MSG500/MVE190
Linear Models - Lectures 5 and 6

Rebecka Jörnsten
Mathematical Statistics
University of Gothenburg/Chalmers University of Technology

November 12, 2012

1 RECAP

- The noise level of the data is estimates as $\widehat{\sigma^2} = MSE/(n-p)$, where p is the number of model parameters (intercept and slopes).
- The 'usefullness' of the model can be assessed using the $R^2 = (SS_T - RSS)/RSS$, i.e. the reduction in variability in y as a results of using the regression model.
- The goodness-of-fit F-test makes this assessment more formal - is the association between y and x greater than one could expect to see by chance given the sample size and noise level?
- The F-statistic is computed as $F_{observed} = (SS_{reg}/(p-1))/MSE$, where $SS_{reg} = SS_T - RSS$.
- Under the null that $\beta_1 = 0$, $F_{observed}$ follows the F-distribution $F_{p-1,n-p}$.
- We reject the null if $F_{observed}$ exceeds a chosen critical value of $F_{p-1,n-}$, e.g. the $1-\alpha$ quantile for an α-level test.
- The slope parameters can also be tested: the sampling distribution for $\hat{\beta}_1$ is t_{n-p}. That is, $\frac{\hat{\beta}_1 - \beta_1}{SE(\hat{\beta}_1)} \sim t_{n-p}$ where $SE(\hat{\beta}_1) = \sqrt{\frac{\widehat{\sigma^2}}{\sum_i (x_i - \bar{x})^2}}$.

2 Multivariate regression models

Let $\mathbf{y} = \{y_1, \cdots, y_n\}'$ be a $n \times 1$ vector of dependent variable observations. Let $\boldsymbol{\beta} = \{\beta_0, \beta_1\}'$ be the 2×1 vector of regression parameters, and $\boldsymbol{\epsilon} = \{\epsilon_1, \cdots, \epsilon_n\}'$ be the $n \times 1$ vector of additive errors. We construct the so-called *design matrix* X (dimension $n \times 2$) as follows:

$$X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ . & . \\ 1 & x_n \end{pmatrix}$$

We can now write the simple linear regression model in two ways:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \ i = 1, \cdots, n, \tag{1}$$

or equivalently

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}. \tag{2}$$

The matrix formulation easily generalizes to multiple linear regression, involving predictor variables $x_1, \cdots, x_{p-1}$. We construct the $n \times p$ design matrix X:

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & . & x_{1,p-1} \\ 1 & x_{21} & x_{22} & . & x_{2,p-1} \\ . & . & . & . & . \\ 1 & x_{n1} & x_{n2} & . & x_{n,p-1} \end{pmatrix}$$

The multiple regression can be written as

$$y_i = \beta_0 + \beta_1 x_{i,1} + \cdots + \beta_{p-1} x_{i,p-1} + \epsilon_i,\ i = 1, \cdots, n, \tag{3}$$

or equivalently

$$\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{4}$$

where $\boldsymbol{\beta} = \{\beta_0, \beta_1, \cdots, \beta_{p-1}\}'$.

We use Least-Squares to fit a regression line to the data $\{\mathbf{x}_i, y_i\}_{i=1}^n$, where $\mathbf{x}_i = \{x_{i,1}, \cdots, x_{i,p-1}\}$. That is, we find the regression coefficient estimates $\widehat{\boldsymbol{\beta}}$ that minimizes the criterion

$$Q(\boldsymbol{\beta}) = (\mathbf{y} - X\boldsymbol{\beta})'(y - X\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - \mathbf{x}_i\boldsymbol{\beta})^2.$$

Taking derivatives with respect to $\boldsymbol{\beta}$, and setting these to 0, we obtain the *normal equations*:

$$\frac{dQ}{d\boldsymbol{\beta}} = -2X'(\mathbf{y} - X\boldsymbol{\beta}) = \mathbf{0} \Rightarrow$$

$$(X'X)\boldsymbol{\beta} = X'\mathbf{y} \tag{5}$$

To solve for $\boldsymbol{\beta}$ we apply the inverse of $X'X$ to both sides of equation5 and obtain:

$$\widehat{\boldsymbol{\beta}} = (X'X)^{-1}X'\mathbf{y} \tag{6}$$

.

