Statistics 512: Applied Linear Models

Topic 6

Topic Overview

This topic will cover

- One-way Analysis of Variance (ANOVA)

One-Way Analysis of Variance (ANOVA)

- Also called "single factor ANOVA".
- The response variable Y is continuous (same as in regression).
- There are two key differences regarding the explanatory variable X.
  1. It is a qualitative variable (e.g. gender, location, etc). Instead of calling it an *explanatory variable*, we now refer to it as a *factor*.
  2. No assumption (i.e. linear relationship) is made about the nature of the relationship between X and Y. Rather we attempt to determine whether the response differ significantly at different levels of X. This is a generalization of the *two-independent-sample t-test*.
- We will have several different ways of parameterizing the model:
  1. the cell means model
  2. the factor effects model
     - two different possible constraint systems for the factor effects model

Notation for One-Way ANOVA

X (or A) is the qualitative factor

- r (or a) is the number of *levels*
- we often refer to these as *groups* or *treatments*

Y is the continuous response variable

- $Y_{i,j}$ is the jth observation in the ith group.
- $i = 1, 2, \ldots, r$ levels of the factor X.
- $j = 1, 2, \ldots, n_i$ observations at factor level i.

KNNL Example (page 685)

- See the file `nknw677.sas` for the SAS code.
- Y is the number of cases of cereal sold (`CASES`)
- X is the design of the cereal package (`PKGDES`)
- There are 4 levels for X representing 4 different package designs: $i = 1$ to 4 levels
- Cereal is sold in 19 stores, one design per store. (There were originally 20 stores but one had a fire.)
- $j = 1, 2, \ldots, n_i$ stores using design i. Here $n_i = 5, 5, 4, 5$. We simply use n if all of the n_i are the same. The total number of observations is $n_T = \sum_{i=1}^{r} n_i = 19$.

```
data cereal;
   infile 'H:\System\Desktop\CH16TA01.DAT';
   input cases pkgdes store;
proc print data=cereal;
```

Obs	cases	pkgdes	store
1	11	1	1
2	17	1	2
3	16	1	3
4	14	1	4
5	15	1	5
6	12	2	1
7	10	2	2
8	15	2	3
9	19	2	4
10	11	2	5
11	23	3	1
12	20	3	2
13	18	3	3
14	17	3	4
15	27	4	1
16	33	4	2
17	22	4	3
18	26	4	4
19	28	4	5

Note that the "`store`" variable is just j; here it does not label a particular store, and we do not use it (only one design per store).

Model (Cell Means Model)

Model Assumptions

- Response variable is normally distributed

- Mean may depend on the level of the factor
- Variance is constant
- All observations are independent

Cell Means Model

$Y_{i,j} = \mu_i + \epsilon_{i,j}$

- μ_i is the theoretical mean of all observations at level i.
- $\epsilon_{i,j} \sim^{iid} N(0, \sigma^2)$ and hence $Y_{i,j} \sim^{iid} N(\mu_i, \sigma^2)$.
- Note there is no "intercept" term and we just have a potentially *different* mean for each level of X. In this model, the mean does not depend numerically on the actual value of X (unlike the linear regression model).

Parameters

- The parameters of the model are $\mu_1, \mu_2, \ldots, \mu_r, \sigma^2$.
- Basic analysis question is whether or not the explanatory variable helps to explain the mean of Y. In this case, this is the same as asking whether or not μ_i depends on i. So we will want to test $\mathrm{H}_0 : \mu_1 = \mu_2 = \ldots = \mu_r$ against the alternative hypothesis that the means are not all the same.
  We may further be interested in grouping the means into subgroups that are equivalent (statistically indistinguishable).

Estimates

- Estimate μ_i by the mean of the observations at level i. That is,

$$\hat{\mu}_i = \bar{Y}_{i.} = \frac{\sum_j Y_{i,j}}{n_i}$$

- For each level i, get an estimate of the variance,

$$s_i^2 = \frac{\sum_{j=1}^{n_i} (Y_{i,j} - \bar{Y}_{i.})^2}{n_i - 1}$$

- We combine these s_i^2 to get an estimate of σ^2 in the following way.

Pooled Estimate of σ^2

If the n_i are all the same we would simply average the s_i^2; otherwise use a weighted average. (Do *not* average the s_i.) In general we pool the s_i^2, using weights proportional to the degrees of freedom $n_i - 1$ for each group. So the pooled estimate is

$$\begin{aligned} s^2 &= \frac{\sum_{i=1}^r (n_i-1)s_i^2}{\sum_{i=1}^r (n_i-1)} = \frac{\sum_{i=1}^r (n_i-1)s_i^2}{n_T - r} = \frac{\sum_{i=1}^r \sum_{j=1}^{n_i} (Y_{i,j} - \bar{Y}_{i\cdot})^2}{n_T - r} \\ &= MSE. \end{aligned}$$

In the special case that there are an equal number of observations per group ($n_i = n$) then $n_T = nr$ and this becomes

$$s^2 = \frac{(n-1)\sum_{i=1}^r s_i^2}{nr - r} = \frac{1}{r}\sum_{i=1}^r s_i^2,$$

a simple average of the s_i^2.

Run `proc glm`

`glm` standards for "General Linear Model". The `class` statement tells `proc glm` that `pkgdes` is a "classification" variable, i.e. categorical. The `class` statement defines variables which are qualitative in nature. The `means` statement requests sample means and standard deviations for each factor level.

```
proc glm data=cereal;
   class pkgdes;
   model cases=pkgdes;
   means pkgdes;
```

```
The GLM Procedure
    Class Level Information
Class         Levels    Values
pkgdes             4    1 2 3 4
Number of observations    19
```

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	588.2210526	196.0736842	18.59	<.0001
Error	15	158.2000000	10.5466667		
Corrected Total	18	746.4210526			

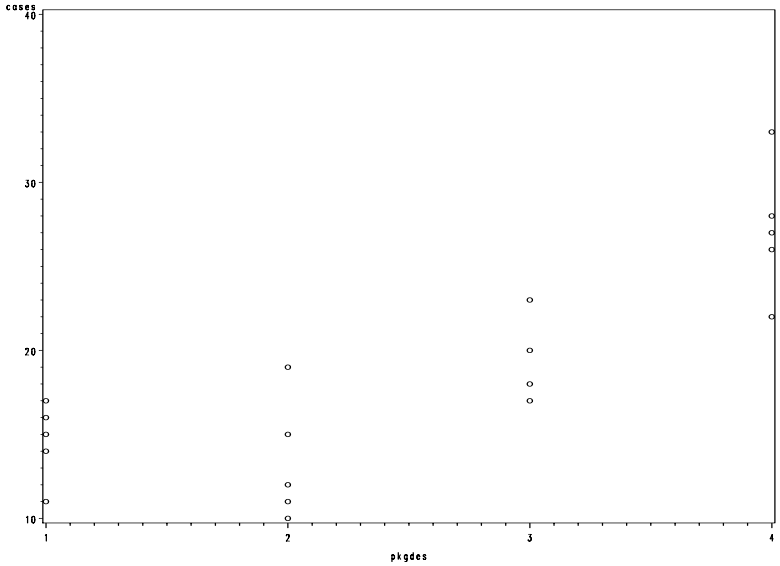
R-Square	Coeff Var	Root MSE	cases Mean
0.788055	17.43042	3.247563	18.63158

`means` statement output

Level of pkgdes	N	cases Mean	cases Std Dev
1	5	14.6000000	2.30217289
2	5	13.4000000	3.64691651
3	4	19.5000000	2.64575131
4	5	27.2000000	3.96232255

Plot the data.

```
symbol1 v=circle i=none;
proc gplot data=cereal;
   plot cases*pkgdes;
```

