Multiple Linear Regression Analysis across FMRI 3d Datasets

B. Douglas Ward
Biophysics Research Institute
Medical College of Wisconsin
email: ward@post.its.mcw.edu

January 13, 1998

Abstract

Program 3dRegAna was developed to provide multiple linear regression analysis across *AFNI* 3d datasets. Applications of program 3dRegAna include: simple linear regression, polynomial regression, multiple linear regression, regression analysis involving combinations of quantitative and qualitative predictor variables, and analysis of variance (ANOVA) for cases with unequal sample sizes (provided that the number of factor levels is not too large).

For each input dataset, the user must enter the quantitative level that applies for each of the independent (predictor) variables. The user also specifies which variables are to appear in the linear regression model.

If repeat observations are available, program 3dRegAna first performs a test for "lack of fit", to determine if the model is adequate for explaining variation in the data, for each voxel in the dataset. For those voxels where lack of fit is *not* indicated, program 3dRegAna then calculates the least squares fit of the regression parameters for the specified model, the F-statistic for significance of the overall regression, the coefficient of multiple determination R^2, and the t-statistics for significance of the individual parameters.

Program 3dRegAna output includes separate *AFNI* 3d datasets containing the individual parameter estimates, along with the corresponding F_{reg} statistic, R^2, and t-statistics, as requested by the user. The resulting output may be stored either as multiple *AFNI* 2 sub-brick datasets, or as a single *AFNI* "bucket" type dataset.

1 Program 3dRegAna

1.1 Purpose

Program 3dRegAna was developed to provide multiple linear regression analysis across *AFNI* 3d datasets. For each input dataset, the user must enter the quantitative level that applies for each of the independent (predictor) variables. The user also specifies which variables are to appear in the *full linear regression model*, as well as a simpler *reduced model*.

If repeat observations are available, program 3dRegAna first performs a "lack of fit" test for each voxel in the dataset. The F_{lof} statistic is used to determine if the full linear regression model is adequate for explaining variation in the data. Although the lack of fit test is not mandatory, it is strongly recommended for cases where repeat observations are available. Due to the large number of voxels in a typical FMRI dataset, visual inspection of each of the individual linear regression functions is not practical; therefore, automatic screening for adequacy of the regression model is important.

Program 3dRegAna continues with the regression analysis for those voxels where lack of fit is *not* indicated. The least squares fit of the regression parameters for the full model is calculated individually for each voxel. The regression statistic F_{reg} is calculated; this statistic indicates the significance of the full model relative to the reduced model. Therefore, the F_{reg} statistic can be used in tests of hypotheses about the model structure. The coefficient of multiple determination, R^2, represents the proportion of variation in the data that is explained by the full model. As such, it indicates how well the full model explains the data, and can be used in comparing alternative models. The t-statistics, which indicate the statistical significance of individual parameters within the full model, are also calculated for each voxel. Program 3dRegAna output includes separate *AFNI* 3d datasets containing the individual parameter estimates, along with the corresponding F_{reg} statistic, R^2, and t-statistics. Also, the user has the option of storing the output as a single *AFNI* "bucket" type dataset.

Applications of program 3dRegAna include: simple linear regression, polynomial regression, multiple linear regression, regression analysis involving combinations of quantitative and qualitative predictor variables, and analysis of variance (ANOVA) for cases with unequal sample sizes (provided that the number of factor levels is not too large).

Section 1.2 discusses the theory underlying multiple linear regression analysis. This section is a very brief summary of material that may be found in references such as ([1],[2]). Sections 1.3, 1.4, and 1.5 explain the batch commands necessary to run program 3dRegAna, and what the various command line options do. Section 1.6 contains examples illustrating the use of program 3dRegAna.

1.2 Theory

1.2.1 Multiple Linear Regression Models

A linear regression model is a linear function of its *parameters*. For example,

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \quad i = 1, \ldots, n,$$

is a linear function of the (unknown) parameters β_0 and β_1 (here, ε_i is additive noise). The index i denotes the observation number. Note that

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \varepsilon_i, \quad i = 1, \ldots, n,$$

although a nonlinear function of the independent variable X, is, in fact, a linear regression model, since Y is a linear function of β_0, β_1, and β_2.

The general linear regression model, written as a function of p independent variables X_1, X_2, …, X_{p-1} (and the constant $X_0 \equiv 1$), is given by:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_{p-1} X_{i,p-1} + \varepsilon_i.$$

where

Y_i are the measurements (intensities for a given voxel)

β_0, β_1, …, β_{p-1} are the unknown parameters (to be estimated for each voxel);

X_{i1}, X_{i2}, …, $X_{i,p-1}$ are the known predictor variables (constants for a given FMRI image);

ε_i are random errors, i.i.d. $N(0, \sigma^2)$;

$i = 1, \ldots, n$.

Using the following matrix notation:

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & X_{11} & \cdots & X_{1,p-1} \\ 1 & X_{21} & \cdots & X_{2,p-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_{n1} & \cdots & X_{n,p-1} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix},$$

the above general linear regression model can be written

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

The linear regression problem is then to find an estimate $\mathbf{b}$ of the vector of unknown parameters

$$\mathbf{b} = \hat{\boldsymbol{\beta}},$$

which provides a good “fit” to the data. The observed voxel intensity data is then estimated by:

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{b}$$

The usual criterion for estimating $\mathbf{b}$ is to minimize the error sum of squares:

$$\begin{aligned} SSE &= Q(\mathbf{b}) = \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2 \\ &= \left(\mathbf{Y} - \hat{\mathbf{Y}}\right)^t \left(\mathbf{Y} - \hat{\mathbf{Y}}\right). \end{aligned}$$

It is easy to show that

$$\mathbf{b} = \left(\mathbf{X}^t\mathbf{X}\right)^{-1} \mathbf{X}^t\mathbf{Y}$$

gives the least squares estimate of $\boldsymbol{\beta}$. Then, SSE can be written:

$$SSE = \mathbf{Y}^t \left[\mathbf{I} - \mathbf{X}\left(\mathbf{X}^t\mathbf{X}\right)^{-1}\mathbf{X}^t\right]\mathbf{Y}$$

1.2.2 F-test for Lack of Fit

The first question that must be considered is whether the specified model adequately explains the data. This is done by a test of the alternative hypotheses:

$$\begin{array}{ll} H_o: & E(Y) = \beta_0 + \beta_1 X_1 + \cdots + \beta_{p-1} X_{p-1} \\ H_a: & E(Y) \neq \beta_0 + \beta_1 X_1 + \cdots + \beta_{p-1} X_{p-1} \end{array}$$

In order to conduct a statistical test for whether the specified model is adequate for explaining variation in the observed results, it is necessary that there be repeat observations. Let c be the number of distinct rows in the $\mathbf{X}$ matrix; i.e., there are c distinct levels at which the response was measured, where $c < n$. Then there are $n - c$ repeat observations at one or more of the c levels. These repeat observations allow us to estimate the "pure error" arising from measurement errors or random variation in the underlying process, independent of error in the model itself. Let $\bar{Y}_j$ be the mean of the observations at the jth level, $1 \leq j \leq c$. Then the sum of squares due to "pure error" is given by:

$$SSPE = \sum_{j=1}^{c} \sum_{i=1}^{n_j} \left(Y_{ij} - \bar{Y}_j \right)^2$$

where n_j is the number of observations at the jth level. Note that $SSPE$ has $n - c$ degrees of freedom. For each of the c distinct levels, the hypothesized model provides the estimated response $\hat{Y}_j$. The difference between the mean response and the estimated response at each level is due to inadequacy of the model itself. Therefore, the sum of squares due to "lack of fit" is given by:

$$SSLF = \sum_{j=1}^{c} n_j \left(\bar{Y}_j - \hat{Y}_j \right)^2$$

Therefore, the error sum of squares SSE can be decomposed as a sum of the "pure error" sum of squares, and the sum of squares due to "lack of fit":

$$SSE = SSPE + SSLF.$$

Since SSE has $n - p$ degrees of freedom, it follows that $SSLF$ has $(n - p) - (n - c) = c - p$ degrees of freedom. To test for adequacy of the model, one then calculates the statistic F^*_{lof} :

$$\begin{aligned} F^*_{lof} &= \frac{MSLF}{MSPE} \\ &= \frac{SSLF/(c-p)}{SSPE/(n-c)} \end{aligned}$$

The test statistic F^*_{lof} has an $F(c - p, n - c)$ distribution under the null hypothesis ([1],[2]). Therefore, the decision rule for the above test of hypotheses is:

$$\begin{array}{lll} \text{if } F^*_{lof} & \leq & F(1 - \alpha, c - p, n - c), \text{ conclude } H_o \\ \text{if } F^*_{lof} & > & F(1 - \alpha, c - p, n - c), \text{ conclude } H_a. \end{array}$$