It is not always possible to solve equation 5. When one might we run into trouble? Well, if the X matrix contains some near perfectly correlated columns (meaning some of the explanatory variables are highly correlated), the matrix $X'X$ may be singular and it won't be possible to construct its inverse.

Let's take a closer look at the two side of the normal equations: $X'X \simeq Cov(X)$, i.e. the left hand side of equation 5 captures the covariance structure among the independent variables. On the right hand side we have $X'y \simeq Cov(X, y) = (Cov(x_1, y), Cov(x_2, y), \ldots, Cov(xp-1, y))$, i.e. the vector of pairwise covariances between each independent variable x_j and the outcome y.

What happens when we solve for $\boldsymbol{\beta}$? If $X'X$ is a diagonal matrix, which would happen in all the explanatory variables are uncorrelated, then $\hat{\boldsymbol{\beta}} \simeq (Corr(x_1, y), Corr(x_2, y), \ldots, Corr(x_{p-1}, y))$, that is, each $\hat{beta}_j$ tells us how variable x_j is related to y. For this particular scenario we can say that the β_j's have a *direct* interpretation of how much each x_j affects y.

In most real life situations, $X'X$ will *not* be diagonal. The more correlated the x's are, the larger the off-diagonal elements of $X'X$ will be. When we then solve for $\boldsymbol{\beta}$ using equation 6 *all* x's will contribute to all β_j!

- When x's are highly correlated, $(X'X)^{-1}$ may not exist ($det(X'X) = 0$)
- If x's are highly correlated but the inverse exists, realize that this inverse is *numerically unstable*
- Numerical instability means that small changes to the data may lead to radical changes for the estimates $\hat{\beta}_j$ (magnitude *and* sign may change)
- Correlations among the x's means we lose the direct interpretation of β_j as the impact of x_j on y. β_j will now also depend on the correlation between other x_k and y.

Example:
Assume $x_2 = a + bx_1$ and $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$. There are an infinite number of regression models that fit these data equally well as long as the sum of the two coefficients are the same. Here's a small demonstration: I generate data from the same true model $y = 3 + 2x_1 - x_2 + \epsilon$. I explore two cases; (a) when x_1 and x_2 are independent and (b) when the x's are highly correlated (correlation .9). In Figure 1 I show the $\hat{\beta}_1$ and $\hat{\beta}_2$ obtained from 25 different data sets. The case a) estimates are shown as black circles and the case (b) as red asterisks. Note that the spread among the black circles is much less than

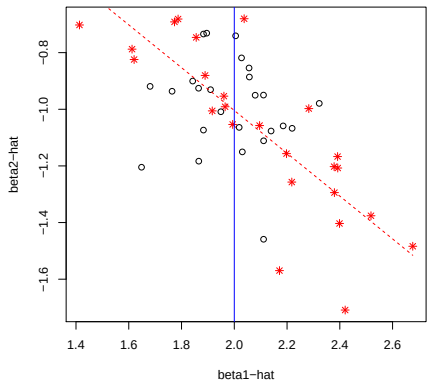

Figure 1: $\hat{\beta}_1$ and $\hat{\beta}_2$ estimated from 25 different data sets generated from the same true model, indicated with blue horizontal and vertical lines. Black circles correspond to estimates obtained when x-variables are uncorrelated and red asterisks estimates obtained when x's are highly correlated.

among the red asterisks, demonstrating that there the estimation uncertainty of estimating β's is much higher when x's are correlated. Note also that the $\hat{\beta}_1$ and $\hat{\beta}_2$ are highly correlated for case (b). The red dotted line is an estimate of this correlation - as you see it takes the form of $\hat{\beta}_1 + \hat{\beta}_2 =$ constant.