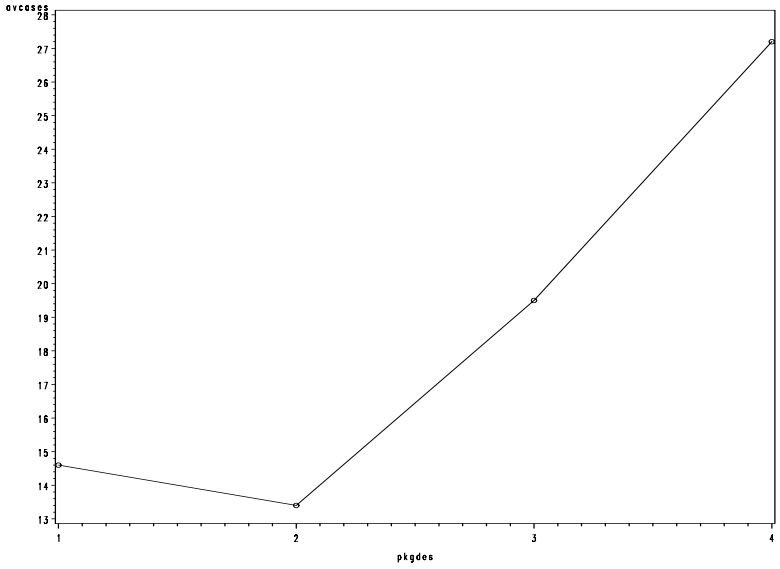


Look at the means and plot them.

```
proc means data=cereal;
   var cases; by pkgdes;
   output out=cerealmeans mean=avcases;
proc print data=cerealmeans;
```

Obs	pkgdes	_TYPE_	_FREQ_	avcases
1	1	0	5	14.6
2	2	0	5	13.4
3	3	0	4	19.5
4	4	0	5	27.2

```
symbol1 v=circle i=join;
proc gplot data=cerealmeans;
   plot avcases*pkgdes;
```



Some more notation

- The mean for group or treatment i is $\bar{Y}_{i.} = \frac{\sum_{j=1}^{n_i} Y_{i,j}}{n_i}$.
- The overall of "grand" mean is $\bar{Y}_{..} = \frac{\sum_{i=1}^{r}\sum_{j=1}^{n_i} Y_{i,j}}{n_T}$.
- The total number of observations is $n_T = \sum_{i=1}^{r} n_i$.

ANOVA Table

Source	df	SS	MS
Reg	$r-1$	$\sum_i n_i(Y_{i.} - Y_{..})^2$	$\frac{SSR}{df_R}$
Error	$n_T - r$	$\sum_{i,j}(Y_{i,j} - Y_{i.})^2$	$\frac{SSE}{df_E}$
Total	$n_T - 1$	$\sum_{i,j}(Y_{i,j} - \bar{Y}_{..})^2$	$\frac{SST}{df_T}$

Expected Mean Squares

$\mathrm{E}(MSR) = \sigma^2 + \frac{\sum_i n_i(\mu_i - \mu_.)^2}{r-1}$, where $\mu_. = \frac{\sum_i n_i \mu_i}{n_T}$.
$\mathrm{E}(MSE) = \sigma^2$.
$\mathrm{E}(MSR) > \mathrm{E}(MSE)$ when some group means are different. See KNNL pages 694 - 696 for more details. In more complicated models, these tell us how to construct the F-test.

F-test

$$\begin{aligned} \mathrm{H}_0 &: \mu_1 = \mu_2 = \ldots = \mu_r \\ \mathrm{H}_a &: \text{not all } \mu_i \text{ are equal} \end{aligned}$$

$$F = \frac{MSR}{MSE}$$

- Under H_0, $F \sim F_{(r-1,n_T-r)}$
- Reject H_0 when F is large.
- Report the p-value

Factor Effects Model

The factor effects model is just a re-parameterization of the cell means model. It is a useful way at looking at more complicated models; for now it may not seem worth the trouble but it will be handy later. Often the null hypotheses are easier to interpret with the factor effects model. The model is $Y_{i,j} = \mu + \tau_i + \epsilon_{i,j}$ where $\epsilon_{i,j} \sim^{iid} N(0, \sigma^2)$.

Parts of the Model

- μ is the overall or grand mean (it looks like an intercept). Note: The text calls this $\mu_.$, a notation I will not use in the notes.
- The τ_i represent the difference between the overall mean and the mean for level i. So whereas the cell means model looks at the mean for each level, this model looks at the amount by which the mean at each level deviates from some "standard".

Parameters

- The parameters of the factor effects model are $\mu, \tau_1, \tau_2, \ldots, \tau_r, \sigma^2$. There are $r + 2$ of these.
- Recall that the cell means model had $r + 1$ parameters: $\mu_1, \mu_2, \ldots, \mu_r, \sigma^2$, so in our new model one of the τ's is redundant. Thus we will need to place a restraint on the τ's to avoid estimating this "extra" parameter. (The models should be equivalent.)
- The relationship between the models is that $\mu_i = \mu + \tau_i$ for every i. If we consider the sum of these, we have $\sum \mu_i = r\mu + \sum \tau_i$. If the n_i are equal this is just $r\mu = r\mu + \sum \tau_i$ so the constraint we place on the model is $\sum \tau_i = 0$. Thus we need only estimate all of the τ's, except for one which may be obtained from the others.

Constraints – An Example

Suppose $r = 3$, $\mu_1 = 10$, $\mu_2 = 20$, $\mu_3 = 30$. Without the restrictions, we could come up with several equivalent sets of parameters for the factor effects model. Some include

$$\begin{aligned}
\mu &= 0, \tau_1 = 10, \tau_2 = 20, \tau_3 = 30 \text{ (same)} \\
\mu &= 20, \tau_1 = -10, \tau_2 = 0, \tau_3 = 10 \\
\mu &= 30, \tau_2 = -20, \tau_2 = -10, \tau_3 = 0 \\
\mu &= 5000, \tau_1 = -4990, \tau_2 = -4980, \tau_3 = -4970
\end{aligned}$$

In this situation, these parameters are called *not estimable* or not well defined. That is to say that there are many solutions to the least squares problem (not a unique choice) and in fact the $\mathbf{X}'\mathbf{X}$ matrix for this parameterization does not have an inverse. While there are many different restrictions that could be used (e.g. $\mu = 0$ would lead to the cell means model), the common restriction that $\sum_i \tau_i = 0$ sets things up so that μ is the grand average and the τ's represent the deviations from that average. This effectively reduces the number of parameters by 1. The details are a bit more complicated when the n_i are not all equal; in that case it is appropriate to weight the terms in the sum by their relative sample sizes. See KNNL pages 701-704 for details.

In summary, we always have $\mu_i = \mu + \tau_i$ as the relationship between the cell means model and the factor effects model. The constraint $\sum_i \tau_i = 0$ implies $\mu_\cdot = \frac{1}{r}\sum_i \mu_i$ (grand mean). (If weights $w_i = \frac{n_i}{n_T}$ are used the corresponding statements are $\sum_i w_i\tau_i$ and $\mu = \sum_i w_i\mu_i$.)

Hypothesis Tests

- The group or factor level effects are $\tau_i = \mu_i - \mu_\cdot$.
- The cell means model hypotheses were

$$\begin{aligned} \mathrm{H}_0 &: \ \mu_1 = \mu_2 = \ldots = \mu_r \\ \mathrm{H}_a &: \ \text{not all of the } \mu_i \text{ are equal} \end{aligned}$$

- For the factor effects model these translate to

$$\begin{aligned} \mathrm{H}_0 &: \ \tau_1 = \tau_2 = \ldots = \tau_r = 0 \\ \mathrm{H}_a &: \ \text{at least one of the } \tau_i \text{ is not } 0 \end{aligned}$$

Estimators of Parameters

With the zero-sum constraint $\sum_i \tau_i = 0$, the estimates are $\hat{\mu} = \bar{Y}_{\cdot\cdot}$ and $\hat{\tau}_i = \bar{Y}_{i\cdot} - \bar{Y}_{\cdot\cdot}$.

Solution used by SAS

Recall, $\mathbf{X}'\mathbf{X}$ may not have an inverse. We can use a *generalized inverse* in its place. $(\mathbf{X}'\mathbf{X})^-$ is the standard notation for a generalized inverse.

Definition: the *generalized inverse* of a matrix $\mathbf{A}$ is any matrix $\mathbf{A}^-$ satisfying $\mathbf{A}\mathbf{A}^-\mathbf{A} = \mathbf{A}$. The generalized inverse is not unique. There are many generalized inverses, each corresponding to a different constraint (underdetermined system). The matrix $\mathbf{A}$ does not have to be square.

