Y_{ij} - Y_{i.})^2$	$n-k$	RMS=$RSS/(n-k)$
Total	TSS=$\sum_i \sum_j (Y_{ij} - Y_{..})^2$	$n-1$	TMS=$TSS/(n-1)$

Here is SPLUS's ANOVA table for this ANOVA model.

```
> anova(coag.aov)     # some output omitted below
```



```
Analysis of Variance Table

Response: coag

Terms added sequentially (first to last)
          Df Sum of Sq Mean Sq  F Value         Pr(F)
     diet  3       228    76.0 13.57143 0.00004658471
Residuals 20       112     5.6
>  228/(228+ 112) # R^2
[1] 0.6705882
```

The F statistic for testing whether the factor explains the variation in Y is

$$F^* = \frac{RegMS}{RMS} = \frac{(\text{between SS})/(k-1)}{(\text{within SS})/(n-k)}$$

Under the null hypothesis

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k$$

F^* is distributed as $F(k-1, n-k)$. Large values of F^* argue in favor of

$$H_a : \text{not all } \mu_i \text{ are equal}$$

We can also look at the effects (deviations from means), either by direct calculation or with `model.tables()`:

```
> model.tables(coag.aov,type="effects")
Refitting model to allow projection
Tables of effects

 diet
     A B C  D
    -3 2 4 -3
rep  4 6 6  8


> x1 _ ifelse(coag$diet=="A",1,0)
> x2 _ ifelse(coag$diet=="B",1,0)
> x3 _ ifelse(coag$diet=="C",1,0)
> x4 _ ifelse(coag$diet=="D",1,0)
> d1 _ x1-x4
> d2 _ x2-x4
> d3 _ x3-x4
> summary(lm(coag$coag ~ x1+x2+x3+x4 - 1))

Call: lm(formula = coag$coag ~ x1 + x2 + x3 + x4 - 1)
Residuals:
 Min    1Q     Median   3Q Max
  -5 -1.25 -4.36e-016 1.25   5
```

```
Coefficients:
     Value Std. Error t value Pr(>|t|)
x1 61.0000  1.1832    51.5544  0.0000
x2 66.0000  0.9661    68.3165  0.0000
x3 68.0000  0.9661    70.3867  0.0000
x4 61.0000  0.8367    72.9089  0.0000

Residual standard error: 2.366 on 20 degrees of freedom
Multiple R-Squared: 0.9989
F-statistic: 4399 on 4 and 20 degrees of freedom, the p-value is 0

Correlation of Coefficients:
   x1 x2 x3
x2 0
x3 0  0
x4 0  0  0
> summary(lm(coag$coag ~ d1+d2+d3))

Call: lm(formula = coag$coag ~ d1 + d2 + d3)
Residuals:
 Min    1Q     Median   3Q Max
  -5 -1.25 1.249e-016 1.25   5

Coefficients:
                Value Std. Error   t value  Pr(>|t|)
(Intercept)   64.0000    0.4979   128.5367    0.0000
         d1   -3.0000    0.9736    -3.0813    0.0059
         d2    2.0000    0.8453     2.3659    0.0282
         d3    4.0000    0.8453     4.7319    0.0001

Residual standard error: 2.366 on 20 degrees of freedom
Multiple R-Squared: 0.6706
F-statistic: 13.57 on 3 and 20 degrees of freedom, the p-value is 0.00004658

Correlation of Coefficients:
   (Intercept)      d1      d2
d1  0.2106
d2 -0.0346     -0.4075
d3 -0.0346     -0.4075 -0.3061

> -( - 3 + 2 + 4 )
[1] -3
```