Call: lm(formula = y x1 + x2)
Residuals: Min 1Q Median 3Q Max -1.50136 -0.72254 0.08235 0.65359 2.35275
Coefficients: Estimate Std. Error t value Pr(>|t|) (Intercept) 2.8367 0.1404 20.204 < 2e-16 *** x1 2.2399 0.1361 16.457 < 2e-16 *** x2 -0.8371 0.1517 -5.519 1.43e-06 *** — Signif. codes: 0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1
Residual standard error: 0.9806 on 47 degrees of freedom Multiple R-squared: 0.863, Adjusted R-squared: 0.8572 F-statistic: 148.1 on 2 and 47 DF, p-value: < 2.2e-16 Call: lm(formula = y x1 +

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.8367	0.1404	20.20	0.0000
x1	2.2399	0.1361	16.46	0.0000
x2	-0.8371	0.1517	-5.52	0.0000

Table 1: Regression summary - uncorrelated x's

x2)
Residuals: Min 1Q Median 3Q Max -2.25568 -0.66198 -0.08142 0.67605 2.16192
Coefficients: Estimate Std. Error t value Pr(>|t|) (Intercept) 3.0389 0.1504 20.210 < 2e-16 *** x1 2.6708 0.3624 7.369 2.26e-09 *** x2 -1.4384 0.3263 -4.408 6.02e-05 *** — Signif. codes: 0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1
Residual standard error: 1.059 on 47 degrees of freedom Multiple R-squared: 0.6143, Adjusted R-squared: 0.5979 F-statistic: 37.44 on 2 and 47 DF, p-value: 1.887e-10

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.0389	0.1504	20.21	0.0000
x1	2.6708	0.3624	7.37	0.0000
x2	-1.4384	0.3263	-4.41	0.0001

Table 2: Regression summary - correlated x's

In Tables 1 and 2 I provide the regression model summaries for the two cases (a) uncorrelated x's and (b) correlated x's. As you see, the standard errors for case (b) are higher, indicating that estimation

is more difficult when the x's are correlated.s

3 Properties

3.1 The Hat-matrix

Note, the fitted values can be written as

$$\widehat{\mathbf{y}} = X\widehat{\boldsymbol{\beta}} = X(X'X)^{-1}X'\mathbf{y},$$

where we denote the $n \times n$ matrix $X(X'X)^{-1}X'$ by H, the "Hat-matrix". The matrix H is an idempotent projection matrix:

$$HH = H \Rightarrow$$

$$X'e = X'(\mathbf{y} - \widehat{\mathbf{y}}) = X'(\mathbf{y} - H\mathbf{y}) = X'(I - H)\mathbf{y} = X'\mathbf{y} - (X'X)(X'X)^{-1}X'\mathbf{y} = 0,$$

i.e. the residuals are orthogonal to all predictor variables. Note how the residuals could be written as

$$e = (I - H)\mathbf{y}.$$

In addition,

$$e'\widehat{\mathbf{y}} = ((I - H)\mathbf{y})'H\mathbf{y} = \mathbf{y}'(I - H)H\mathbf{y} = 0,$$

i.e. fitted values are orthogonal to the residuals.

We say that H is a *projection matrix*. It takes the data $\mathbf{y}$ and projects it onto the plane spanned by X such that we obtain the fitted values $\hat{\mathbf{y}} = H\mathbf{y}$.

If we write this on the long form we have:

$$\begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ . \\ \hat{y}_n \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} & . & h_{1n} \\ h_{21} & h_{22} & h_{23} & . & h_{2n} \\ . & . & . & . & . \\ h_{n1} & h_{n2} & h_{n3} & . & h_{nn} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ . \\ y_n \end{pmatrix}$$

The diagonal elements of the H-matrix, $h_{ii}, i = 1, \ldots, n$ constitute the leverage. In multivariate regression the leverage takes the form

$$h_{ii} = \mathbf{x}_i(X'X)^{-1}\mathbf{x}_i', \text{ where } \mathbf{x}_i \text{ is the i-th row in } X.$$

For which observations do we get a large leverage? When $\mathbf{x}_i$ (the vector of all x-variable values for observation i) is extreme compared $\bar{\mathbf{x}}$ (the vector of the means of each x-variable). Here we have to be a little careful. What is "extreme" in a multivariate setting? In the simple linear regression model we only had to compare x_{i1} (the i-th value of variable x_1) with its mean $\bar{x}_1$. Now we also have to pay attention to the structure *among* the different x-variables as captured in $X'X$. The leverage will be high for observations i that deviate from the structure $X'X$. Here's an illustration: Following the same set up as in Figure 1 I generate a data set with uncorrelated x's and one with correlated x's. I plot each x against y and both x-variables against each-other in Figure 2. The observations with maximum leverage are marked with red asterisks in each plot. Note that for uncorrelated x's, maximum leverage does correspond to x-values far from the mean of x. For correlated x's extreme leverage is obtained in 'surprising' locations, like when the x-values deviate from the correlation pattern of the bulk of the data.