The above can be summarized in the following ANOVA table:

Source	SS	df	MS	F
Regression	SSR	p-1	$MSR = \frac{SSR}{p-1}$	$F_{reg} = \frac{MSR}{MSE}$
Error	SSE	n-p	$MSE = \frac{SSE}{n-p}$	
Lack of Fit	$SSLF$	c-p	$MSLF = \frac{SSLF}{c-p}$	$F_{lof} = \frac{MSLF}{MSPE}$
Pure Error	$SSPE$	n-c	$MSPE = \frac{SSPE}{n-c}$	
Total (corrected)	$SSTO$	n-1		

If there are repeat observations, and if the operator so specifies (see description of -flof command below), program 3dRegAna will perform the F-test for lack of fit for each voxel in the dataset. Those voxels which show a statistically significant lack of fit will be excluded from further analysis. The capability to test for lack of fit is particularly important for analysis of FMRI data, which involves many thousands of voxels. It is common for linear regression analysis software to omit a formal test for lack of fit. Typically, one would plot the linear regression function on top of the actual data, so there would be a visual indication of a lack of fit. However, for FMRI data, it is not practical to visually inspect each of the thousands of separate linear regression functions. Thus, an automatic procedure for testing for lack of fit is essential.

In the following discussion we will assume that, for the particular voxel under consideration, there is *no* lack of fit.

1.2.3 F-test for Significance of the Linear Regression

In building a model, one often wishes to determine if adding more parameters to the model is necessary (or, if certain parameters that are already in the model can be safely eliminated). Of course, the error sum of squares is reduced when more parameters are added. The question is, does the reduction in the error justify the additional complexity of the model? This usually takes the form of a statistical test whether adding additional parameters to the model results in a statistically significant improvement in the explanatory capability of the model. For the case of a simple linear regression,

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i, \quad i = 1, \ldots, n,$$

one would wish to test

$$\begin{aligned} H_o: &\quad \beta_1 = 0 \\ H_a: &\quad \beta_1 \neq 0 \end{aligned}$$

to determine if there is a linear relationship between the measured data Y_i and the X_i. More generally, suppose an investigator has already fitted the model:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_{q-1} X_{i,q-1} + \varepsilon_i,$$

and he is trying to decide whether to use the more elaborate model:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_{q-1} X_{i,q-1} + \beta_q X_{i,q} + \cdots + \beta_{p-1} X_{i,p-1} + \varepsilon_i$$

(we are assuming, without loss of generality, that the variables of interest are numbered sequentially). This is done by a test of the alternative hypotheses:

$$\begin{array}{ll} H_o: & \beta_q = \beta_{q+1} = \cdots = \beta_{p-1} = 0 \\ H_a: & \beta_k \neq 0, \text{ for some } k,\ q \leq k \leq p-1. \end{array}$$

A test of the null hypothesis is made by first calculating the error sum of squares for the *reduced model* ($SSE(R)$):

$$SSE(R) = \mathbf{Y}^t \left[\mathbf{I} - \mathbf{X}_r \left(\mathbf{X}_r^t \mathbf{X}_r\right)^{-1} \mathbf{X}_r^t\right] \mathbf{Y}$$

and then the error sum of squares for the *full model* ($SSE(F)$):

$$SSE(F) = \mathbf{Y}^t \left[\mathbf{I} - \mathbf{X}_f \left(\mathbf{X}_f^t \mathbf{X}_f\right)^{-1} \mathbf{X}_f^t\right] \mathbf{Y}$$

where the matrices $\mathbf{X}_r$ and $\mathbf{X}_f$ are the submatrices of the original independent variable matrix $\mathbf{X}$ obtained by extracting those columns corresponding to the reduced and full sets of variables:

$$\mathbf{X}_r = \begin{array}{c} \begin{array}{cccc} X_0 & X_1 & & X_{q-1} \end{array} \\ \begin{bmatrix} 1 & X_{11} & \ldots & X_{1,q-1} \\ 1 & X_{21} & \ldots & X_{2,q-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_{n1} & \ldots & X_{n,q-1} \end{bmatrix} \end{array} \qquad \mathbf{X}_f = \begin{array}{c} \begin{array}{cccccc} X_0 & X_1 & & X_{q-1} & & X_{p-1} \end{array} \\ \begin{bmatrix} 1 & X_{11} & \ldots & X_{1,q-1} & \ldots & X_{1,p-1} \\ 1 & X_{21} & \ldots & X_{2,q-1} & \ldots & X_{2,p-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_{n1} & \ldots & X_{n,q-1} & \ldots & X_{n,p-1} \end{bmatrix} \end{array}$$

We have reason to reject the null hypothesis if $SSE(F)$ is much less than $SSE(R)$. However, if $SSE(F)$ is only slightly smaller than $SSE(R)$, then we do not have reason to reject the null hypothesis. Consider the test statistic F^*_{reg} :

$$\begin{aligned} F^*_{reg} &= \frac{MS(\text{Regression})}{MS(\text{Error})} = \frac{\dfrac{SSE(R) - SSE(F)}{df_R - df_F}}{\dfrac{SSE(F)}{df_F}} \\ &= \frac{SSE(X_0,\ldots,X_{q-1}) - SSE(X_0,\ldots,X_{p-1})}{(n-q)-(n-p)} \div \frac{SSE(X_0,\ldots,X_{p-1})}{n-p} \end{aligned}$$

where df_R is the number of degrees of freedom for the reduced model, and df_F is the number of degrees of freedom for the full model. Specifically, we have:

$$\begin{aligned} df_R &= n-q, \\ df_F &= n-p, \\ \text{so } df_R - df_F &= p-q. \end{aligned}$$

The above can be summarized in the following ANOVA table:

Source	SS	df	MS	F
Error (reduced)	$SSE(R)$	n-q	$MSE(R) = \frac{SSE(R)}{n-q}$	
Regression	SSR	p-q	$MSR = \frac{SSR}{p-q}$	$F_{reg} = \frac{MSR}{MSE(F)}$
Error (full)	$SSE(F)$	n-p	$MSE(F) = \frac{SSE(F)}{n-p}$	
Lack of Fit	$SSLF$	c-p	$MSLF = \frac{SSLF}{c-p}$	$F_{lof} = \frac{MSLF}{MSPE}$
Pure Error	$SSPE$	n-c	$MSPE = \frac{SSPE}{n-c}$	
Total (corrected)	$SSTO$	n-1		

By the above reasoning, we see that a large value for F^*_{reg} indicates that the full regression model is significant, whereas a small value for F^*_{reg} suggests that the full model is not significantly better at explaining the data than the reduced model. The test statistic F^*_{reg} has an $F(p-q, n-p)$ distribution under the null hypothesis ([1],[2]). Therefore, the decision rule for the above test of hypothesis is:

$$\begin{aligned} \text{if } F^*_{reg} &\leq F(1-\alpha, p-q, n-p), \text{ conclude } H_o \\ \text{if } F^*_{reg} &> F(1-\alpha, p-q, n-p), \text{ conclude } H_a. \end{aligned}$$

Program 3dRegAna calculates the F^*_{reg} statistic for each voxel and, if so requested by the user, appends these values as the second sub-brick of an *AFNI* "fift" 3d dataset.

1.2.4 Coefficient of Multiple Determination $\mathbf{R}^2$

The coefficient of multiple determination, R^2, can be used as an indicator for how well the full model fits the data. R^2 is defined:

$$R^2 = 1 - \frac{SSE(F)}{SSTO}$$

where $SSTO$ is the total (corrected for the mean) sum of squares:

$$\begin{aligned} SSTO &= \sum_{i=1}^{n} \left(Y_i - \bar{Y}\right)^2 \\ &= \mathbf{Y}^t \left[\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^t\right] \mathbf{Y} \end{aligned}$$

(here, $\mathbf{1}$ is the column vector of 1's). Roughly speaking, R^2 is the proportion of the variation in the data (about the mean) that is explained by the full regression model. Note that, for every voxel, $0 \leq R^2 \leq 1$. Also, note that the reduced model plays no role in calculating R^2.

Program 3dRegAna calculates R^2 for each voxel and, if so requested by the user, appends these values as the second sub-brick of an *AFNI* "fith" 3d dataset.

1.2.5 t-test for Significance of Individual Parameters

When developing linear regression models, it is frequently useful to know the significance of the individual parameters that constitute the model. This may help identify terms in the model that may be safely discarded in order to simplify the model. Therefore, linear regression programs often provide the t-statistics for the individual parameters in the model.