The particular $(\mathbf{X}'\mathbf{X})^-$ used in `proc glm` corresponds to the constraint $\tau_r = 0$ (note this is different from our constraint). Recall that μ and the τ_i are not uniquely estimable separately. But the linear combinations $\mu + \tau_i$ are estimable. These are estimated by the cell means model.

KNNL Example page 685

- The code is in the file `nknw677.sas`.
- Y is the number of cases of cereal sold
- X is the design of the cereal package
- $i = 1$ to 4 levels
- $j = 1$ to n_i stores with design i

SAS Coding for X

SAS does this automatically/internally. You don't need to do the work to specify this in SAS.
The n_T rows of the design matrix are copies of the following four possible rows:

1 1 0 0 0 for level 1 (i.e. this is row i if $X_i = 1$)
1 0 1 0 0 for level 2
1 0 0 1 0 for level 3
1 0 0 0 1 for level 4

So our design matrix is

$$X = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The columns correspond to the parameter vector $\beta = \begin{bmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix}$.

You can see that the parameter μ acts a little like the intercept parameter in regression.

Some options in `proc glm`

```
proc glm data=cereal;
   class pkgdes;
   model cases=pkgdes/xpx inverse solution;
```

Result of `xpx` option: the `xpx` option actually gives the following matrix:

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Y} \\ \mathbf{Y}'\mathbf{X} & \mathbf{Y}'\mathbf{Y} \end{bmatrix}$$

The X'X Matrix

	Intercept	pkgdes 1	pkgdes 2	pkgdes 3	pkgdes 4	cases
Intercept	19	5	5	4	5	354
pkgdes 1	5	5	0	0	0	73
pkgdes 2	5	0	5	0	0	67
pkgdes 3	4	0	0	4	0	78
pkgdes 4	5	0	0	0	5	136
cases	354	73	67	78	136	7342

Result of `inverse` option: the `inverse` option actually gives the following matrix:

$$\begin{bmatrix} (\mathbf{X}'\mathbf{X})^- & (\mathbf{X}'\mathbf{X})^-\mathbf{X}'\mathbf{Y} \\ \mathbf{Y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^- & \mathbf{Y}'\mathbf{Y} - \mathbf{Y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^-\mathbf{X}'\mathbf{Y} \end{bmatrix}.$$

X'X Generalized Inverse (g2)

	Intercept	pkgdes 1	pkgdes 2	pkgdes 3	pkgdes 4
Intercept	0.2	-0.2	-0.2	-0.2	0
pkgdes 1	-0.2	0.4	0.2	0.2	0
pkgdes 2	-0.2	0.2	0.4	0.2	0
pkgdes 3	-0.2	0.2	0.2	0.45	0
pkgdes 4	0	0	0	0	0
cases	27.2	-12.6	-13.8	-7.7	0

Parameter estimates are in upper right corner; SSE is lower right corner.

Parameter estimates (from `solution` option)

Parameter		Estimate	Standard Error	t Value	Pr > |t|
Intercept		27.20000000 B	1.45235441	18.73	<.0001
pkgdes	1	-12.60000000 B	2.05393930	-6.13	<.0001
pkgdes	2	-13.80000000 B	2.05393930	-6.72	<.0001
pkgdes	3	-7.70000000 B	2.17853162	-3.53	0.0030
pkgdes	4	0.00000000 B	.	.	.

Note that these are just the same estimates as in the inverse matrix.

Caution Message

```
NOTE: The X'X matrix has been found to be singular, and a
      generalized inverse was used to solve the normal equations.
      Terms whose estimates are followed by the letter 'B' are not
      uniquely estimable.
```

Interpretation

If $\tau_r = 0$ (in our case, $\tau_4 = 0$), then the corresponding estimate should be zero. This means that the "intercept" μ in SAS is estimated by the mean of the observations in group 4. Since $\mu + \tau_i$ is the mean of group i, the τ_i are the differences between the mean of group i and the mean of group 4.

`means` Statement Output

Level of	cases		
pkgdes	N	Mean	Std Dev
1	5	14.6000000	2.30217289
2	5	13.4000000	3.64691651
3	4	19.5000000	2.64575131
4	5	27.2000000	3.96232255

Parameter Estimates from `means`

		$\hat{\mu} = 27.2$	$= 27.2$
1	14.6	$\hat{\tau}_1 = 14.6 - 27.2$	$= -12.6$
2	13.4	$\hat{\tau}_2 = 13.4 - 27.2$	$= -13.8$
3	19.5	$\hat{\tau}_3 = 19.5 - 27.2$	$= -7.7$
4	27.2	$\hat{\tau}_4 = 27.2 - 27.2$	$= 0$

The `means` output gives the estimates of the μ_i (cell means model). By subtracting off the mean for the last level from each of these means we get estimates for the factor effects (τ's) which match the results of the `solution` option.

Bottom line: you can use SAS to automatically get estimates for either the cell means model or the factor effects model with the last $\tau = 0$. You can also use appropriate subtractions to get estimates for any other constraint you choose. For example, if we want to use $\hat{\mu} = \frac{5\times14.6+5\times13.4+4\times19.5+5\times27.2}{19} = \frac{354}{19} = 18.63$ then subtract 18.63 from each of the μ_i estimates to get the τ_i estimates.

Summary: Single Factor Analysis of Variance

Cell Means Model

$Y_{i,j} = \mu_i + \epsilon_{i,j}$

- μ_i is the theoretical mean of all observations at level i
- $\epsilon_{i,j} \sim^{iid} N(0, \sigma^2)$ and hence $Y_{i,j} \sim^{iid} N(\mu_i, \sigma^2)$
- Note there is no "intercept" term and we just have a potentially *different* mean for each level of X. In this model, the mean does not depend numerically on the actual value of X (unlike the linear regression model).

With the cell means model there are no problems with parameter estimability and matrix inversion. Use the `means` statement in `proc glm` to get these estimates.

Factor Effects Model

$Y_{i,j} = \mu + \tau_i + \epsilon_{i,j}$ where $\epsilon_{i,j} \sim^{iid} N(0, \sigma^2)$

- This is a reparameterization of the cell means model and a useful way at looking at more complicated models.
- It is more useful since the null hypotheses are easier to state/interpret. But there are problems with singularity of $\mathbf{X'X}$.
- We utilize a constraint (e.g. $\sum \tau_i = 0$ or in SAS $\tau_r = 0$) to deal with these problems.

Section 16.8: Regression Approach to ANOVA

Essentially one-way ANOVA is linear regression with indicator (dummy) explanatory variables. We can use multiple regression to reproduce the results based on the factor effects model $Y_{i,j} = \mu + \tau_i + \epsilon_{i,j}$ where we will restrict $\sum_i \tau_i = 0$ by forcing $\tau_r = -\sum_{i=1}^{r-1} \tau_i$.

Coding for Explanatory Variables

We will define $r-1$ variables X_k, $k = 1, 2, \ldots, r-1$. Values of these variables will be denoted $X_{i,j,k}$, where the i, j subscript refers to the case $Y_{i,j}$ (i = factor level, j = # of observation at that level)

$$X_{i,j,k} = \begin{cases} 1 & \text{if } Y_{i,j} \text{ is from level } k \\ -1 & \text{if } Y_{i,j} \text{ is from level } r \\ 0 & \text{if } Y_{i,j} \text{ is from any other level} \end{cases}$$

Recall that our notation for $Y_{i,j}$ means that $Y_{i,j}$ is from level i. Thus the X variables are

$$X_{i,j,k} = \begin{cases} 1 & i = k \\ -1 & i = r \\ 0 & i \neq k, r \end{cases}$$

$i = 1, \ldots, r$, $j = 1, \ldots, n_i$, $k = 1, \ldots, r-1$
The k subscript refers to the column of the design matrix (not including the column of 1's), and the i, j subscripts indicate the rows (same order as the $Y_{i,j}$).