3.2 Mean and Variance

$$E[\widehat{\boldsymbol{\beta}}] = E[(X'X)^{-1}X'\mathbf{y}] = (X'X)^{-1}X'X\boldsymbol{\beta} = \boldsymbol{\beta}.$$

I.e., the least-squares estimates are unbiased.

$$V[\widehat{\boldsymbol{\beta}}] = V[(X'X)^{-1}X'\mathbf{y}] = (X'X)^{-1}X'V(\mathbf{y})X(X'X)^{-1},$$

since X is not random. $V(\mathbf{y}) = \sigma^2 I$, since the errors are uncorrelated (and therefore so are the $y's$). It follows that

$$V[\widehat{\boldsymbol{\beta}}] = \sigma^2(X'X)^{-1}.$$

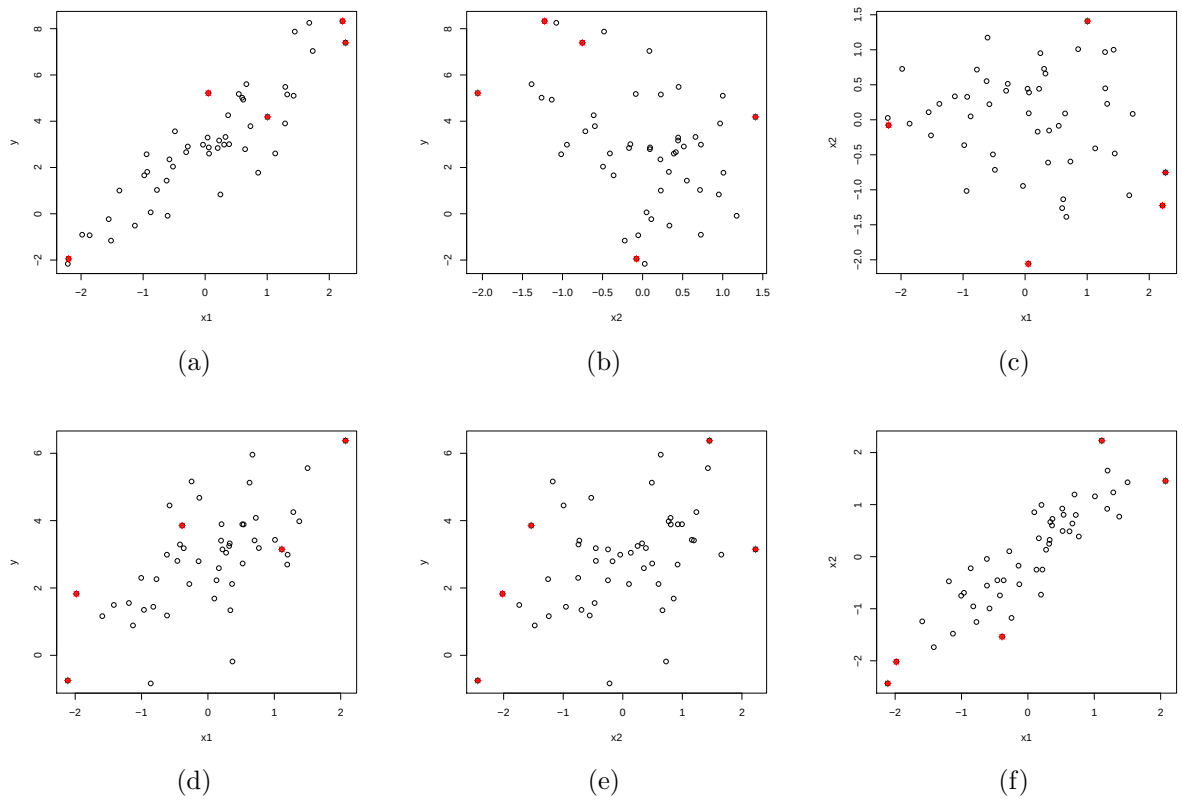

Figure 2: Scatter plots of y vs x_1 (a) and (d), y vs x_2 (b) and (e), x_2 vs x_1 (c) and (f). Panels (a)-(c), uncorrelated x's, Panels (d)-(f) correlated x's. The observation with maximum leverage are marked with red asterisks.

3.3 Interpretation

Note, $X'X \sim Cov(X)$, where the diagonal of the $p \times p$ matrix $X'X$ is the variances of the individual predictor variables (assuming x's are centered). Now, what would happen in some of the predictor variables are closely related (e.g. weight and height). If individual x's are correlated (close to linearly dependent), the $X'X$ matrix is near-singular. To solve for $\boldsymbol{\beta}$ we need to apply the inverse of $X'X$ to both sides of equation (5). If $X'X$ is near-singular this is a highly unstable operation.

What does this mean? Well, consider the regression model in equation (3). If x_1 and x_2 are closely related predictor variables, then we have no way of distinguishing between them in the regression model. Let's take the extreme example $x_1 = x_2$. If this is the case, then any combination of β_1, β_2 where $\beta_1 + \beta_2$ is constant is an equally good regression model. This extreme case is an example of an "unidentifiable" model - there is no unique best model.