For the linear regression model, the variance-covariance matrix for the regression coefficients is given by:

$$\begin{aligned}\mathbf{s}^2(\mathbf{b}) &= MSE \cdot \left(\mathbf{X}_f^t \mathbf{X}_f\right)^{-1} \\ &= \frac{SSE(F)}{n-p} \cdot \left(\mathbf{X}_f^t \mathbf{X}_f\right)^{-1}\end{aligned}$$

The standard deviation for the kth parameter is estimated by the square root of the (k,k)th element of the variance-covariance matrix:

$$s(b[k]) = \sqrt{[\mathbf{s}^2(\mathbf{b})]_{k,k}}$$

Then the test statistic t^* :

$$t^*[k] = \frac{b[k]}{s(b[k])}$$

has a $t(n-p)$ distribution under the null hypothesis ([1],[2]).

Note that the reduced model definition plays no role in the calculation of t^*.

Program 3dRegAna calculates the t^* statistic for each parameter at each voxel location and, if so requested by the user, appends these values as the second sub-brick of an *AFNI* "fitt" dataset for each parameter.

1.2.6 Summary

The overall procedure is summarized below.

1. Extract the independent variable matrices for the reduced and full models: $\mathbf{X}_r$ and $\mathbf{X}_f$. Calculate the matrices:

$$\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^t, \quad \mathbf{I} - \mathbf{X}_r\left(\mathbf{X}_r^t\mathbf{X}_r\right)^{-1}\mathbf{X}_r^t, \quad \left(\mathbf{X}_f^t\mathbf{X}_f\right)^{-1},$$
$$\left(\mathbf{X}_f^t\mathbf{X}_f\right)^{-1}\mathbf{X}_f^t, \quad \text{and} \quad \mathbf{I} - \mathbf{X}_f\left(\mathbf{X}_f^t\mathbf{X}_f\right)^{-1}\mathbf{X}_f^t.$$

   Since these matrices are constant for all voxels, these calculations are done only once.

2. Calculate the total (corrected for the mean) sum of squares:

$$SSTO = \mathbf{Y}^t\left[\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^t\right]\mathbf{Y}$$

   If $SSTO$ is very small, skip steps (3)—(7) (see the -rmsmin command, explained below).

3. Calculate the error sum of squares for the *reduced model*:

$$SSE(R) = \mathbf{Y}^t \left[\mathbf{I} - \mathbf{X}_r \left(\mathbf{X}_r^t \mathbf{X}_r\right)^{-1} \mathbf{X}_r^t\right] \mathbf{Y}$$

4. Calculate the error sum of squares for the *full model*:

$$SSE(F) = \mathbf{Y}^t \left[\mathbf{I} - \mathbf{X}_f \left(\mathbf{X}_f^t \mathbf{X}_f\right)^{-1} \mathbf{X}_f^t\right] \mathbf{Y}$$

5. If repeat observations are available, calculate the pure error sum of squares and lack of fit:

$$SSPE = \sum_{j=1}^{c} \sum_{i=1}^{n(j)} \left(Y_{ij} - \bar{Y}_j\right)^2$$
$$SSLF = SSE(F) - SSPE$$
$$F^*_{lof} = \frac{SSLF/(n-p)}{SSPE/(c-p)}$$

   If F^*_{lof} is very large, skip steps (6)—(7) (see the -flof command, explained below).

6. Calculate the linear regression fit of the *full model*:

$$\mathbf{b} = \left[\left(\mathbf{X}_f^t \mathbf{X}_f\right)^{-1} \mathbf{X}_f^t\right] \mathbf{Y}$$

$$\mathbf{s}^2(\mathbf{b}) = \frac{SSE(F)}{n-p} \cdot \left[\left(\mathbf{X}_f^t \mathbf{X}_f\right)^{-1}\right]$$

7. Calculate statistics for the linear regression:

$$F^*_{reg} = \frac{MS(\text{Regression})}{MS(\text{Error})} = \frac{\dfrac{SSE(R) - SSE(F)}{df_R - df_F}}{\dfrac{SSE(F)}{df_F}}$$

$$R^2 = 1 - \frac{SSE(F)}{SSTO}$$
$$t^*[k] = \frac{b[k]}{s(b[k])}$$

8. Repeat steps (2)—(7) for each voxel in the data set.

1.3 Usage

The syntax for execution of program 3dRegAna is as follows:

3dRegAna -rows n -cols m
-xydata $\mathbf{X}_{11}$ $\mathbf{X}_{12}$ ... $\mathbf{X}_{1m}$ filename
$\vdots$
-xydata $\mathbf{X}_{n1}$ $\mathbf{X}_{n2}$ ... $\mathbf{X}_{nm}$ filename
-model $\mathbf{i}_1$ ... $\mathbf{i}_{p-q}$: $\mathbf{j}_1$ ... $\mathbf{j}_q$
[-diskspace] [-workmem mega] [-rmsmin r] [-fdisp f] [-flof $\boldsymbol{\alpha}$]
[-fcoef k prefixname] [-rcoef k prefixname] [-tcoef k prefixname]
[-bucket n prefixname] [-brick $\mathbf{m}_1$ options] ... [-brick $\mathbf{m}_n$ options]

The different command line options are explained below.

1.4 Option

-rows n
The mandatory -rows command is used to indicate the number n of input datasets; equivalently, n is the number of rows in the independent variable matrix. The -rows command must appear *first* in the batch command file.

-cols m
The mandatory -cols command is used to indicate the number m of independent X variables, whose values will be entered for each dataset. Note that the number m does *not* include the column of 1's, which is automatically supplied by the program. The -cols commands must appear *second* in the batch command file.

-xydata $\mathbf{X}_{i1}$ $\mathbf{X}_{i2}$ ... $\mathbf{X}_{im}$ *fname*
The -xydata command specifies the values for the m independent X variables that apply for the observed *AFNI* 3d dataset whose filename is *fname*.

-model $\mathbf{i}_1$ ... $\mathbf{i}_{p-q}$: $\mathbf{j}_1$ ... $\mathbf{j}_q$ The -model command is used to specify the independent variables for the reduced and the full models. The integers j_1, ..., j_q following the ":" are the indices for the independent variables in the reduced model, i.e.,

$$\hat{Y} = f_r(X_{j_1}, \ldots, X_{j_q}).$$

The integers i_1, ..., i_{p-q} preceding the ":" are the indices for the independent variables in the full model which are not already in the reduced model, i.e.,

$$\hat{Y} = f_f(X_{j_1}, \ldots, X_{j_q}, X_{i_1}, \ldots, X_{i_{p-q}}).$$

Note: The index 0 *must always* appear to the right of the ":". This means that the constant term is always in the reduced model.

-diskspace

The -diskspace command tells program 3dRegAna to calculate how much free disk space is required to solve the given problem. The amount of disk space required depends upon the dimensions of the input datasets (i.e., number of voxels per image) as well as the number of output datasets requested. The program prints to the screen how much disk space is required, and a tabulation of the disk space currently available on the system. The program then asks the operator if he wishes to continue.

-workmem mega

The optional -workmem command specifies the number of megabytes of RAM to use for the statistical workspace. The default value is 12. The program will (usually) run faster if this value is set higher.

-rmsmin r

The optional -rmsmin command is used to set the minimum rms error r required in order to reject the constant response model. The full model will *not* be calculated for those voxels whose measured data, when fitted with the constant response model, has error rms $< r$. More precisely, if $\sqrt{\frac{SSTO}{n-1}} < r$, then no further calculations are performed for this voxel. This option is used primarily to speed program execution by screening out voxels which lie outside the brain. The user should choose a value for r which is much smaller than the natural measurement error. The default value is $r = 0$.

-fdisp f

The optional -fdisp command is used to control output to the user's terminal during program execution. For each voxel in the data set, if the calculated F^*_{reg} is greater than or equal to f, then the estimated full model regression coefficients, F_{reg} statistic, R^2, and t-statistics, are written to the screen; otherwise, nothing is written to the screen for that particular voxel. Note that the -fdisp command effects screen output only, and has absolutely no effect upon the data file output generated by the program.

