Adapted from P. Smith, and expanded

11 Factors, ANOVA, and Regression: SAS versus Splus

Factors. A *factor* is a variable with finitely many values or *levels* which is treated as a predictor within regression-type models. Factors can be used as qualitative predictors by introducing $k-1$ dummy variables to represent a factor with k levels. The objective is to parameterize a so-called *analysis of variance model*

$$y_{ij} \;=\; \mu + \alpha_i + \epsilon_{ij}, \quad i = 1, \ldots, k, \quad j = 1, \ldots, r \tag{1}$$

as a regression model. Imagine the response-variables y_{ij} as a vector $\mathbf{Y} = (Y_\nu)$ of length $n = Ir$, and that F_ν is a categorical variable (which can take on the possible levels $f_1, \ldots, f_k$) . We express $\mathbf{Y}$ linearly in terms of the vector $\mathbf{1}$ of n ones and of $k-1$ additional columns $C^{(l)}$, $l = 1, \ldots, k-1$:

$$\mathbf{Y} \;=\; b_0\,\mathbf{1} + \sum_{l=1}^{k-1} b_l\, C^{(l)}$$

What we want to achieve by this is that the mean of an observation Y_ν for which $F_\nu = l$ has the two equivalent (using (1)) representations

$$\mu + \alpha_l \;=\; b_0 + \sum_{i=1}^{k-1} b_i\, C_\nu^{(i)} \tag{2}$$

The simplest coding of dummy variables, in a dataset with observation-index $\nu = 1, \ldots, n$ and categorical variable F_ν (which can take on the possible levels $f_1, \ldots, f_k$) is to define columns

$$C_\nu^{(l)} \;=\; I_{[F_\nu = f_l]} \quad , \qquad \nu = 1, \ldots, n$$

for each of $l = 1, \ldots, k-1$. In that case, from (2) we learn that

$$\mu + \alpha_l \;=\; b_0 + b_l \quad \text{if} \qquad l < k \quad , \qquad \text{and otherwise} \qquad = b_0$$

As its default, **Splus** uses so-called *Helmert* contrasts to define columns. To show the pattern it uses, consider (in the case $k = 5$) the matrix with entry defining $C_\nu^{(l)}$ whenever F_ν is equal to the row-index $i = 1, \ldots k-1$, and the column-index l also ranges from 1 to $k-1$:

```
-1   -1   -1   -1
 1   -1   -1   -1
 0    2   -1   -1
 0    0    3   -1
 0    0    0    4
```

Another scheme for coding $C_{\nu}^{(l)}$ is designated **contr.sum** and in the case $k = 5$ has matrix

```
 1    0    0    0
 0    1    0    0
 0    0    1    0
 0    0    0    1
-1   -1   -1   -1
```

replacing the previous one.

Note that the *contrasts* chosen for a **factor** are associated with the factor at the time of its creation. So for example, if **aclass** is a vector with character components "a", "b", "c", "d", "e" which we want to turn into a factor **aclass** with **contr.sum** contrasts, then we use the command

```
> aclass <- C(factor(aclass), contr=contr.sum)
```

Then whenever **aclass** is used as a categorical predictor in a linear model in **Splus**, the **contr.sum** contrasts wqill be used to encode it.

While it would be clumsy to create and code dummy variables to represent factors 'by hand', in either SAS or Splus, this construction is done automatically in linear-model fitting with factor predictors, by **lm** or **glm** in Splus and by PROC ANOVA or PROC GLM in SAS.

(*Note:* As mentioned in class, PROC ANOVA does not work correctly in all problems involving qualitative predictors. PROC GLM will give correct results and is considerably more powerful than PROC ANOVA. You should *not* use PROC ANOVA except for models with "balance" and on systems where conserving memory is a prime consideration. All of the PROC ANOVA coding in Cody and Smith will work identically under PROC GLM.)