The effect of this is seen in the variance of the least squares estimates. If $X'X$ is near-singular, the determinant is close to 0 and the terms in the inverse $(X'X)^{-1}$ can get very large. Therefore, the variance of the estimates $\widehat{\boldsymbol{\beta}}$ is high whenever predictor variables are correlated. For correlated predictor variables x_1, x_2 you would expect $\hat{\beta}_1$ and $\hat{\beta}_2$ to have high variance and be *negatively* correlated (since their sum $\beta_1 + \beta_2 \simeq$ constant). This is exactly what was demonstrated in Figure 1 and in Tables 1 and 2.

Let's talk a bit more about $V(\hat{\boldsymbol{\beta}}) = \sigma^2(X'X)^{-1}$. We can identify the three sources of errors in this expression:

1. The higher the noise level σ^2, the higher the estimation variance for $\hat{\boldsymbol{\beta}}$.
2. As before, $nCov(X) \sim X'X$, and so the larger the sample size, the smaller the estimation variance.

3. The more spread in any x ($V(\mathbf{x}_j)$) the smaller the estimation uncertainty - BUT not only for $\hat{\beta}_j$ if the x's are correlated. In fact, the more dependency we have between the x's the larger $V(\hat{\boldsymbol{\beta}})$ (see Tables 1 and2).

The more correlation we have between x's, the more correlated their estimates will be, and the higher the estimation variance.

3.4 Some more properties

Since

$$\hat{\mathbf{y}} = H\mathbf{y} \rightarrow E[\hat{\mathbf{y}}] = E[\mathbf{y}],\ V[\hat{\mathbf{y}}] = HV[\mathbf{y}]H = \sigma^2 H.$$

The fitted values have marginal variance given by the leverage h_{ii} (diagonal elements of H).

Similarly, for the residuals,

$$e = (I - H)\mathbf{y} \rightarrow E[e] = 0,\ V[e] = \sigma^2(I - H).$$

Note that nothing has really changed from the simple linear model case as long as we formulate everything in terms of the hat-matrix H.

4 Basic Inference

If we assume $\epsilon \sim N(0, \sigma^2)$, the derivation of the t-test and F-test in the multiple regression case follow from the same line of thought as the simple case.