The regression model $Y_{i,j} = \beta_0 + \beta_1 X_{i,j,1} + \ldots + \beta_{r-1} X_{i,j,r-1} + \epsilon_{i,j}$ really becomes $\begin{array}{l} Y_{i,j} = \beta_0 + \beta_i + \epsilon_{i,j}, \\ Y_{r,j} = \beta_0 - \beta_1 - \ldots - \beta_{r-1} + \end{array}$
when the X's are plugged in as 0, 1, or -1. Comparing this to the factor effects model $Y_{i,j} = \mu + \tau_i + \epsilon_{i,j}$ we can make the following equivalencies:

$$\begin{aligned} \beta_0 &\equiv \mu; \\ \beta_i &\equiv \tau_i, i = 1, \ldots, r-1 \\ \tau_r &\equiv -(\beta_1 + \ldots + \beta_{r-1}) = -\sum_{i=1}^{r-1} \tau_i \text{ so that } \textstyle\sum_{i=1}^{r} \tau_i = 0. \end{aligned}$$

Thus, defining the indicator variables as we have done also specifies the constraint.

KNNL Example

- KNNL page 706 (`nknw698.sas`)
- This is the "cereal box" example that we have previously been working with. It is a bit messy because $n_i = 5, 5, 4, 5$. You have an easier example in the homework (n_i is constant).
- The grand mean is not the same as the mean of the group means in this case since the n's are different. Here we have $\mu = \frac{\sum_i n_i \mu_i}{n_T}$.

Look at the means

```
proc means data=cereal printalltypes;
   class pkgdes;
   var cases;
   output out=cerealmeans mean=mclass;
```

```
The MEANS Procedure
                        Analysis Variable : cases
  N
Obs     N            Mean         Std Dev         Minimum         Maximum
-------------------------------------------------------------------------
 19    19      18.6315789       6.4395525      10.0000000      33.0000000
-------------------------------------------------------------------------
```

```
                          Analysis Variable : cases
             N
pkgdes     Obs      N            Mean         Std Dev         Minimum         Maximum
-------------------------------------------------------------------------------------
     1       5      5      14.6000000       2.3021729      11.0000000      17.0000000
     2       5      5      13.4000000       3.6469165      10.0000000      19.0000000
     3       4      4      19.5000000       2.6457513      17.0000000      23.0000000
     4       5      5      27.2000000       3.9623226      22.0000000      33.0000000
-------------------------------------------------------------------------------------
```

```
proc print data=cerealmeans;
```

Obs	pkgdes	_TYPE_	_FREQ_	mclass
2	1	1	5	14.6000
3	2	1	5	13.4000
4	3	1	4	19.5000
5	4	1	5	27.2000

Note: `Type = 0` indicates the grand mean, `Type = 1` indicates the means are for the levels of a predictor variable. `Type = 2` would indicate that we had two predictor variables and for each a level was specified.

Explanatory Variables

Set X_1 to be 1 for design 1, -1 for design 4, and 0 otherwise; X_2 is 1 for design 2, -1 for design 4, and 0 otherwise; X3 is 1 for design 3, -1 for design 4, and 0 otherwise..

```
data cereal; set cereal;
   x1=(pkgdes eq 1)-(pkgdes eq 4);
   x2=(pkgdes eq 2)-(pkgdes eq 4);
   x3=(pkgdes eq 3)-(pkgdes eq 4);
proc print data=cereal; run;
```

New Variables

Obs	cases	pkgdes	store	x1	x2	x3
1	11	1	1	1	0	0
2	17	1	2	1	0	0
3	16	1	3	1	0	0
4	14	1	4	1	0	0
5	15	1	5	1	0	0
6	12	2	1	0	1	0
7	10	2	2	0	1	0
8	15	2	3	0	1	0
9	19	2	4	0	1	0
10	11	2	5	0	1	0
11	23	3	1	0	0	1
12	20	3	2	0	0	1
13	18	3	3	0	0	1
14	17	3	4	0	0	1
15	27	4	1	-1	-1	-1

```
16      33        4        2      -1    -1    -1
17      22        4        3      -1    -1    -1
18      26        4        4      -1    -1    -1
19      28        4        5      -1    -1    -1
```

Interpret X's in terms of parameters

Note the μ is implicit just like the intercept

pkgdes	Int	x1	x2	x3	
1	1	1	0	0	$\mu + \tau_1$
2	1	0	1	0	$\mu + \tau_2$
3	1	0	0	1	$\mu + \tau_3$
4	1	-1	-1	-1	$\mu - \tau_1 - \tau_2 - \tau_3$

Run the regression

```
proc reg data=cereal;
   model cases=x1 x2 x3;
```

```
The REG Procedure
Model: MODEL1
Dependent Variable: cases
                          Analysis of Variance
                                 Sum of           Mean
Source                   DF       Squares         Square    F Value    Pr > F
Model                     3     588.22105      196.07368      18.59    <.0001
Error                    15     158.20000       10.54667
Corrected Total          18     746.42105

Root MSE              3.24756    R-Square     0.7881
Dependent Mean       18.63158    Adj R-Sq     0.7457
Coeff Var            17.43042

                     Parameter Estimates
                   Parameter       Standard
Variable     DF     Estimate          Error    t Value    Pr > |t|
Intercept     1     18.67500        0.74853      24.95      <.0001
x1            1     -4.07500        1.27081      -3.21      0.0059
x2            1     -5.27500        1.27081      -4.15      0.0009
x3            1      0.82500        1.37063       0.60      0.5562
```

Compare with `proc glm`

```
proc glm data=cereal;
   class pkgdes;
   model cases=pkgdes;
```

```
The GLM Procedure
Dependent Variable: cases
                                    Sum of
Source                      DF         Squares     Mean Square    F Value    Pr > F
```

```
Model                          3     588.2210526     196.0736842      18.59    <.0001
Error                         15     158.2000000      10.5466667
Corrected Total               18     746.4210526

R-Square     Coeff Var      Root MSE    cases Mean
0.788055      17.43042      3.247563      18.63158

Source                        DF       Type I SS     Mean Square    F Value    Pr > F
pkgdes                         3     588.2210526     196.0736842      18.59    <.0001
```

Interpret the Regression Coefficients

Var	Est		
Int	18.675	$b_0 = \hat{\mu}$	(mean of the means)
x1	-4.075	$b_1 = \hat{\tau}_1 = \bar{Y}_{1\cdot} - \hat{\mu}$	(effect of level 1)
x2	-5.275	$b_2 = \hat{\tau}_2 = \bar{Y}_{2\cdot} - \hat{\mu}$	(effect of level 2)
x3	0.825	$b_3 = \hat{\tau}_3 = \bar{Y}_{3\cdot} - \hat{\mu}$	(effect of level 3)

$$b_0 + b_1 = 18.675 - 4.075 = 14.6 \text{ (mean for level 1)}$$
$$b_0 + b_2 = 18.675 - 5.275 = 13.4 \text{ (mean for level 2)}$$
$$b_0 + b_2 = 18.675 + 0.825 = 19.5 \text{ (mean for level 3)}$$
$$b_0 - b_1 - b_2 - b_3 = 18.675 + 4.075 + 5.275 - 0.825 = 27.2 \text{ (mean for level 4)}$$

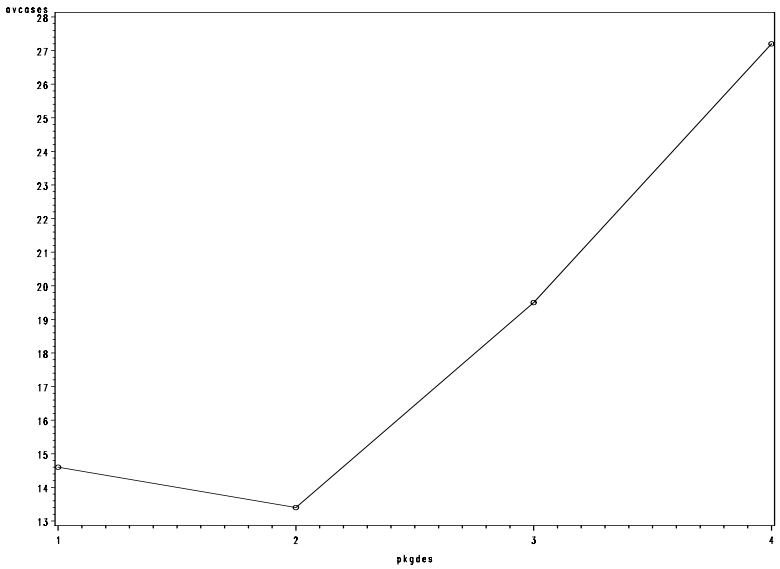
means statement output

```
Level of           ------------cases------------
pkgdes       N             Mean          Std Dev
1            5       14.6000000       2.30217289
2            5       13.4000000       3.64691651
3            4       19.5000000       2.64575131
4            5       27.2000000       3.96232255
```