-flof α

The optional -flof command is used to set the threshold for discarding voxels due to lack of fit of the full model. The value of α, $0 \leq \alpha \leq 1$, is the user set probability for falsely rejecting the null hypothesis that the full model is adequate for explaining variation in the data. Note: if α is set to 0, all voxels would be accepted; if α is set to 1, all voxels would be rejected. Given α, the value F_{lof} is calculated which yields a cumulative probability of $1-\alpha$ under the F-distribution, with $c-p$ for the numerator degrees of freedom, and $n-c$ for the denominator degrees of freedom, i.e.,

$$F_{lof} = F(1-\alpha, c-p, n-c).$$

For each voxel in the data set, if the calculated F^*_{lof} is greater than or equal to F_{lof} (indicating that the full model is not adequate for explaining variation in the data), then the regression analysis is not performed for that voxel. All output parameters and statistics for that voxel are set to zero.

By default, the test for lack of fit is *not* performed. However, it is strongly recommended that this option be used whenever there are repeat observations in the data.

-fcoef k ***prefixname***

The optional -fcoef command tells program 3dRegAna to save the least squares estimated value of the kth parameter in an *AFNI* 3d data set, along with the F-statistics for significance of the regression. The output is then written to the file with the user specified prefix filename. This output consists of a 2 sub-brick *AFNI* dataset of type "fift". The first sub-brick consists of the regression parameter estimate $b[k]$, and the second sub-brick contains the corresponding F-statistics F^*_{reg}. It must be emphasized that the F-statistics pertain to significance of the overall regression, and do not indicate the significance of the individual parameter (compare with command -tcoef).

$$\text{\textit{AFNI} "fift" dataset} \quad \begin{cases} \boxed{b[k] = \text{L.S. est. of } \beta[k]} \\[2ex] \boxed{F^*_{reg} = \dfrac{MS(\text{Reg})}{MS(\text{Error})} = \dfrac{\dfrac{SSE(R) - SSE(F)}{df_R - df_F}}{\dfrac{SSE(F)}{df_F}}} \end{cases}$$

When this data file is used as input to program afni, the second sub-brick can be used as a "threshold" for determining which voxels "light-up"; the intensity is then determined by the first sub-brick. So, by setting the appropriate threshold, only those voxels having the user-specified p-values for significance of the linear regression will "light-up". The color coding of the voxels which light-up indicates the sign and magnitude of the estimated regression coefficients.

-rcoef k ***prefixname***

The optional -rcoef command tells program 3dRegAna to save the least squares estimated value of the kth parameter in an *AFNI* 3d data set, along with the coefficient of multiple determination R^2. The output is then written to the file with the user specified prefix filename. This output consists of a 2 sub-brick *AFNI* dataset of type "fith". The first sub-brick consists of the regression parameter estimate $b[k]$, and the second sub-brick contains the corresponding R^2. It must be emphasized that R^2 is an indicator of the efficacy of the overall regression, and does not indicate the significance of the individual parameter (compare with command -tcoef).

$$\text{\textit{AFNI} "fith" dataset} \quad \begin{cases} \boxed{b[k] = \text{L.S. est. of } \beta[k]} \\[2ex] \boxed{R^2 = 1 - \dfrac{SSE(F)}{SSTO}} \end{cases}$$

When this data file is used as input to program afni, the second sub-brick can be used as a "threshold" for determining which voxels "light-up"; the intensity is then determined by the first sub-brick. So, by setting the appropriate threshold, only those voxels exceeding the user-specified threshold value for the coefficient of multiple determination will "light-up". The color coding of the voxels which light-up indicates the sign and magnitude of the estimated regression coefficients.

-tcoef k ***prefixname***

The optional -tcoef command is used to save the estimated value of the kth parameter for the *full* regression model, as well as the corresponding t-statistic for significance of that parameter, for each voxel in the dataset. The result is stored as a 2 sub-brick *AFNI* data set of type "fitt" in the file with the user specified prefix filename. The first sub-brick consists of the regression parameter estimate $b[k]$, and the second sub-brick contains the corresponding t-statistic t^* for each voxel.

$$\text{\textit{AFNI} ``fitt'' dataset} \quad \left\{ \begin{array}{l} \boxed{b[k] = \text{L.S. est. of } \beta[k]} \\ \\ \boxed{t^* = \dfrac{b[k]}{s(b[k])}} \end{array} \right.$$

When this is used as an input file to program afni, the second sub-brick can be used to set the threshold for determining which voxels have an estimated parameter value which is significantly different from zero. The ".HEAD" file informs program afni that the second sub-brick contains t-statistics, and that $df = n - p$ should be used for the degrees of freedom.

-bucket n ***prefixname***

The -bucket command is used to create a single *AFNI* "bucket" type dataset having n sub-bricks. The output is written to the file with the user specified prefix filename. Each of the individual sub-bricks can then be accessed for display within program afni. The purpose of this command is to simplify file management, since all of the output datasets for a particular problem can now be contained within the single *AFNI* bucket dataset. (See Example 3).

If $n = 0$, then the *default* output bucket dataset is created. This default dataset has $2p + 2$ sub-bricks, as illustrated below.

Brick	Label	Contents
#0	Coef #0 est.	$b[0] = \text{L.S. est. of } \beta[0]$
#1	Coef #0 t-stat	$t^* = \dfrac{b[0]}{s(b[0])}$
#2	Coef #1 est.	$b[1] = \text{L.S. est. of } \beta[1]$
#3	Coef #1 t-stat	$t^* = \dfrac{b[1]}{s(b[1])}$
⋮	⋮	⋮
#2p-2	Coef #p-1 est.	$b[p-1] = \text{L.S. est. of } \beta[p-1]$
#2p-1	Coef #p-1 t-stat	$t^* = \dfrac{b[p-1]}{s(b[p-1])}$
#2p	F-stat Regression	$F^*_{reg} = \dfrac{MS(\text{Reg})}{MS(\text{Error})}$
#2p+1	Rˆ2 Regression	$R^2 = 1 - \dfrac{SSE(F)}{SSTO}$

If $n > 0$, then the contents and labels for the individual sub-bricks within the bucket dataset are specified by the user, by means of the -brick command described below. Note that the -bucket command *must* precede the -brick commands.

-brick m ***options***
The -brick command is used to specify the contents and labels for the mth sub-brick ($0 \leq m < n$) within the bucket dataset. There must be one -brick command for each of the n sub-bricks in the dataset (where n has been previously specified by the -bucket command). (See Example 3).

There are 4 versions of the -brick command:

-brick m coef k label	The mth sub-brick is to contain the least squares estimate of the kth regression coefficient.
-brick m tstat k label	The mth sub-brick is to contain the t-statistic for significance of the kth regression coefficient.
-brick m fstat label	The mth sub-brick is to contain the F-statistic for significance of the regression.
-brick m rstat label	The mth sub-brick is to contain the coefficient of multiple determination R^2.

In each case, the label for the sub-brick is specified by *label*.

1.5 Notes

- Since program 3dRegAna uses temporary disk files to hold the results of intermediate calculations, the user should be aware that there must be enough available disk space for the temporary files, in addition to the permanent output files. If the -diskspace command is used, then prior to the start of the linear regression calculations, program

3dRegAna calculates the maximum amount of disk space required to solve the given problem, and reports this value to the user, along with the amount of disk space currently available. The user has the option of halting program execution at this point if there is insufficient disk space.

- Program 3dRegAna will not overwrite previously existing files. Therefore, if a file already exists with the same name as an output file specified by the user, the program will report this fact and halt. This applies as well to the temporary data files created by the program. All temporary files created by program 3dRegAna have the suffix "`.3dregana`". If, for any reason, the program should terminate execution prematurely, the user must manually remove any "`.3dregana`" files that may be left.
- Program execution time depends on the number of input datasets and the number of voxels per dataset. Execution time can be reduced by first applying a mask to zero out non-brain voxels. Then, the -rmsmin command can be used to exclude these non-brain voxels from the regression analysis.
- This program cannot handle complex-valued datasets or time-dependent datasets.

1.6 Examples

Example 1. Simple Linear Regression It is hypothesized that there is a simple linear relationship between the frequency of an external excitation and the corresponding neural activation intensity.

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

To test this hypothesis, FMRI data is collected for excitation frequencies 1.5, 2.5, 3.5, and 4.5 Hz. Three images are produced for each excitation frequency, in randomized order. The batch command file which was used to execute program 3dRegAna is as follows:

Batch Command File for Example 1

```
3dRegAna  \
-rows 12 \
-cols 1 \
-xydata 1.5 fred005+orig \
-xydata 1.5 fred012+orig \
-xydata 1.5 fred010+orig \
-xydata 2.5 fred007+orig \
-xydata 2.5 fred001+orig \
-xydata 2.5 fred006+orig \
-xydata 3.5 fred008+orig \
-xydata 3.5 fred003+orig \
-xydata 3.5 fred009+orig \
```

```
-xydata 4.5 fred004+orig \
-xydata 4.5 fred011+orig \
-xydata 4.5 fred002+orig \
-diskspace \
-rmsmin 1.0 \
-fdisp 10 \
-model 1:0 \
-flof 0.01 \
-fcoef 0 fred.constant \
-fcoef 1 fred.linear
```

■

The commands -rows 12 is used to indicate that there are 12 input datasets. Since there is only one predictor variable, X_1 (not counting $X_0 \equiv 1$), the command -cols 1 appears next. This is followed by 12 -xydata commands, one for each dataset. Each -xydata command is immediately followed by the value of X_1 (i.e., the excitation frequency) for that dataset, and the name of the file containing the 3d dataset.