We thus have:

- The test statistics $F_{observed} = [(SS_T - RSS)/(p-1)]/[RSS/(n-p)]$, where SS_T is the total sum of squares $\sum_i (y_i - \bar{y})^2$, and RSS is the error sum of squares in the p-parameter multiple regression fit: $\sum_i (y_i - \hat{y}_i)^2$.
- Under the null, $\beta_j = 0$ for all $j = 1, \cdots, p-1$, both $SS_T/(n-1)$ and $RSS/(n-p)$ as well as $SS_{reg}/(p-1) = (SS_T - RSS)/(p-1)$ provide estimates for the error variance σ^2.
- Under the null, we thus expect $F_{observed}$ to be close to 1. In fact, under the null, $F_{observed}$ should come from an F-distribution with $p-1$ and $n-p$ degrees of freedom.
- We compare $F_{observed}$ to the $1-\alpha$ quantiles of the $F_{p-1,n-p}$ distribution. If $F_{observed}$ exceeds the $1-\alpha$ quantile, we reject the null at the α level, and conclude that *at least 1* of $\beta_1, \cdots, \beta_{p-1}$ is different from 0.

Similarly, for inference on a single regression coefficient:

- We define the test statistic $t_{observed} = \hat{\beta}_j / SE(\hat{\beta}_j)$, where $SE(\hat{\beta}_j) = \sqrt{\hat{\sigma}^2 diag((X'X)^{-1})_j}$ is the *standard error* of the estimate $\hat{\beta}_j$ (Remember, $V(\widehat{\boldsymbol{\beta}}) = \sigma^2(X'X)^{-1}$).
- If the true $\beta_j = 0$, $t_{observed}$ should come from a t-distribution with $n-p$ degrees of freedom.
- If the true $\beta_j \neq 0$, the test statistic will be inflated (positive or negative).
- We reject the null hypothesis if $|t_{observed}|$ exceeds the $1-\alpha/2$ quantile of the t_{n-p}-distribution.

Caveat: if x's are correlated, then so are their estimates. In that case, testing each regression coefficient separately with a t-test can be misleading. Mathematically, you can't really tell them apart.

4.1 The R-squared

The R-squared is computed just as before:

$$R^2 = \frac{SS_{reg}}{SS_T} = \frac{SS_T - RSS}{SS_T} = 1 - \frac{RSS}{SS_T}.$$

However, when we include a large number of explanatory variables (x-variables) in the model, the R-squared can get deceptively inflated. In fact, if you include close to n variables in your model you can achieve an R-squared close to 1 even if the x's are unrelated to y! **Try this out yourself.**

The *adjusted R-squared* takes the number of model parameters into account as follows:

$$R^2_{adj} = 1 - \frac{RSS}{SS_T}\frac{n-1}{n-p} = 1 - \frac{MSE}{MST}$$

4.2 The F-test for subset selection

The goodness-of-fit F-test investigates the null hypothesis that *all* slope parameters are equal to 0 against the alternative hypothesis that *at least one* slope parameter is different from 0. This alternative is rather vague. The F-test can be used to compare more precise subtypes of models.

Here is an example where I want to test if the first $\beta_j = 0, j = 1, \ldots, k$

- Null hypothesis: $\beta_j = 0, j = 1, \ldots, k$.
- Alternative: At least one of $\beta_j, j = 1, \ldots, k$ is not 0
- I don't specify for $\beta_j, j = k+1, \ldots, p-1$

I name the model under the alternative *the complex model* and the model under the null *the simple model*. I now fit each of the models to the data (in the case of the null by simply not including variables $x_1, \ldots, x_k$ in the fitting). I compute the residual sum of squares for each of the models: $RSS_{complex}$ and RSS_{simple}. The F-statistic is computed as

$$F_{observed} = \frac{\big((RSS_{simple} - RSS_{complex})/(df_{simple} - df_{complex})\big)}{RSS_{complex}/(df_{complex})},$$

where $df_{complex} = n-p$ and $df_{simple} = n-(p-k)$, i.e. the degrees of freedom of the errors for each model.

Now, if the null is true $F_{observed}$ is distributed as $F_{k,n-p}$, i.e. an F-distribution with the degrees of freedom given by the difference in the number of parameters between the simple and complex models and the error degrees of freedom of the complex models. The rationale for this F-test follows the derivation of the F-test for goodness-of-fit. If the null is true, the RSS of the simple model can provide an estimate of the error variance σ^2 just as well as the complex model, etc.