Plot the means

```
proc means data=cereal;
   var cases; by pkgdes;
   output out=cerealmeans mean=avcases;
symbol1 v=circle i=join;
proc gplot data=cerealmeans;
   plot avcases*pkgdes;
```

The means



Confidence Intervals

- $\bar{Y}_{i\cdot} \sim N\left(\mu_i, \frac{\sigma^2}{n_i}\right)$ (since $Y_{i,j} \sim N(\mu_i, \sigma^2)$)
- CI for μ_i is $\bar{Y}_{i\cdot} \pm t_c \frac{s}{\sqrt{n_i}}$ (remember $s = \sqrt{MSE}$, $\frac{s}{\sqrt{n_i}}$ is often called the *standard error of the mean*)
- t_c is computed from the $t_{n_T - r}(1 - \frac{\alpha}{2})$ distribution.

CI's using `proc means`

You can get CI's from `proc means`, but it does not use the above formula. Instead `proc means` uses $\frac{s_i}{\sqrt{n_i}}$ for the CI at level i (CI for μ_i). It uses the within-group variation to estimate the standard error for each level, and *does not assume all levels have a common variance.* The *df* for t_c is $n_i - 1$ for level i. Thus the CI's using `proc means` will have different widths depending on their s_i's and n_i's.

```
proc means data=cereal
   mean std stderr clm
   maxdec=2;
   class pkgdes;
   var cases;
```

```
The MEANS Procedure
                    Analysis Variable : cases
           N                                    Lower 95%    Upper 95%
pkgdes   Obs    Mean  Std Dev  Std Error  CL for Mean  CL for Mean
-------------------------------------------------------------------
     1     5  14.60     2.30       1.03        11.74        17.46
     2     5  13.40     3.65       1.63         8.87        17.93
     3     4  19.50     2.65       1.32        15.29        23.71
```

```
     4     5  27.20     3.96       1.77        22.28        32.12
-----------------------------------------------------------------
```

CI's using `proc glm`

These use the pooled standard error formula (s not s_i) and the df is $n_T - r$ as given in the formula above. This is the way we will generally prefer since we have more degrees of freedom due to the constant variance assumption (and hence smaller MSE and SE's).

```
proc glm data=cereal;
   class pkgdes;
   model cases=pkgdes;
   means pkgdes/t clm;

The GLM Procedure
t Confidence Intervals for cases
Alpha                          0.05
Error Degrees of Freedom         15
Error Mean Square          10.54667
Critical Value of t         2.13145
                                        95% Confidence
pkgdes          N            Mean           Limits
4               5          27.200       24.104       30.296
3               4          19.500       16.039       22.961
1               5          14.600       11.504       17.696
2               5          13.400       10.304       16.496
```

These CI's are often narrower than the ones from `proc means` because more degrees of freedom (common variance). Notice that these CI's are all the same width except for design 3 ($n_i = 4$). They are sorted by descending mean. Here the `glm` CI is narrower for designs 2, 3, and 4 but slightly wider for design 1. (Design 1 had the smallest $s_i = 1.03$.)

Multiplicity Problem

- We have constructed 4 (in general, r) 95% confidence intervals. So the overall (family) confidence level (confidence that every interval contains its mean) is less than 95%.
- Many different kinds of adjustments have been proposed.
- We have previously discussed the Bonferroni correction (i.e., use α/r)

`bon` option in SAS

```
proc glm data=cereal;
   class pkgdes;
   model cases=pkgdes;
   means pkgdes/bon clm;
```

```
The GLM Procedure
Bonferroni t Confidence Intervals for cases
Alpha                          0.05
Error Degrees of Freedom         15
Error Mean Square           10.54667
Critical Value of t          2.83663
                                        Simultaneous 95%
pkgdes          N            Mean       Confidence Limits
4               5          27.200       23.080      31.320
3               4          19.500       14.894      24.106
1               5          14.600       10.480      18.720
2               5          13.400        9.280      17.520
```

Hypothesis Tests on Individual Means

Not common, but can be done.
Use `proc means` options `t` and `probt` for a test of the null hypothesis $H_0 : \mu_i = 0$
To test $H_0 : \mu_i = c$, where c is an arbitrary constant, first use a `data` step to subtract c from all observations and then run `proc means` options `t` and `probt`

```
proc means data=cereal mean std stderr clm maxdec=2;
   class pkgdes;
   var cases;
```

```
The MEANS Procedure
                    Analysis Variable : cases
          N
pkgdes  Obs        Mean    Std Dev  Std Error  t Value  Pr > |t|
-----------------------------------------------------------------
     1    5  14.6000000  2.3021729  1.0295630    14.18    0.0001
     2    5  13.4000000  3.6469165  1.6309506     8.22    0.0012
     3    4  19.5000000  2.6457513  1.3228757    14.74    0.0007
     4    5  27.2000000  3.9623226  1.7720045    15.35    0.0001
-----------------------------------------------------------------
```

Can also use GLM's `mean` statement with `clm` option and see whether it contains 0 (or the hypothesized value). This has the advantage of more df than the `proc means` way.

Differences in means

$$\bar{Y}_{i.} - \bar{Y}_{k.} \sim N\left(\mu_i - \mu_k, \frac{\sigma^2}{n_i} + \frac{\sigma^2}{n_k}\right)$$

We can test for equality of means by testing whether this difference is 0 (or looking to see whether 0 is in the CI).
CI for $\mu_i - \mu_k$ is $\bar{Y}_{i.} - \bar{Y}_{k.} \pm t^c s\{\bar{Y}_{i.} - \bar{Y}_{k.}\}$, where $s\{\bar{Y}_{i.} - \bar{Y}_{k.}\} = s\sqrt{\frac{1}{n_i} + \frac{1}{n_k}}$.

Multiple Comparisons: Determining the critical value

We deal with the multiplicity problem by adjusting t_c. Many different choices are available. These roughly fall into two categories:

- Change α level
- Use a different distribution

We will consider 4 slightly different testing procedures:

LSD

- Least Significant Difference (LSD) - this is the "least conservative" procedure we have.
- Simply ignores multiplicity issue and controls the test-alpha. If we have a lot of tests, it becomes very likely that we will make Type I errors (reject when we should not).
- Has better power than the rest of the tests.
- Uses $t_c = t_{n_T - r}(1 - \frac{\alpha}{2})$.
- Called `t` or `LSD` in SAS.
- This procedure is really too liberal and is not one that we often use.

Bonferroni

- More conservative than Tukey, but better if we only want to do comparisons for a *small number* of pairs of treatment means.
- Use the error budgeting idea to get family confidence level at least $1 - \alpha$.
- Sacrifices a little more power than Tukey.
- There are $\begin{pmatrix} r \\ 2 \end{pmatrix} = \frac{r(r-1)}{2}$ comparisons among r means, so replace α by $\frac{2\alpha}{r(r-1)}$ and use $t_c = t_{n_T - r}(1 - \frac{\alpha}{r(r-1)})$. For large r, Bonferroni is too conservative.
- Called `bon` in SAS.

Tukey

- More conservative than LSD.
- Specifies exact family alpha-level for comparing *all pairs* of treatment means, but has less power than LSD (wider CI's).
- Based on the *studentized range distribution* (maximum minus minimum divided by the standard deviation). See Table B.9.

- Uses $t_c = \frac{q_c}{\sqrt{2}}$.
- Details are in KNNL Section 17.5.
- Called `tukey` in SAS.