The -diskspace command means that the program is to display the amount of disk space required for program execution, along with the available disk space, prior to the start of the actual calculation. The command -rmsmin 1.0 indicates that the regression analysis should be calculated for a voxel only if the rms error from fitting just a constant to the data exceeds 1.0. The command -fdisp 10 indicates that the regression coefficients, F-statistics, R^2, and t-statistics should be written to the screen for those voxels whose F^*_{reg} is equal to or greater than 10.

Next, the -model 1 : 0 command indicates that the full model is given by:

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i.$$

and the reduced model is:

$$Y_i = \beta_0 + \varepsilon_i$$

Since there are repeat observations, the command -flof 0.01 is used to perform an F-test for lack of fit. This allows the test of hypotheses:

$$\begin{aligned} H_o: &\quad E(Y) = \beta_0 + \beta_1 X \\ H_a: &\quad E(Y) \neq \beta_0 + \beta_1 X \end{aligned}$$

at each voxel location. In this example, there are 4 unique values for X, hence $c = 4$, so,

$$\begin{aligned} F_{lof} &= F(1-\alpha, c-p, n-c) \\ &= F(0.99, 2, 8) \approx 8.65 \end{aligned}$$

Therefore, any voxel with $F^*_{lof} \geq 8.65$ is automatically excluded from further analysis due to lack of fit.

The output consist of two AFNI "fift" datasets: dataset fred.constant+orig.HEAD (and .BRIK) contains the estimate for the constant parameter β_0 for each voxel in the first sub-brick, and the F-statistics for significance of the linear regression in the second sub-brick;

dataset fred.linear+orig.HEAD (and .BRIK) contains the estimate for the linear slope parameter β_1 in the first sub-brick, and the F-statistics for significance of the linear regression in the second sub-brick. Note that the second sub-brick for these two datasets is identical; only the first sub-bricks differ. The second sub-brick provides the results for the test of hypotheses:

$$
\begin{aligned}
H_o: &\quad \beta_1 = 0 \\
H_a: &\quad \beta_1 \neq 0
\end{aligned}
$$

at each voxel location. Therefore, when viewed with afni, and for a given operator specified significance level, the same set of voxel locations will light up for both datasets (i.e., those voxels for which there is a statistically significant linear dependence of FMRI intensity upon task excitation frequency). However, the color coding of the voxels will vary depending upon the magnitude of the respective regression coefficient at each voxel location.

Example 2. Polynomial Regression A researcher is using FMRI to study neural activation as a function of drug dosage. It is suspected that some voxels will show a linear (intensity) response to drug dosage level, whereas other voxels may show a quadratic response. Two images were obtained at each of five drug dosage levels. The following batch commands produce a least squares fit of the data to a quadratic drug response model:

Batch Command File for Example 2

```
3dRegAna  \
-rows 10 \
-cols 2 \
-xydata 0.5 0.25 ethel001+orig \
-xydata 0.5 0.25 ethel004+orig \
-xydata 1.5 2.25 ethel002+orig \
-xydata 1.5 2.25 ethel000+orig \
-xydata 2.5 6.25 ethel008+orig \
-xydata 2.5 6.25 ethel006+orig \
-xydata 3.5 16.00 ethel007+orig \
-xydata 3.5 16.00 ethel005+orig \
-xydata 4.5 20.25 ethel009+orig \
-xydata 4.5 20.25 ethel003+orig \
-diskspace \
-rmsmin 1.0 \
-fdisp 10 \
-model 2 1 :  0 \
-flof 0.01 \
-tcoef 0 ethel.constant \
-tcoef 1 ethel.linear \
```

```
-tcoef 2 ethel.quadratic \
-fcoef 2 ethel.regression
```

■

The command -rows 10 is used to indicate that there are 10 input datasets. Since there are two predictor variables, X_1 and $X_2 \equiv X_1^2$, the command -cols 2 appears next. This is followed by the 10 -xydata commands, one for each dataset. Each -xydata command is immediately followed by the value of X_1 (i.e., the drug dosage level) and X_1^2 for that dataset, and the name of the file containing the corresponding 3d dataset.

Next, the -model 2 1 : 0 command indicates that the program should calculate the reduced model

$$Y_i = \beta_0 + \varepsilon_i$$

and the full model

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \varepsilon_i.$$

where X_i = drug dosage level, and Y_i = measured response. This allows the test of hypotheses:

$$\begin{array}{ll} H_o: & \beta_1 = \beta_2 = 0 \\ H_a: & \beta_1 \neq 0 \text{ or } \beta_2 \neq 0 \end{array}$$

at each voxel location. Thus, the command -model 2 1 : 0 tests for the statistical significance of adding both variables X and X^2 to the model. That is, do X and X^2, combined, significantly improve the explanatory power of the model?

Since there are repeat observations, the command -flof 0.01 is used to perform an F-test for lack of fit. This allows the test of hypotheses:

$$\begin{array}{ll} H_o: & E(Y) = \beta_0 + \beta_1 X + \beta_2 X_i^2 \\ H_a: & E(Y) \neq \beta_0 + \beta_1 X \ + \beta_2 X_i^2 \end{array}$$

at each voxel location. In this example, there are 5 unique values for $\mathbf{X}$, hence $c = 5$, so,

$$\begin{aligned} F_{lof} &= F(1-\alpha, c-p, n-c) \\ &= F(0.99, 3, 5) \approx 12.1 \end{aligned}$$

Therefore, any voxel with $F^*_{lof} \geq 12.1$ is automatically excluded from further analysis due to lack of fit.

The last four commands generate the output files. The three -tcoef commands each creates an *AFNI* "fitt" 3d dataset; the first sub-brick contains the least squares estimate of the regression coefficient for the respective independent variable (from the full model), and the second sub-brick contains the t-statistic for statistical significance of that parameter (being different from zero) within the full model. The -fcoef command creates an *AFNI* "fift" 3d dataset; the second sub-brick contains the F-statistics for significance of the regression.

Note that just by changing the -model command, one can test various alternative hypotheses:

	H_a		H_o
-model 1 : 0	$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$	vs.	$Y_i = \beta_0 + \varepsilon_i$
-model 2 : 0	$Y_i = \beta_0 + \beta_2 X_i^2 + \varepsilon_i$	vs.	$Y_i = \beta_0 + \varepsilon_i$
-model 2 : 1 0	$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \varepsilon_i$	vs.	$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$
-model 2 1 : 0	$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \varepsilon_i$	vs.	$Y_i = \beta_0 + \varepsilon_i$

Example 3. Multiple Linear Regression Five predictor variables, X_1, X_2, X_3, X_4, and X_5, were recorded for each of a set of 100 FMRI images, where
X_1 = drug dosage level,
X_2 = (drug dosage level)^2,
X_3 = subject's age,
X_4 = subject's weight,
X_5 = drug dosage level $\div$ subject's weight.

The investigator has already determined that variables X_1 and X_2 are useful in explaining the observed variation in the neural activation data. Now, he wishes to determine if adding variables X_4 and X_5 (both related to the subject's weight) to the model will significantly improve the predictive ability of the model. The following batch command file could be used to perform the test.

Batch Command File for Example 3

```
3dRegAna  \
-rows 100 \
-cols 5 \
 -xydata  1.30   1.69   32  127  0.0102  a366+tlrc \
 -xydata  2.50   6.25   47  165  0.0152  k894+tlrc \
 -xydata  3.20  10.24   25  225  0.0142  c804+tlrc \
    ⋮       ⋮      ⋮     ⋮    ⋮      ⋮          ⋮
 -xydata  2.25   5.06   29  107  0.0210  w904+tlrc \
 -xydata  1.75   3.06   39  250  0.0070  f011+tlrc \
 -xydata  2.75   7.56   44  120  0.0229  m515+tlrc \
-diskspace \
-rmsmin 1.0 \
-fdisp 10 \
-model 4 5 :  0 1 2 \
-fcoef 1 dosage \
-fcoef 2 dosage^2 \
-fcoef 4 weight \
-fcoef 5 dose%wght
```

■

The command -rows 100 indicates that 100 *AFNI* 3d datasets will be used as input. This is followed by -cols 5, indicating that 5 numerical values will be entered for each dataset. Following this there are 100 separate -xydata commands, one for each dataset.