11.1 Analysis of Artificial Dataset

We proceed to generate a small simulated dataset in Splus, with which we can illustrate various aspects of linear model fitting — especially the Analysis of Variance and associated Tables of sums of squares — in Splus and SAS with categorical predictors.

```
Small Log to get Started with Splus Using Factors in Anova
=====================================================4/7/03
> alphvec <- runif(5)
  alphvec <- alphvec - mean(alphvec)
> round(alphvec,6)
[1]  0.020548  0.032814  0.109756 -0.387467  0.224349

> aclass <- C(factor(rep(letters[1:5],rep(20,5))), contr=contr.sum)
> levels(aclass)
[1] "a" "b" "c" "d" "e"
> attr(aclass,"levels")
[1] "a" "b" "c" "d" "e"
> Xv <- rnorm(100)
  Zv <- 3*(runif(100)-0.4)
> Ydat <- 1.37 + rep(alphvec, rep(20,5)) +
     2*Xv - Zv + rnorm(100)*0.75

> Ylm <- lm(Ydat ~ aclass + Xv + Zv)
> round(Ylm$coef,6)
 (Intercept) aclass1   aclass2  aclass3   aclass4       Xv       Zv
    1.376631 0.00979 -0.080907 0.032477 -0.323643 1.927776 -0.99445

### This is OK: recall that the true coeffficient for Xv was 2, and
###   for Zv  was -1: but how do we recover the ‘class-mean’
### parameters alpha originally generated in the vector alphvec ?
> Ylm$contrasts
$aclass:
  [,1] [,2] [,3] [,4]
a    1    0    0    0
b    0    1    0    0
c    0    0    1    0
d    0    0    0    1
e   -1   -1   -1   -1
```

```
### This shows how Splus coded the classes, as specified for aclass.
  To see this in a more explicit and intepretable way, recall that
  observations 10,30, 50, 70, 90 respectively have levels
  "a","b","c","d","e":

> model.matrix(Ylm)[seq(10,90,20),]
   (Intercept) aclass1 aclass2 aclass3 aclass4         Xv         Zv
10           1       1       0       0       0 -0.6116110  0.1064593
30           1       0       1       0       0 -1.3854599  0.9958901
50           1       0       0       1       0  1.5509195 -0.6121052
70           1       0       0       0       1 -0.9070166  1.7867886
90           1      -1      -1      -1      -1  1.0122793  1.6386513

> sum(abs(model.matrix(Ylm) %*% Ylm$coef - Ylm$fitted))
[1] 2.375877e-14

##  Thus in terms of the coefficients of the model, the (estimated)
##  means for the five classes of  aclass   should be found as
##  follows:

> c(model.matrix(Ylm)[seq(10,90,20),1:5] %*% Ylm$coef[1:5]
[1] 1.386422 1.295724 1.409109 1.052988 1.738914

> .Last.value - mean(.Last.value)
[1]  0.009790442 -0.080907097  0.032477362 -0.323643407  0.362282700
### THIS IS THE SET OF ESTIMATED CLASS-MEAN CONTRASTS
##      (which sum to 0)

> alphvec
[1]  0.02054783  0.03281407  0.10975606 -0.38746695  0.22434899
### AND THIS WAS THE SET OF TRUE CLASS-MEAN CONTRASTS
### So there is not too bad a correspondence.
```

Also, note that we did not really need to fix or know the contrasts actually used in coding the factors in order to display the means for the five **aclass**-determined groups. All we needed was to apply the model.matrix in a set of rows with known **aclass** level to the fitted coefficients ! Similarly, we can find the standard-errors for the fitted class-means, as follows:

```
> round(sqrt(diag(model.matrix(Ylm)[seq(10,90,20),1:5] %*%
  summary(Ylm)$cov.unscaled[1:5,1:5] %*% t(model.matrix(Ylm)[
  seq(10,90,20),1:5]))),4)
[1] 0.2355 0.2330 0.2269 0.2402 0.2242
```

So none of the estimated class-mean contrasts was actually significantly different from 0 in this example ! However, there certainly **were** group differences with respect to raw means:

```
> round(apply(matrix(Ydat, ncol=5),2,mean),4)
[1] 1.7012 0.9494 1.1916 0.2058 2.0572
> round(apply(matrix(Xv, ncol=5),2,mean),4)
[1]  0.3987  0.0844  0.0435 -0.0797  0.1340
> round(apply(matrix(Zv, ncol=5),2,mean),4)
[1]  0.4564  0.5120  0.3031  0.6973 -0.0604
```

We will refer to these means later on, in considering the SAS output.

```
    ### Now proceed to Analysis of Variance
> anova(Ylm)                  ### For comparison with SAS below
Analysis of Variance Table

Response: Ydat

Terms added sequentially (first to last)
          Df Sum of Sq  Mean Sq  F Value        Pr(F)
aclass     4   40.7006  10.1751  15.6473 7.980251e-10
Xv         1  304.3097 304.3097 467.9659 0.000000e+00
Zv         1   62.6572  62.6572  96.3539 6.000000e-16
Residuals 93   60.4762   0.6503

> sum(Ylm$resid^2)
[1] 60.47621
> sum(lm(formula = Ydat ~ aclass + Xv)$resid^2)
[1] 123.1334    ### = 60.4762 + 62.6572
> sum(lm(formula = Ydat ~ aclass)$resid^2)
[1] 427.4432    ### = 123.1334 + 304.3097
> sum((Ydat-mean(Ydat))^2)
[1] 468.1437    ### = 427.4432 + 40.7006
```

The cumulative sums of squares in the ANOVA Table, **from the bottom up** starting with the final-model sum of squared residuals, can always be interpreted as the sum of squared residuals from earlier stages of the model-fitting !