Scheffé

- Most conservative of the tests (sometimes).
- Controls family alpha level for testing ALL linear combinations of means (we'll talk about these later) but has less power (and so get CI's that are too wide). For testing pairs of treatment means it is (a bit) too conservative.
- Based on the F distribution
- $t_c = \sqrt{(r-1)F_{r-1,n_T-r}(1-\alpha)}$
- Protects against data snooping
- Called `scheffe` in SAS

Multiple Comparisons Summary

LSD is too liberal (get Type I errors / CI's too narrow).
Scheffe is conservative (no power for certain comparisons/ CI's wide).
Bonferroni is OK for small r (but conservative for large r).
Tukey (HSD) is recommended for general use.

Examples

```
proc glm data=a1;
   class pkgdes;
   model cases=pkgdes;
   means pkgdes/lsd tukey bon scheffe;
   means pkgdes/lines tukey;
```

LSD

```
The GLM Procedure
t Tests (LSD) for cases
NOTE: This test controls the Type I comparisonwise error rate, not the
experimentwise error rate.
Alpha                          0.05
Error Degrees of Freedom         15
Error Mean Square          10.54667
Critical Value of t         2.13145
Comparisons significant at the 0.05 level are indicated by ***.
                Difference
  pkgdes           Between      95% Confidence
```

```
Comparison          Means          Limits
4    - 3            7.700       3.057   12.343  ***
4    - 1           12.600       8.222   16.978  ***
4    - 2           13.800       9.422   18.178  ***
3    - 4           -7.700     -12.343   -3.057  ***
3    - 1            4.900       0.257    9.543  ***
3    - 2            6.100       1.457   10.743  ***
1    - 4          -12.600     -16.978   -8.222  ***
1    - 3           -4.900      -9.543   -0.257  ***
1    - 2            1.200      -3.178    5.578
2    - 4          -13.800     -18.178   -9.422  ***
2    - 3           -6.100     -10.743   -1.457  ***
2    - 1           -1.200      -5.578    3.178
```

Tukey

```
The GLM Procedure
Tukey's Studentized Range (HSD) Test for cases
NOTE: This test controls the Type I experimentwise error rate.
Alpha                                     0.05
Error Degrees of Freedom                    15
Error Mean Square                      10.54667
Critical Value of Studentized Range    4.07597
Comparisons significant at the 0.05 level are indicated by ***.
               Difference
  pkgdes          Between     Simultaneous 95%
Comparison          Means    Confidence Limits
4    - 3            7.700       1.421   13.979  ***
4    - 1           12.600       6.680   18.520  ***
4    - 2           13.800       7.880   19.720  ***
3    - 4           -7.700     -13.979   -1.421  ***
3    - 1            4.900      -1.379   11.179
3    - 2            6.100      -0.179   12.379
1    - 4          -12.600     -18.520   -6.680  ***
1    - 3           -4.900     -11.179    1.379
1    - 2            1.200      -4.720    7.120
2    - 4          -13.800     -19.720   -7.880  ***
2    - 3           -6.100     -12.379    0.179
2    - 1           -1.200      -7.120    4.720
```

Note $\frac{4.07}{\sqrt{2}} = 2.88$ is the t^c value used to make the CI.
Output (`lines` option)

```
           Mean      N    pkgdes
Tukey  Grouping
A        27.200      5    4

B        19.500      4    3
B
B        14.600      5    1
B
B        13.400      5    2
```

Run `nknw711.sas` for yourself to see and compare the intervals using the `lsd`, `bon` and `scheffe` options. They are wider than the `tukey` intervals. However, all three corrected methods (`bon`, `tukey`, `scheffe`) ultimately give the same conclusion for this example, namely, that design 4 has a significantly higher mean than the other three, but designs 1, 2, and 3 are not significantly different from one another.

Some other options in `proc glm`

- `alpha=0.`xx either in the procedure statement or after '`/`' in the `model` or `means` statement will change your alpha-level for the respective statement(s).
- `/DUNNETT('Control')` will perform tests that compare treatments to a control (where the 'control' in parentheses is the name of the level which is the control). This has more power than Tukey with the same family alpha in the case where you are making only those comparisons.
- `/LINES` will cause the tests to take a more convenient output (see last example above).

Linear Combinations of Means

- Often we wish to examine (test hypotheses about, make CI's for) particular linear combinations of the group means.
- These combinations should come from research questions, not from an examination of the data.
- A linear combination of means is any quantity of the form $L = \sum_i c_i \mu_i$ for any constants c_i. We estimate L with $\hat{L} = \sum_i c_i \bar{Y}_{i\cdot} \sim N(L, \mathrm{Var}(\hat{L}))$.
- Its variance is $\mathrm{Var}(\hat{L}) = \sum_i c_i^2 \mathrm{Var}(\bar{Y}_i)$,, which can be estimated by $s^2 \sum_i \frac{c_i^2}{n_i}$.

Contrasts

- A contrast is a special case of a linear combination with $\sum_i c_i = 0$. These turn out to be particularly useful because the interesting hypothesis tests are of the form $\mathrm{H}_0 : L = 0$.
- Example 1: $\mu_1 - \mu_2 \qquad (c_1 = 1,\, c_2 = -1)$
- Used to test whether levels 1 and 2 have equal means.
- Example 2: $\mu_1 - \frac{1}{2}(\mu_2 + \mu_3) \qquad (1, -0.5, -0.5)$
- Used to test whether level 1 has the same mean as the combination of levels 2/3.
- Example 3: $(\mu_1 + \mu_2)/2 - (\mu_3 + \mu_4)/2 \qquad (0.5,\, 0.5,\, -0.5,\, -0.5)$
- Used to test whether the first two levels have the same mean as the last two (think $1, 2$ = men; $3, 4$ = women and $1, 3$ = diet A; $2, 4$ = diet B - this would then test for gender differences)

`contrast` and `estimate` options in SAS

For Example 3 above

```
proc glm data=a1;
   class pkgdes;
   model cases=pkgdes;
   contrast '1&2 v 3&4' pkgdes .5 .5 -.5 -.5;
   estimate '1&2 v 3&4' pkgdes .5 .5 -.5 -.5;

Contrast   DF  Contrast SS  Mean Square  F Value  Pr > F
1&2 v 3&4   1  411.4000000  411.4000000    39.01  <.0001

                         Standard
Parameter     Estimate       Error  t Value  Pr > |t|
1&2 v 3&4  -9.35000000  1.49705266    -6.25    <.0001
```

The `contrast` statement performs the F-test. The `estimate` statement performs a t-test and gives the parameter estimate.

Multiple Contrasts

We can simultaneously test a collection of contrasts (1 df each contrast)
Example 1, $H_0 : \mu_1 = (\mu_2 + \mu_3 + \mu_4)/3$
The F statistic for this test will have an F_{1,n_T-r} distribution
Example 2, $H_0 : \mu_2 = \mu_3 = \mu_4$.
The F statistic for this test will have an F_{2,n_T-r} distribution
We do this by setting up one contrast for each comparison and doing them simultaneously.

```
proc glm data=a1;
   class pkgdes;
   model cases=pkgdes;
   contrast '1 v 2&3&4' pkgdes 1 -.3333 -.3333 -.3333;
   estimate '1 v 2&3&4' pkgdes 3 -1 -1 -1 /divisor=3;
   contrast '2 v 3 v 4' pkgdes 0 1 -1  0, pkgdes 0 0  1 -1;

Contrast   DF     Contrast SS     Mean Square    F Value    Pr > F
1 v 2&3&4   1     108.4739502     108.4739502      10.29    0.0059
2 v 3 v 4   2     477.9285714     238.9642857      22.66    <.0001

                            Standard
Parameter     Estimate           Error    t Value    Pr > |t|
1 v 2&3&4  -5.43333333      1.69441348      -3.21      0.0059
```

Chapter 18 – Diagnostics: Overview

We will take the diagnostics and remedial measures that we learned for regression and adapt them to the ANOVA setting. Many things are essentially the same, while some things require modification.

Residuals

- Predicted values are cell means: $\hat{Y}_{i,j} = \bar{Y}_{i.}$.
- Residuals are the differences between the observed values and the cell means $e_{i,j} = Y_{i,j} - \bar{Y}_{i.}$.

Basic plots

- Plot the data vs the factor levels (the values of the explanatory variables)
- Plot the residuals vs the factor levels
- Construct a normal quantile plot of the residuals

Notice that we are no longer checking for linearity since this is not an assumption in ANOVA.