The command -model 4 5 : 0 1 2 is used to test the alternative models:

$$
\begin{aligned}
H_o : &\quad Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i \\
H_a : &\quad Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_4 X_{i4} + \beta_5 X_{i5} + \varepsilon_i
\end{aligned}
$$

In other words, this is a test for the significance of variables X_4 and X_5, assuming that variables X_1 and X_2 are *already* in the model.

For this particular example, the reduced and full independent variable matrices are given by:

$$
\mathbf{X}_r = \begin{array}{c} \begin{array}{ccc} X_0 & X_1 & X_2 \end{array} \\ \begin{bmatrix} 1 & 1.30 & 1.69 \\ 1 & 2.50 & 6.25 \\ 1 & 3.20 & 10.24 \\ \vdots & \vdots & \vdots \\ 1 & 2.25 & 5.06 \\ 1 & 1.75 & 3.06 \\ 1 & 2.75 & 7.56 \end{bmatrix} \end{array}
\qquad
\mathbf{X}_f = \begin{array}{c} \begin{array}{ccccc} X_0 & X_1 & X_2 & X_4 & X_5 \end{array} \\ \begin{bmatrix} 1 & 1.30 & 1.69 & 127 & 0.0102 \\ 1 & 2.50 & 6.25 & 165 & 0.0152 \\ 1 & 3.20 & 10.24 & 225 & 0.0142 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 2.25 & 5.06 & 107 & 0.0210 \\ 1 & 1.75 & 3.06 & 250 & 0.0070 \\ 1 & 2.75 & 7.56 & 120 & 0.0229 \end{bmatrix} \end{array}
$$

Note that the 3rd column of data is not used. It is *not* necessary to rewrite the data portion of the batch command file in order to experiment with different sets of predictor variables. Only the -model command (and the output commands) need be changed.

The -fcoef commands create 4 *AFNI* "fift" 3d datasets; the first sub-brick of each contains the respective parameter estimate; the second sub-brick contains the F-statistics for significance of the regression. The second sub-brick is identical for each of these 4 datasets, only the first sub-bricks differ.

When afni is used to view the above "fift" datasets, only those voxels for which variables X_4 and X_5 are statistically significant in explaining variation in the data (variation that is *not* already accounted for by variables X_0, X_1, and X_2) will "light-up". The color coding of the voxels will correspond to the respective parameter estimates.

As we have seen, this batch command file generates 4 separate *AFNI* 3d datasets (8 files, counting the '.HEAD' and '.BRIK' files). To simplify file management, all of these datasets can be grouped into a single AFNI "bucket" dataset by appending the -bucket command, as shown below.

Batch Command File for Example 3

```
3dRegAna  \
-rows 100 \
-cols 5 \
```

```
        ⋮
(same as above)
        ⋮
-model 4 5 :  0 1 2 \
-bucket 0 drug.results
```

■

As may be seen, the batch command file is the same as before, except that the 4 -fcoef commands have been replaced by the single -bucket command. The '0' following -bucket indicates that the default dataset should be created; the AFNI bucket dataset is written to file 'drug.results+tlrc' (.HEAD and .BRIK). The structure of the bucket dataset is described in the following table.

Brick	Label	Contents
#0	Coef #0 est.	$b[0]$ = L.S. est. of $\beta[0]$
#1	Coef #0 t-stat	t^* for $b[0]$
#2	Coef #1 est.	$b[1]$ = L.S. est. of $\beta[1]$
#3	Coef #1 t-stat	t^* for $b[1]$
#4	Coef #2 est.	$b[2]$ = L.S. est. of $\beta[2]$
#5	Coef #2 t-stat	t^* for $b[2]$
#6	Coef #4 est.	$b[4]$ = L.S. est. of $\beta[4]$
#7	Coef #4 t-stat	t^* for $b[4]$
#8	Coef #5 est.	$b[5]$ = L.S. est. of $\beta[5]$
#9	Coef #5 t-stat	t^* for $b[5]$
#10	F-stat Regression	F^*_{reg}
#11	R^2 Regression	R^2

The default option for the -bucket command is easy to use, but there are a couple limitations. First, the labels generated by the program for the sub-bricks are generic; i.e., they are not very informative. The second limitation is that everything is included in the output. However, the user may only be interested in certain specific parameters or statistics; the remainder may only clutter-up the output. The alternative approach is for the user to specify what goes into each sub-brick, and the label for each sub-brick, as illustrated below.

Batch Command File for Example 3

```
3dRegAna  \
-rows 100 \
-cols 5 \
      ⋮
```

(same as above)

⋮

```
-model 4 5 :  0 1 2 \
-bucket 5 drug.results \
-brick 0 coef 1 Dosage \
-brick 1 coef 2 Dosage^2 \
-brick 2 coef 4 Weight \
-brick 3 coef 5 ''Dose/Weight'' \
-brick 4 fstat ''F-statistic''
```

■

The '5' following -bucket indicates that the the bucket dataset will contain 5 sub-bricks. The AFNI bucket dataset is written to file 'drug.results+tlrc' (.HEAD and .BRIK), as before. The 5 -brick commands are used to specify the contents of the individual sub-bricks, as well as their labels.

Brick	Label	Contents
#0	Dosage	$b[1]$ = L.S. est. of $\beta[1]$
#1	Dosage^2	$b[2]$ = L.S. est. of $\beta[2]$
#2	Weight	$b[4]$ = L.S. est. of $\beta[4]$
#3	Dose/Weight	$b[5]$ = L.S. est. of $\beta[5]$
#4	F-statistic	F^*_{reg}

We see that the labels for the individual sub-bricks are more informative. Also, the dataset contains only the output quantities of interest to the user.

Example 4. Quantitative and Qualitative Independent Variables In the previous examples, the independent (predictor) variables always appeared at quantitative levels. Now, we will consider an example where one of the predictor variables is quantitative, and the other is qualitative.

We will again suppose that the drug dosage level is the quantitative factor: X_1 = drug dosage. For the qualitative factor, we will use whether the subject is a non-smoker, a light smoker, or a heavy smoker. We could define predictor variable X_2:

$$X_2 = \begin{cases} 0, & \text{subject is non-smoker} \\ 1, & \text{subject is light smoker} \\ 2, & \text{subject is heavy smoker} \end{cases}$$

and then fit the linear regression model:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i$$

The problem with this approach is that it inherently assumes a linear relationship between the response and the levels of variable X_2; i.e., "level of smoking" is being treated as a quantitative variable.

In order to treat "level of smoking" qualitatively, each level of smoking could be given an indicator variable, e.g.:

$$X_2 = \begin{cases} 1, & \text{subject is non-smoker} \\ 0, & \text{otherwise} \end{cases}$$

$$X_3 = \begin{cases} 1, & \text{subject is light smoker} \\ 0, & \text{otherwise} \end{cases}$$

$$X_4 = \begin{cases} 1, & \text{subject is heavy smoker} \\ 0, & \text{otherwise} \end{cases}$$

and then fit the linear regression model:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_4 X_{i4} + \varepsilon_i$$

However, this causes computational difficulties, as may be seen by examining the $\mathbf{X}_f$ matrix:

$$\mathbf{X}_f = \begin{array}{c} \begin{array}{ccccc} \scriptstyle X_0 & \scriptstyle X_1 & \scriptstyle X_2 & \scriptstyle X_3 & \scriptstyle X_4 \end{array} \\ \begin{bmatrix} 1 & 0.5 & 1 & 0 & 0 \\ 1 & 1.5 & 1 & 0 & 0 \\ 1 & 2.5 & 1 & 0 & 0 \\ 1 & 3.5 & 1 & 0 & 0 \\ 1 & 1.5 & 0 & 1 & 0 \\ 1 & 2.0 & 0 & 1 & 0 \\ 1 & 4.2 & 0 & 1 & 0 \\ 1 & 0.5 & 0 & 1 & 0 \\ 1 & 2.4 & 0 & 0 & 1 \\ 1 & 0.8 & 0 & 0 & 1 \\ 1 & 3.6 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

In terms of column vectors, $X_0 = X_2 + X_3 + X_4$, i.e., the column vectors are *not* linearly independent. This makes it impossible to invert the $\mathbf{X}_f^t \mathbf{X}_f$ matrix. The solution is simple: eliminate one of the indicator variables. A natural choice in this case would be:

$$X_2 = \begin{cases} 1, & \text{subject is light smoker} \\ 0, & \text{otherwise} \end{cases}$$

$$X_3 = \begin{cases} 1, & \text{subject is heavy smoker} \\ 0, & \text{otherwise} \end{cases}$$

in which case the full regression model is:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i$$

