ORIGIN ≡ 0

General Linear Models and "dummy" Coding

Coding of Groups as "dummy" Variables:

Multiple Linear Regression is not restricted to cardinal data, but may be readily adapted to other forms of data. In this guise - the so called "General Linear Model" or "GLM", is a very wide-ranging method that can be shown to unify much of standard statistics including t-tests and ANOVA. Shown here is one standard extension to independent variables *in data classes* (as in t-tests or ANOVA) through the judicious use of "dummy coding". Because all variables can be "binned" into data classes (similar to making a histogram), this approach has broad application.

Two sample t-Test with assumed equal variances
See *Biostatistics* Worksheet 160

Zar Example 8.1

	data	group
1	8.8	gpB
2	8.4	gpB
3	7.9	gpB
4	8.7	gpB
5	9.1	gpB
6	9.6	gpB
7	9.9	gpG
8	9	gpG
9	11.1	gpG
10	9.6	gpG
11	8.7	gpG
12	10.4	gpG
13	9.5	gpG

Two-Sample t-Test assuming Equal Variances:
Output from R:

```
#COMPARING TWO-SAMPLE T-TEST WITH GLM:
ZAR=read.table("c:/DATA/Biostatistics/ZarEX8.1R.txt")
ZAR
attach(ZAR)
Y=data
X=group

#PERFORMING TWO-SAMPLE TWO-SIDED t-TEST:
t.test(Y~X,alternative="two.sided",var.equal=TRUE, conf.level=0.95)
detach(ZAR)
```

```
        Two Sample t-test

data:  Y by X
t = -2.4765, df = 11, p-value = 0.03076
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.8752609 -0.1104534
sample estimates:
mean in group gpB mean in group gpG
         8.750000          9.742857
```

Zar Example 8.1

```
    data group
1    8.8     0
2    8.4     0
3    7.9     0
4    8.7     0
5    9.1     0
6    9.6     0
7    9.9     1
8    9.0     1
9   11.1     1
10   9.6     1
11   8.7     1
12  10.4     1
13   9.5     1
```

Regression Approach: with "dummy" Variable for group with levels: gpB=0 & gpG=1

```
#PERFORMING EQUIVALENT GLM  REGRESSION:
ZAR2=read.table("c:/DATA/Biostatistics/ZarEX8.1bR.txt")
ZAR2
attach(ZAR2)
Y=data
X=group
summary(lm(Y~X))
detach(ZAR2)
```

```
> summary(lm(Y~X))

Call:
lm(formula = Y ~ X)

Residuals:
   Min     1Q Median     3Q    Max
-1.043 -0.350 -0.050  0.350  1.357

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.7500     0.2942  29.743 7.32e-12 ***
X             0.9929     0.4009   2.476   0.0308 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7206 on 11 degrees of freedom
Multiple R-squared: 0.358,      Adjusted R-squared: 0.2996
F-statistic: 6.133 on 1 and 11 DF,  p-value: 0.03076
```

t statistic & P match >

difference in means calculated above match estimate for coefficient β here.

In general, appropriate discrete NUMERIC values allow Regression to accomplish the same calculations as done in t-Tests and ANOVA! When, as above, there are only two levels within a factor, numerical coding of a "dummy" variable is straightforward. In "multi-state" factors where there are more than two levels, coding is more complex usually involves multiple columns of "dummy" variables. There are multiple ways to code a factor into one or more "dummy" variables. Software packages often make options available. They will also choose a default method that may, or may not, be transparent to the end user.

One-Way ANOVA

```
#COMPARING ONE-WAY ANOVA AS GLM:
ZAR3=read.table("c:/DATA/Biostatistics/ZarEX10.1R.txt")
ZAR3
attach(ZAR3)
Y=weights
X=factor(feed)
LM=lm(Y~X)
model.matrix(LM)
```

In this example, there are four levels within factor "feed". When assignment to X was made using:
`X=factor(feed)`
followed by:
`LM=lm(Y~X)`
we told R's GLM procedure to code the factor using dummy variables. The following command:
`model.matrix(LM)`
allows us to see the dummy variables directly, as well as the method of coding R used, called "cont.treatment". An ANOVA table and test can be generated as seen in *Biostatistics* Worksheet 230 where, in fact, GLM did the work for us.

model matrix with "dummy" variables:

```
   (Intercept) X2 X3 X4
1            1  0  0  0
2            1  0  0  0
3            1  0  0  0
4            1  0  0  0
5            1  0  0  0
6            1  1  0  0
7            1  1  0  0
8            1  1  0  0
9            1  1  0  0
10           1  1  0  0
11           1  0  1  0
12           1  0  1  0
13           1  0  1  0
14           1  0  1  0
15           1  0  0  1
16           1  0  0  1
17           1  0  0  1
18           1  0  0  1
19           1  0  0  1
attr(,"assign")
[1] 0 1 1 1
attr(,"contrasts")
attr(,"contrasts")$X
[1] "contr.treatment"
```

datafile:

	weights	feed
1	60.8	1
2	67.0	1
3	65.0	1
4	68.6	1
5	61.7	1
6	68.7	2
7	67.7	2
8	75.0	2
9	73.3	2
10	71.8	2
11	69.6	3
12	77.1	3
13	75.2	3
14	71.5	3
15	61.9	4
16	64.2	4
17	63.1	4
18	66.7	4
19	60.3	4

anova(LM)

```
Analysis of Variance Table

Response: Y
          Df Sum Sq Mean Sq F value   Pr(>F)
X          3 338.94 112.979  12.040 0.000283 ***
Residuals 15 140.75   9.383
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Reading Estimates for Multi-state Unordered Factors in R:

ANOVA models are General Linear Models in which discrete (typically unordered) variables defining multiple groups are entered into the Linear Model regression machinery as specially coded multi-level "factors", with two or more levels for each factor. Coding of factors (i.e., "dummy variables") may be accomplished in different ways. Most statistical programs including R have options for coding "contrasts" for dummy variables and, of course, a default. R's anova() function summarizing the effect of each factor as a whole is unaffected by differences in coding. However the summary() results, looking at levels within a factor, will differ depending upon which coding contrast is employed. Listed below are common options in R.

Treatment Contrasts (R default):

"cont.treatment"

#TREATMENT CONTRASTS (DEFAULT)
LM=lm(Y~X)
contrasts(X)
model.matrix(LM)
summary(LM)

contrasts:

	2	3	4
1	0	0	0
2	1	0	0
3	0	1	0
4	0	0	1

model matrix:

	(Intercept)	X2	X3	X4
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	1	0	0	0
5	1	0	0	0
6	1	1	0	0
7	1	1	0	0
8	1	1	0	0
9	1	1	0	0
10	1	1	0	0
11	1	0	1	0
12	1	0	1	0
13	1	0	1	0
14	1	0	1	0
15	1	0	0	1
16	1	0	0	1
17	1	0	0	1
18	1	0	0	1
19	1	0	0	1

```
attr(,"assign")
[1] 0 1 1 1
attr(,"contrasts")
attr(,"contrasts")$X
[1] "contr.treatment"
```

```
Call:
lm(formula = Y ~ X)

Residuals:
   Min     1Q Median     3Q    Max
 -3.82  -2.76   0.38   2.19   3.98

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   64.620      1.370  47.171  < 2e-16 ***
X2             6.680      1.937   3.448 0.003586 **
X3             8.730      2.055   4.248 0.000701 ***
X4            -1.380      1.937  -0.712 0.487204
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

^ Here, the Estimate of "Intercept" refers to the the mean of the first factor level (i.e., factor level "1") encountered in the dataset. Estimates of X2, X3 & X4 are *differences* between means of each of these levels from that of X1 (the intercept). The t-tests are marginal tests of H_0: no difference between mean of X2 versus X1, X3 versus X1, and X4 versus X1, for each row in the summary() output.

Note: This system of contrast coding is almost, *but not quite*, equivalent to Treatments Model in ANOVA. Whereas Treatments models in ANOVA employ the grand mean μ and treatment effects α_i, the "contr.treatments" contrast system in R uses the mean for the first group specified in the model formula (reading from left to right) μ_1 and differences between the first group and others ($\mu_1 - \mu_2$), ($\mu_1 - \mu_3$), ...

Cell Means Contrasts:

"contr.treatment"

contrasts:

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

model matrix:

```
   X1 X2 X3 X4
1   1  0  0  0
2   1  0  0  0
3   1  0  0  0
4   1  0  0  0
5   1  0  0  0
6   0  1  0  0
7   0  1  0  0
8   0  1  0  0
9   0  1  0  0
10  0  1  0  0
11  0  0  1  0
12  0  0  1  0
13  0  0  1  0
14  0  0  1  0
15  0  0  0  1
16  0  0  0  1
17  0  0  0  1
18  0  0  0  1
19  0  0  0  1
attr(,"assign")
[1] 1 1 1 1
attr(,"contrasts")
attr(,"contrasts")$X
[1] "contr.treatment"
```

```
#CELL MEANS CONTRASTS:
LM=lm(Y~X-1)#Note use of -1
contrasts(X)
model.matrix(LM)
summary(LM)
```

< Note -1 in formula

```
Call:
lm(formula = Y ~ X - 1)

Residuals:
   Min     1Q Median     3Q    Max
 -3.82  -2.76   0.38   2.19   3.98

Coefficients:
   Estimate Std. Error t value Pr(>|t|)
X1   64.620      1.370   47.17   <2e-16 ***
X2   71.300      1.370   52.05   <2e-16 ***
X3   73.350      1.532   47.89   <2e-16 ***
X4   63.240      1.370   46.16   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.063 on 15 degrees of freedom
Multiple R-squared: 0.9984,     Adjusted R-squared: 0.998
F-statistic:  2340 on 4 and 15 DF,  p-value: < 2.2e-16
```

^ Here Estimates refer to means for each treatment level X1-X4. The marginal t-tests now refer to H_0: X1 - 0 = 0, X2 - 0 = 0, etc.

Note: Estimates provided by summary() are equivalent to the Cell Means Model in ANOVA, and the t-tests compare each estimate separately with zero.