KNNL Example

- KNNL page 734 (`nknw712.sas`)
- Compare 4 brands of rust inhibitor (X has $r = 4$ levels)
- Response variable is a measure of the effectiveness of the inhibitor
- There are 40 observations, 10 units per brand ($n = 10$ constant across levels)

```
data rust;
    infile 'H:\System\Desktop\CH17TA02.DAT';
    input eff brand;
```

Recode the factor: just to show they can be letters instead of numbers if you want.

```
data rust; set rust;
   if brand eq 1 then abrand='A';
   if brand eq 2 then abrand='B';
   if brand eq 3 then abrand='C';
   if brand eq 4 then abrand='D';
proc print data=rust;
```

Store the residuals in dataset `rustout`.

```
proc glm data=rust;
   class abrand;
   model eff = abrand;
   output out=rustout r=resid;
```

Residuals have the same syntax as in `proc reg`.

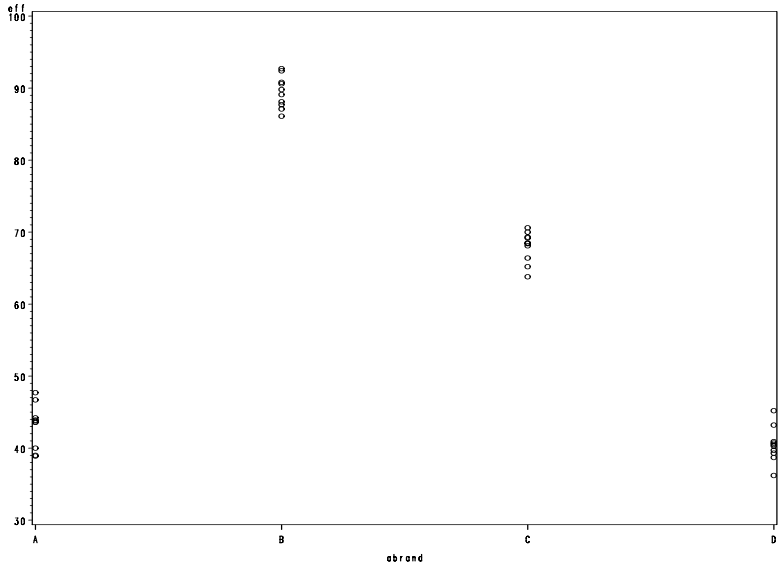
Plots

- Data versus the factor
- Residuals versus the factor (or predictor)
- Normal quantile plot of the residuals

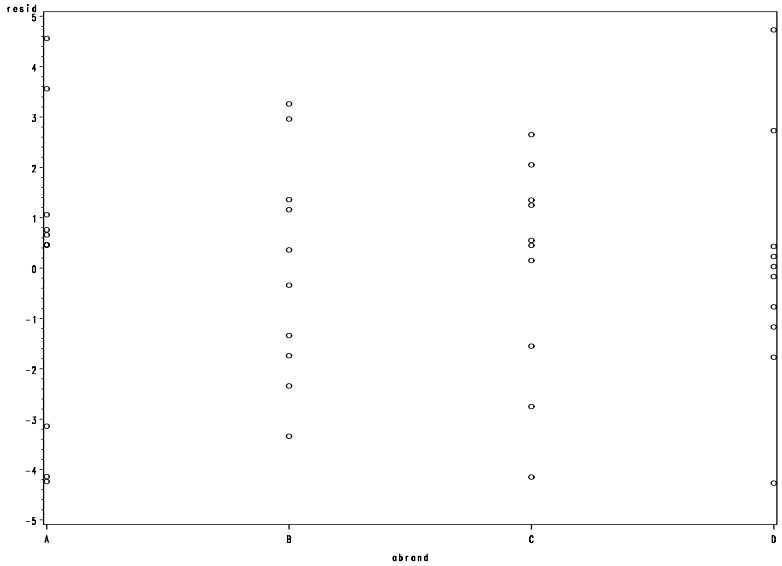
Plot vs the factor

```
symbol1 v=circle i=none;
proc gplot data=rustout;
plot (eff resid)*abrand;
```

Data vs the factor

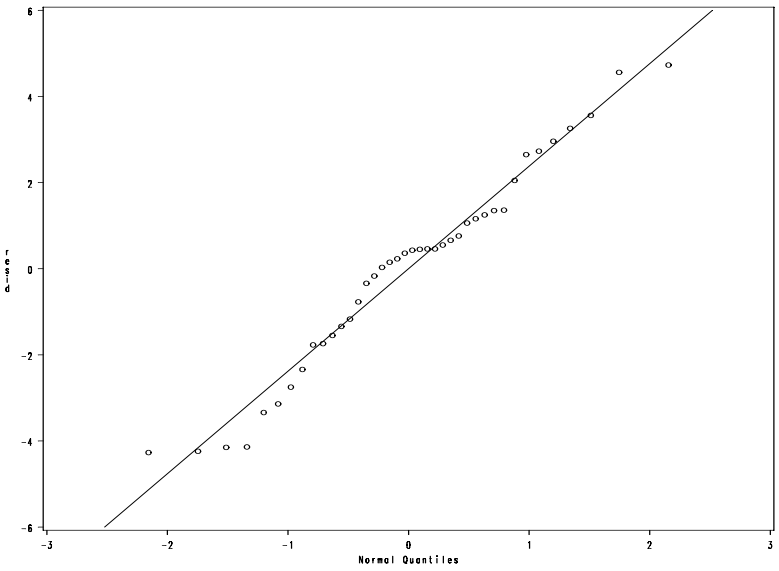


Residuals vs the factor



Normal quantile plot of the residuals

```
proc univariate data = rustout;
  qqplot resid / normal (L=1 mu=est sigma=est);
```



Summary of plot diagnostics

Look for

- Outliers
- Variance that depends on level
- Non-normal errors

Plot residuals vs time and other variables

Homogeneity tests

Homogeneity of variance (homoscedasticity) is assumed in the model. We can test for that.

$$\begin{aligned} \mathrm{H}_0: &\quad \sigma_1^2 = \sigma_2^2 = \ldots = \sigma_r^2 \quad \text{(constant variance)} \\ \mathrm{H}_A: &\quad \text{not all } \sigma_i^2 \text{ are equal} \quad \text{(non-constant variance)} \end{aligned}$$

- Several significance tests are available. Note that this was also available in regression if each X has multiple Y observations (this is usually true in ANOVA): see section 3.6.
- Text discusses Hartley, modified Levene.
- SAS has several including Bartlett's (essentially the likelihood ratio test) and several versions of Levene.

ANOVA is robust with respect to moderate deviations from normality, but ANOVA results can be sensitive to the homogeneity of variance assumption. In other words we usually worry more about constant variance than we do about normality. However, there is a complication: some homogeneity tests are sensitive to the normality assumption. If the normality assumption is not met, we may not be able to depend on the homogeneity tests.

Levene's Test

- Do ANOVA on the squared residuals.
- Modified Levene's test uses absolute values of the residuals. Modified Levene's test is recommended because it is less sensitive to the normality assumption.

KNNL Example

KNNL page 783 `nknw768.sas`
Compare the strengths of 5 types of solder flux (X has $r = 5$ levels)
Response variable is the pull strength, force in pounds required to break the joint
There are 8 solder joints per flux ($n = 8$)

```
data solder;
   infile 'H:\System\Desktop\CH18TA02.DAT';
   input  strength type;
```

Modified Levene's Test

```
proc glm data=wsolder;
   class type;
   model strength=type;
   means type/hovtest=levene(type=abs);
```

```
Dependent Variable: strength
                                      Sum of
Source                       DF         Squares     Mean Square    F Value    Pr > F
Model                         4     353.6120850      88.4030213      41.93    <.0001
Error                        35      73.7988250       2.1085379
Corrected Total              39     427.4109100

R-Square     Coeff Var      Root MSE    strength Mean
0.827335      10.22124      1.452081         14.20650

     Levene's Test for Homogeneity of strength Variance
        ANOVA of Absolute Deviations from Group Means

                      Sum of        Mean
Source        DF     Squares      Square    F Value    Pr > F
type           4      8.6920      2.1730       3.07    0.0288
Error         35     24.7912      0.7083
```

Rejecting H_0 means there is evidence that variances are not homogeneous.