Finally, the estimated variance $\hat{\sigma}^2$ is always obtained as the residual sum of squares divided by the number of observations reduced by number of estimated coefficients.

```
> c(summary(Ylm)$sigma^2, Ylm$df, sum(Ylm$resid^2)/Ylm$df)
[1] 0.6502819  93.0000000  0.6502819
```

```
> write.table(cbind(Ydat= round(Ydat,7), Xv = round(Xv,7), Zv =
      round(Zv,7), aclass), file="Yanov.dat",  sep="  ")
> !more Yanov.datYdat  Xv  Zv  aclass
-0.8356545  -0.121606  0.7943222  1
-0.0719519  0.3084737  1.229228  1
-0.850251  -1.0913033  -0.1884368  1
5.0168243  2.123058  0.3399656  1
3.280047  1.4878834  1.648248  1
-0.2809213  -0.204657  1.6208564  1
4.1190551  1.2060654  0.7857575  1
2.0874594  0.073946  -0.8112233  1
## Copied into emacs ASCII file to save, before invoking SAS.
##  or ftp'd to ASCII file in cluster-machine.
```

Now we continue in SAS with the same data. The main new element is PROC GLM, which makes use of the factor-variable **aclass** through the declaration-statement: CLASS aclass ; the class means are requested through the MEANS and LSMEANS statement-lines. We will take some pains below to interpret the differences between the class-mean outputs generated from these two commands.

```
    libname SASstf "SASproj" ;
  data SASstf.Yanov ;
  infile "Yanov.dat" ;
     input Ydat Xv Zv aclass $ ;
     if _N_>1 ;  /* to skip first line */
  run;
   /* Now have created permanent dataset "yanov.sas7bdat" */
  proc glm data= SASstf.Yanov ;
     class aclass;
      model Ydat = aclass Xv Zv ;
      means aclass / duncan ;
        /* Duncan relates to the grouping and clustering of the
            class means. But if we want the class-mena parameters
            adjusted for the effects of the continuous predictors
            Xv, Zv, then we need an LSMEANS statement instead !
       LSMEANS aclass;
  run;
```

The edited SAS follow, for comparison with the quantities previously estimated in **Splus**. First comes the 'core' output from PROC GLM.

```
               SAS OUTPUT, edited

  Dependent Variable: Ydat
...
                     Sum of
Source        DF    Squares     Mean Square    F Value    Pr > F

Model          6    407.66751    67.9446       104.48      <.0001
Error         93     60.47622     0.6503
Corr. Tot.    99    468.14373

R-Square     Coeff Var      Root MSE     Ydat Mean

0.870817      66.04187      0.806401      1.221044


Source        DF    Type I SS    Mean Square    F Value    Pr > F
aclass         4     40.7006      10.1751        15.65    <.0001
Xv             1    304.3097     304.3097       467.97    <.0001
Zv             1     62.6572      62.6572        96.35    <.0001


Source        DF    Type III SS  Mean Square    F Value    Pr > F
aclass         4      4.4248       1.1062         1.70    0.1564
Xv             1    294.5692     294.5692       452.99    <.0001
Zv             1     62.6572      62.6572        96.35    <.0001
```

Next comes the output from the MEANS line of PROC GLM: these group-means are **not** adjusted for the effects of the precitor-variables Xv, Zv !

```
          Duncan's Multiple Range Test for Ydat

NOTE: This test controls the Type I comparisonwise error rate,
not the experimentwise error rate.     ...
Means with the same letter are not significantly different.

 Duncan Grouping     Mean     N   aclass

        A          2.0572     20    5
        A
        A          1.7012     20    1

        B          1.1916     20    3
        B
        B          0.9494     20    2

        C          0.2058     20    4


              Least Squares Means
            aclass     Ydat LSMEAN

            1           1.23083484
            2           1.14013731
            3           1.25352177
            4           0.89740100
            5           1.58332711
```

Note that these class means are **not** exactly the same as the numbers previously calculated in Splus: those were 1.386422, 1.295724, 1.409109, 1.052988, 1.738914. However, after subtracting the grand mean of `Ydat`, which is 1.221044, from all of these adjusted class means, we get the class-contrast valueswhich were previously estimated in Splus. The discrepancy between the SAS-estimated LSmean numbers and the numbers we estimated in Splus is due to the contribution to empirical mean of the Xv and Zv terms in the model-fit: the estimated terms $1.927776 * Xv - 0.994450 * Zv$ have empirical mean -0.155587, which is also equal to 1.386422-1.2308348.