The implication of this model is that the response to the drug consists of 3 parallel linear functions:

$$\begin{array}{ll} E(Y) = \beta_0 + \beta_1 X_{i1} & \text{for non-smokers} \\ E(Y) = (\beta_0 + \beta_2) + \beta_1 X_{i1} & \text{for light smokers} \\ E(Y) = (\beta_0 + \beta_3) + \beta_1 X_{i1} & \text{for heavy smokers} \end{array}$$

For this model, the predictor variable matrix is:

$$\mathbf{X}_f = \begin{array}{c} \begin{array}{cccc} X_0 & X_1 & X_2 & X_3 \end{array} \\ \begin{bmatrix} 1 & 0.5 & 0 & 0 \\ 1 & 1.5 & 0 & 0 \\ 1 & 2.5 & 0 & 0 \\ 1 & 3.5 & 0 & 0 \\ 1 & 1.5 & 1 & 0 \\ 1 & 2.0 & 1 & 0 \\ 1 & 4.2 & 1 & 0 \\ 1 & 0.5 & 1 & 0 \\ 1 & 2.4 & 0 & 1 \\ 1 & 0.8 & 0 & 1 \\ 1 & 3.6 & 0 & 1 \end{bmatrix} \end{array}$$

and the corresponding batch command file is:

Batch Command File for Example 4

```
3dRegAna  \
-rows 11 \
-cols 3 \
 -xydata  0.5  0  0  d338+tlrc \
 -xydata  1.5  0  0  h289+tlrc \
 -xydata  2.5  0  0  f040+tlrc \
 -xydata  3.5  0  0  a202+tlrc \
 -xydata  1.5  1  0  c499+tlrc \
 -xydata  2.0  1  0  b437+tlrc \
 -xydata  4.2  1  0  i746+tlrc \
 -xydata  0.5  1  0  g846+tlrc \
 -xydata  2.4  0  1  e325+tlrc \
 -xydata  0.8  0  1  h820+tlrc \
 -xydata  3.6  0  1  g920+tlrc \
-diskspace \
-rmsmin 1.0 \
-fdisp 10 \
-model 1 2 3 :  0 \
-fcoef 0 b0 \
-fcoef 1 b1 \
-fcoef 2 b2 \
-fcoef 3 b3
```

■

As previously mentioned, the above model assumes that the slope of the response function is the same for each of the 3 groups. If we wish to consider a more general model, in which each of the 3 groups has a different slope, then we would use the following model, which includes products of the independent variables:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_4 X_{i2} X_{i1} + \beta_5 X_{i3} X_{i1} + \varepsilon_i$$

The response function for this model consists of 3 linear functions having different slopes:

$$\begin{array}{ll} E(Y) = \beta_0 + \beta_1 X_{i1} & \text{for non-smokers} \\ E(Y) = (\beta_0 + \beta_2) + (\beta_1 + \beta_4) X_{i1} & \text{for light smokers} \\ E(Y) = (\beta_0 + \beta_3) + (\beta_1 + \beta_5) X_{i1} & \text{for heavy smokers} \end{array}$$

For this model, the predictor variable matrix is:

$$\mathbf{X}_f = \begin{array}{c} \begin{array}{cccccc} X_0 & X_1 & X_2 & X_3 & X_4 & X_5 \end{array} \\ \left[\begin{array}{cccccc} 1 & 0.5 & 0 & 0 & 0 & 0 \\ 1 & 1.5 & 0 & 0 & 0 & 0 \\ 1 & 2.5 & 0 & 0 & 0 & 0 \\ 1 & 3.5 & 0 & 0 & 0 & 0 \\ 1 & 1.5 & 1 & 0 & 1.5 & 0 \\ 1 & 2.0 & 1 & 0 & 2.0 & 0 \\ 1 & 4.2 & 1 & 0 & 4.2 & 0 \\ 1 & 0.5 & 1 & 0 & 0.5 & 0 \\ 1 & 2.4 & 0 & 1 & 0 & 2.4 \\ 1 & 0.8 & 0 & 1 & 0 & 0.8 \\ 1 & 3.6 & 0 & 1 & 0 & 3.6 \end{array} \right] \end{array}$$

and the corresponding batch command file is:

Batch Command File for Example 4

```
3dRegAna  \
-rows 11 \
-cols 5 \
 -xydata 0.5 0 0   0   0  d338+tlrc \
 -xydata 1.5 0 0   0   0  h289+tlrc \
 -xydata 2.5 0 0   0   0  f040+tlrc \
 -xydata 3.5 0 0   0   0  a202+tlrc \
 -xydata 1.5 1 0 1.5   0  c499+tlrc \
 -xydata 2.0 1 0 2.0   0  b437+tlrc \
 -xydata 4.2 1 0 4.2   0  i746+tlrc \
 -xydata 0.5 1 0 0.5   0  g846+tlrc \
 -xydata 2.4 0 1   0 2.4  e325+tlrc \
 -xydata 0.8 0 1   0 0.8  h820+tlrc \
 -xydata 3.6 0 1   0 3.6  g920+tlrc \
-diskspace \
```

```
-rmsmin 1.0 \
-fdisp 10 \
-model 4 5 :  0 1 2 3 \
-fcoef 0 b0 \
-fcoef 1 b1 \
-fcoef 2 b2 \
-fcoef 3 b3 \
-fcoef 4 b4 \
-fcoef 5 b5
```

■

The command `-model 4 5 :  0 1 2 3` provides a test of the alternative hypotheses:

$$H_o: \quad Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i$$
$$H_a: \quad Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_4 X_{i2} X_{i1} + \beta_5 X_{i3} X_{i1} + \varepsilon_i$$

That is, when viewed with program afni, only those voxels for which the slopes of the response functions for the 3 different groups are significantly different will light up.

Note that this is only intended as an illustrative example; in practice, the sample size is too small for estimation of so many parameters.

Example 5. ANOVA with Unequal Sample Sizes The multifactor ANOVA programs (3dANOVA2 and 3dANOVA3) require equal sample sizes for each combination of factor levels (i.e., each treatment). However, it may happen that one or more of the treatments have missing observations (e.g., a subject may not show up at the scheduled time, etc.), or it may be difficult to secure equal numbers of volunteer subjects for each combination of factor levels. We will consider how to use the regression approach to performing ANOVA when there are unequal sample sizes. Since this involves the use of indicator variables, it should be noted that, as a practical matter, the number of levels for each factor must not be very large.

Suppose a researcher wishes to study differences in neural activation in response to a particular task arising from two factors: intelligence level, and smoking vs. non-smoking. Factor A (smoking) is evaluated at 2 levels: non-smoking and smoking. Factor B (intelligence) has 3 levels: low IQ, medium IQ, and high IQ. Volunteers are selected at random, and are classified as shown in the following table.

	low IQ	medium IQ	high IQ
Non-smoking	Y_{111}	Y_{121}	Y_{131}
	Y_{112}	Y_{122}	Y_{132}
	Y_{113}	Y_{123}	
		Y_{124}	
Smoking	Y_{211}	Y_{221}	Y_{231}
	Y_{212}	Y_{222}	
	Y_{213}		

In this table, each Y_{ijik} represents an *AFNI* 3d dataset. Because of the unequal sample sizes, program 3dANOVA2 is not applicable. However, the analysis can be performed using linear regression by means of *indicator variables* (see [1], pp.746-754). Indicator variables are just predictor variables which take on only the values +1, -1, and 0, as explained below.

The two-factor ANOVA model is given by :

$$\begin{aligned} Y_{ijk} &= \mu_{..} + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}, \\ \text{where } i &= 1,2; \text{ and } j = 1,2,3, \end{aligned}$$

and subject to the constraints: $\sum \alpha_i = 0$, $\sum \beta_j = 0$, $\sum_i (\alpha\beta)_{ij} = 0$, $\sum_j (\alpha\beta)_{ij} = 0$. These constraints are important, since the $\mathbf{X}$ matrix must be set up so as to have *linearly independent* columns (otherwise, it would not be possible to calculate $(\mathbf{X}^t\mathbf{X})^{-1}$).