Means and SD's

```
The GLM Procedure
Level of           ----------strength----------
type         N             Mean          Std Dev
1            8       15.4200000       1.23713956
2            8       18.5275000       1.25297076
3            8       15.0037500       2.48664397
4            8        9.7412500       0.81660337
5            8       12.3400000       0.76941536
```

The standard deviations do appear quite different.

Remedies

- Delete outliers – Is their removal important?
- Use weights (weighted regression)
- Transformations
- Nonparametric procedures

Weighted Least Squares

- We used this with regression.
- Obtain model for how the sd depends on the explanatory variable (plotted absolute value of residual vs x)
- Then used weights inversely proportional to the estimated variance
- Here we can compute the variance for each level because we have multiple observations (replicates).
- Use these as weights in `proc glm`
- We will illustrate with the soldering example from KNNL.

Obtain the variances and weights

```
proc means data=solder;
   var strength;
   by type;
   output out=weights var=s2;
data weights;
   set weights;
   wt=1/s2;
```

NOTE Data set `weights` has 5 "observations", one for each level.
Merge and then use the weights in `proc glm`

```
data wsolder;
   merge solder weights;
   by type;
proc glm data=wsolder;
   class type;
   model strength=type;
   weight wt;
```

```
The GLM Procedure
Dependent Variable: strength
Weight: wt
                          Sum of
Source           DF      Squares  Mean Square  F Value  Pr > F
Model             4  324.2130988   81.0532747    81.05  <.0001
Error            35   35.0000000    1.0000000
Corrected Total  39  359.2130988

R-Square     Coeff Var      Root MSE    strength Mean
0.902565      7.766410       1.00000          12.87596
```

Note the increase in the size of the F-statistic as well as R^2. Also notice that the MSE is now 1.

Transformation Guides

Transformations can also be used to solve constant variance problems, as well as normality.

- When σ_i^2 is proportional to μ_i, use $\sqrt{Y}$.
- When σ_i is proportional to μ_i, use $\log(Y)$.
- When σ_i is proportional to μ_i^2, use $1/Y$.
- When Y is a proportion, use $2\arcsin(\sqrt{Y})$; this is `2*arsin(sqrt(y))` in a SAS `data` step.
- Can also use Box-Cox procedure.

Nonparametric approach

- Based on ranks
- See KNNL section 18.7, page 795
- See the SAS procedure `npar1way`

Section 17.9: Quantitative Factors

- Suppose the factor X is a quantitative variable (has a numeric order to its values).
- We can still do ANOVA, but regression is a possible alternative analytical approach.
- Here, we will compare models (e.g., is linear model appropriate or do we need quadratic, etc.)
- We can look at extra SS and general linear tests.
- We use the factor first as a continuous explanatory variable (regression) then as a categorical explanatory variable (ANOVA)
- We do all of this in one run with `proc glm`
- This is the same material that we skipped when we studied regression: F Test for *Lack of Fit*, KNNL Section 3.7, p 119.

KNNL Example

- KNNL page 762 (`nknw742.sas`)
- Y is the number of acceptable units produced from raw material
- X is the number of hours of training
- There are 4 levels for X: 6 hrs, 8 hrs, 10 hrs and 12 hrs.
- $i = 1$ to 4 levels ($r = 4$)
- $j = 1$ to 7 employees at each training level ($n = 7$)

```
data training;
 infile 'H:\System\Desktop\CH17TA06.DAT';
 input product trainhrs;
```

Replace `trainhrs` by actual hours; also quadratic.

```
data training; set training;
   hrs=2*trainhrs+4;
   hrs2=hrs*hrs;
```

```
Obs    product    trainhrs    hrs    hrs2
  1       40          1         6      36
...
  8       53          2         8      64
...
 15       53          3        10     100
...
 22       63          4        12     144
...
```

`PROC GLM` with both categorical ("`class`") and quantitative factors: if a variable is not listed on the `class` statement, it is assumed to be quantitative, i.e. a regression variable.

```
proc glm data=training;
   class trainhrs;
   model product=hrs trainhrs / solution;
```

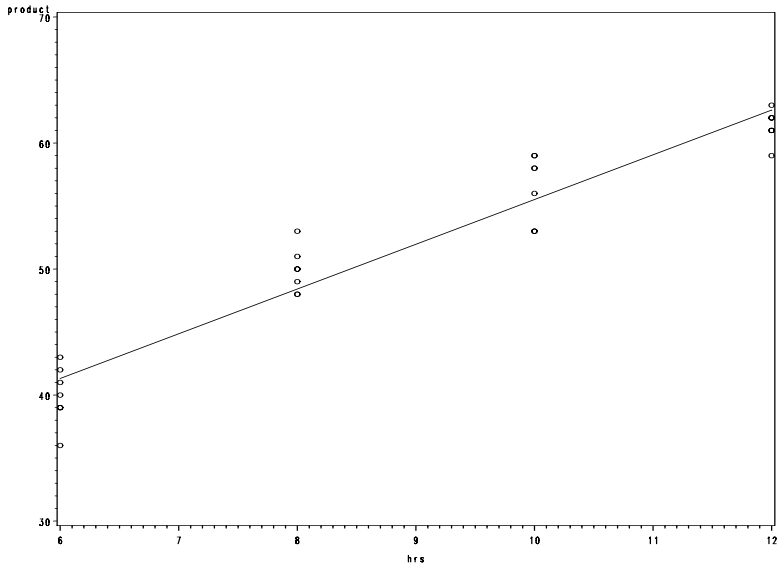
Note the multicollinearity in this problem: $hrs = 12 - 6X_1 - 4X_2 - 2X_3 - 0X_4$. Therefore, we will only get 3 (not 4) model df.

```
The GLM Procedure
Dependent Variable: product
                                      Sum of
Source                        DF         Squares     Mean Square    F Value    Pr > F
Model                          3     1808.678571      602.892857     141.46    <.0001
Error                         24      102.285714        4.261905
Corrected Total               27     1910.964286

R-Square     Coeff Var      Root MSE    product Mean
0.946474      3.972802      2.064438        51.96429

Source                        DF       Type I SS     Mean Square    F Value    Pr > F
hrs                            1     1764.350000     1764.350000     413.98    <.0001
trainhrs                       2       44.328571       22.164286       5.20    0.0133
```

The Type I test for `trainshrs` looks at the lack of fit. It asks, with `hrs` in the model (as a regression variable), does `trainhrs` have anything to add (as an ANOVA) variable? The null hypothesis for that test is that a straight line model with `hrs` is sufficient. Although `hrs` and `trainhrs` contain the same information, `hrs` is forced to fit a straight line, while `trainhrs` can fit any way it wants. Here it appears there is a significant deviation of the means from the fitted line because `trainhrs` is significant; the model fits better when non-linearity is permitted.

Interpretation

The analysis indicates that there is statistically significant lack of fit for the linear regression model ($F = 5.20$; $df = 2, 24$; $p = 0.0133$)



Looking at the plot suggests there is some curvature to the relationship. Let's try a quadratic term in the model.

Quadratic Model

```
proc glm data=training;
   class trainhrs;
   model product=hrs hrs2 trainhrs;
```

```
The GLM Procedure
Dependent Variable: product
                                    Sum of
Source                      DF         Squares     Mean Square    F Value    Pr > F
Model                        3     1808.678571      602.892857     141.46    <.0001
Error                       24      102.285714        4.261905
Corrected Total             27     1910.964286

R-Square     Coeff Var      Root MSE    product Mean
0.946474      3.972802      2.064438        51.96429

Source                      DF       Type I SS     Mean Square    F Value    Pr > F
hrs                          1     1764.350000     1764.350000     413.98    <.0001
hrs2                         1       43.750000       43.750000      10.27    0.0038
trainhrs                     1        0.578571        0.578571       0.14    0.7158
```

When we include a quadratic term for `hrs`, the remaining `trainhrs` is not significant. This indicates that the quadratic model is sufficient and allowing the means to vary in an arbitrary way is not additionally helpful (does not fit any better). Note that the lack of fit test now only has 1 *df* since the model *df* has not changed.