In this particular example, we have $\alpha_2 = -\alpha_1$, so we require only $2 - 1 = 1$ indicator variable for factor A. This indicator variable, X_1, is defined below.

$$X_1 = \begin{cases} +1 & \text{if observation corresponds to non-smoker,} \\ -1 & \text{if observation corresponds to smoker.} \end{cases}$$

Since $\beta_3 = -\beta_1 - \beta_2$, we require only $3 - 1 = 2$ indicator variables for factor B. These indicator variables, X_2 and X_3, are defined:

$$X_2 = \begin{cases} +1 & \text{if observation corresponds to low IQ,} \\ 0 & \text{if observation corresponds to medium IQ,} \\ -1 & \text{if observation corresponds to high IQ,} \end{cases}$$
$$X_3 = \begin{cases} 0 & \text{if observation corresponds to low IQ,} \\ +1 & \text{if observation corresponds to medium IQ,} \\ -1 & \text{if observation corresponds to high IQ,} \end{cases}$$

Also, we require only $(2 - 1) \times (3 - 1) = 2$ indicator variables for AB interaction. These are obtained as the product of the factor A and factor B indicator variables:

$X_4 = X_1 \times X_2$
$X_5 = X_1 \times X_3$

Using these definitions of the indicator variables, the full linear regression model can be written:

$$\begin{aligned} Y_{ijk} &= \mu_{..} + \overbrace{\alpha_1 X_{ijk1}}^{\text{A main effect}} + \overbrace{\beta_1 X_{ijk2} + \beta_2 X_{ijk3}}^{\text{B main effect}} \\ &\quad + \underbrace{(\alpha\beta)_{11} X_{ijk4} + (\alpha\beta)_{12} X_{ijk5}}_{\text{AB interaction}} + \varepsilon_{ijk}, \end{aligned}$$

The data matrices then take the form:

$$\mathbf{Y} = \begin{bmatrix} Y_{111} \\ Y_{112} \\ Y_{113} \\ Y_{121} \\ Y_{122} \\ Y_{123} \\ Y_{124} \\ Y_{131} \\ Y_{132} \\ Y_{211} \\ Y_{212} \\ Y_{213} \\ Y_{221} \\ Y_{222} \\ Y_{231} \end{bmatrix} \qquad \mathbf{X}_f = \begin{array}{c} \begin{array}{cccccc} X_0 & X_1 & X_2 & X_3 & X_4 & X_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & 0 & -1 & 0 \\ 1 & -1 & 1 & 0 & -1 & 0 \\ 1 & -1 & 1 & 0 & -1 & 0 \\ 1 & -1 & 0 & 1 & 0 & -1 \\ 1 & -1 & 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & -1 & 1 & 1 \end{bmatrix} \end{array} \qquad \boldsymbol{\beta} = \begin{bmatrix} \mu_{..} \\ \alpha_1 \\ \beta_1 \\ \beta_2 \\ (\alpha\beta)_{11} \\ (\alpha\beta)_{12} \end{bmatrix}$$

Test for Interaction Effects

The test for interaction effects becomes, for the above regression model, a test of

$$\begin{aligned} H_o &: (\alpha\beta)_{11} = (\alpha\beta)_{12} = 0 \\ H_a &: (\alpha\beta)_{11} \neq 0 \text{ or } (\alpha\beta)_{12} \neq 0 \end{aligned}$$

The reduced model in this case is given by:

$$Y_{ijk} = \mu_{..} + \overbrace{\alpha_1 X_{ijk1}}^{\text{A main effect}} + \overbrace{\beta_1 X_{ijk2} + \beta_2 X_{ijk3}}^{\text{B main effect}} + \varepsilon_{ijk}.$$

That is, the reduced model contains variables X_0, X_1, X_2, and X_3. Therefore, the test for interaction effects is accomplished by using the model command:

```
-model 4 5 : 0 1 2 3
```

The F-statistics calculated for this model provide a test for presence of interaction between factors A and B. The complete batch command file is presented below.

Batch Command File for Example 5

```
3dRegAna  \
-rows 15 \
-cols 5 \
 -xydata   1   1   0   1   0  y111+tlrc \
 -xydata   1   1   0   1   0  y112+tlrc \
 -xydata   1   1   0   1   0  y113+tlrc \
 -xydata   1   0   1   0   1  y121+tlrc \
 -xydata   1   0   1   0   1  y122+tlrc \
 -xydata   1   0   1   0   1  y123+tlrc \
 -xydata   1   0   1   0   1  y124+tlrc \
 -xydata   1  -1  -1  -1  -1  y131+tlrc \
 -xydata   1  -1  -1  -1  -1  y132+tlrc \
 -xydata  -1   1   0  -1   0  y211+tlrc \
 -xydata  -1   1   0  -1   0  y212+tlrc \
 -xydata  -1   1   0  -1   0  y213+tlrc \
 -xydata  -1   0   1   0  -1  y221+tlrc \
 -xydata  -1   0   1   0  -1  y222+tlrc \
 -xydata  -1  -1  -1   1   1  y231+tlrc \
-diskspace \
-rmsmin 1.0 \
-fdisp 10 \
-model 4 5 :  0 1 2 3 \
-fcoef 4 ab11 \
-fcoef 5 ab12
```

■

The output files are ab11+tlrc.HEAD (and .BRIK) and ab12+tlrc.HEAD (and .BRIK). The set of voxels which light up when viewing these datasets using afni indicates which voxels have a significant interaction effect (at the operator specified significance level). Those voxels which do not show interaction effect can then be tested for presence of factor A and factor B main effects, as described below.

Test for Factor A Main Effect

The test for factor A main effect becomes, for the above regression model, a test of

$$
\begin{aligned}
H_o &: \alpha_1 = 0 \\
H_a &: \alpha_1 \neq 0
\end{aligned}
$$

The reduced model in this case is given by:

$$
Y_{ijk} = \mu_{..} + \underbrace{\beta_1 X_{ijk2} + \beta_2 X_{ijk3}}_{\text{B main effect}} + \underbrace{(\alpha\beta)_{11} X_{ijk4} + (\alpha\beta)_{12} X_{ijk5}}_{\text{AB interaction}} + \varepsilon_{ijk}
$$

That is, the reduced model contains variables X_0, X_2, X_3, X_4, and X_5; the test is to determine if variable X_1 should be added to the model. Therefore, the model command to test for factor A main effect is:

$$-\texttt{model}\ 1 : 0\ 2\ 3\ 4\ 5$$

The F-statistics calculated for this model provide a test for presence of factor A main effect. The batch command file is presented below. Note that only the -model and -fcoef commands are changed from the previous batch command file.

Batch Command File for Example 5

```
3dRegAna  \
      ⋮
same as above
      ⋮
-model 1 :  0 2 3 4 5 \
-fcoef 1 alpha1
```

■

The output file is alpha1+tlrc.HEAD (and .BRIK). The set of voxels which light up when viewing this datasets using afni indicates which voxels have a significant factor A (smoking vs. non-smoking) effect (at the operator specified significance level).

Test for Factor B Main Effect

The test for factor B main effect becomes, for the above regression model, a test of

$$\begin{aligned} H_o &: \beta_1 = \beta_2 = 0 \\ H_a &: \beta_1 \neq 0 \text{ or } \beta_2 \neq 0 \end{aligned}$$

The reduced model in this case is given by:

$$Y_{ijk} = \mu_{..} + \underbrace{\alpha_1 X_{ijk1}}_{\text{A main effect}} + \underbrace{(\alpha\beta)_{11} X_{ijk4} + (\alpha\beta)_{12} X_{ijk5}}_{\text{AB interaction}} + \varepsilon_{ijk}$$

That is, the reduced model contains variables X_0, X_1, X_4, and X_5; the test is to determine if variables X_2 and X_3 should be added to the model. Therefore, the model command to test for factor B main effect is:

$$-\texttt{model}\ 2\ 3 : 0\ 1\ 4\ 5$$

The F-statistics calculated for this model provide a test for presence of factor B main effect. The batch command file is presented below. Note that only the -model and -fcoef commands are changed from the previous batch command file.

Batch Command File for Example 5

```
3dRegAna \
    ⋮
same as above
    ⋮
-model 2 3 :  0 1 4 5 \
-fcoef 2 beta1 \
-fcoef 3 beta2
```

■

The output files are beta1+tlrc.HEAD (and .BRIK) and beta2+tlrc.HEAD (and .BRIK). The set of voxels which light up when viewing this datasets using afni indicates which voxels have a significant factor B (intelligence) effect (at the operator specified significance level).

1.7 References

1. N. Draper, H. Smith, *Applied Regression Analysis*, 2nd edition. New York, NY: John Wiley & Sons (1981).

2. J. Neter, W. Wasserman, M. H. Kutner, *Applied Linear Statistical Models*, 2nd edition. Homewood, Illinois: Irwin (1985).