You can use this kind of F-test to compare any *nested* models, i.e. where the simple model is the complex model with some of its parameters set to 0.

5 Introduction to model selection

You can use both t-tests and F-tests to decide on a model, but you need to be a bit careful!

5.1 Using the t-test for selection

You could use the t-test, or just the regression summary (see e.g. Table 1) to select which variables to keep and which to eliminate from the model. Looking at the regression summary you can eliminate all variables x_j whose estimates $\hat{\beta}_j$ are not significant.

What's the problem with this approach?
If the x's are correlated we know from before (see Figure 1) that the estimates are correlated. It will be

an almost random decision if for two highly correlated variables the first, the other, or both are kept in the model. Since estimation variance increases for correlated variables you run the risk of dropping a variable even if it has a real relationship with y.

In addition, this is an example of *multiple testing*. We perform a t-test on each slope coefficient. If we use an α-level test each test has a chance α of leading to a false rejection, i.e. declaring the slope parameter significant although no real relationship exists between this variable and y. If we do this 10 estimates, the probability that we get at least one false rejection is no longer α but $(1-(1-\alpha)^10$ (compare 5% to 40% for the case of $\alpha = .05$).

5.2 F-test and subset selection

The F-test is a way of testing several slope estimates at the same time. However, which subset (complex and simple models) should we compare?
If there are p parameters, $p-1$ slope parameters for each of the x-variables and one intercept, there are 2^{p-1} model combinations we can consider! In Table 3 you can see how quickly the number of subset

number of variables	1	2	3	.	10	20	30
number of models	2	4	8	.	1024	1e+6	1e+9

Table 3: The number of subset models as a function of the number of x-variables

models grows as a function of the number of variables in the model. Most software packages can handle *all-subset* selection only up until about 30 variables.

What are some alternatives then? We could forsake comparing all subsets and perform directed or greedy searches. A common approach is the so-called *Backward search* which is outlined here:

Backward Model Selection:

0. * Initialize by fitting the full model with all p parameters - obtain the $RSS_{complex}$. Set the current number of paramters $p_c = p$. Include the variables $x_1, x_2, \ldots, x_{p-1}$ in the "active" set of variables A.

1. * Reducing the model.

- Examine the fit of each of the $p_c - 1$ subset models corresponding to dropping one of the variables in set A from the model. Denote the corresponding error sum of squares by RSS_k for variables $x_k, k \in A$.
- Identify the variable $x_+, * \in A$ with the minimum $RSS_+ = \min_{k \in A} RSS_k$

2. * Compute the $F_{obs} = \frac{(RSS_+ - RSS_{complex})}{RSS_{complex}/(n-p_c)}$

3. * If $F_{obs} < F_{1,n-p_c}(1-\alpha)$

- don't reject the null hypothesis that $\beta_+ = 0$
- drop x_+ from the model and update set $A = A \setminus x_+$
- set $RSS_{complex} = RSS_+$ and $p_c = p_c - 1$
- Go to [**1.**]

* Else, if $F_{obs} > F_{1,n-p_c}(1-\alpha)$, reject the null hypothesis that $\beta_+ = 0$ and STOP

You can of course perform a *forward* search, considering the addition of one variable in each step and stopping when you cannot reject the hypothesis that the most recently added variable has slope 0.
Here are some cautionary statements about stepwise or greedy model selection:

- Since it is greedy you are not guaranteed to find the best subset model. There are variants of stepwise model selection where you add a random element to the mix which can help (moving a few steps forward, a few steps backward, which allows for erroneous drops or additions of variables to be reversed).
- Greedy searchers can lead to models that are difficult to interpret - variables that are correlated are competing to be in the model and human knowledge may be able to tell sensible models apart where the statistics cannot.

6 Demo 5

We will work with the South-African heart disease data (see e.g. the book "The Elements of Statistical Learning", by Hastie, Friedman and Tibshirani).

```
> SA <- data.frame(read.table("SA.dat", sep = "\t", header = T))
> print(dim(SA))

[1] 312  12

> print(names(SA))

 [1] "age"       "sbp"       "adiposity" "obesity"   "typea"     "alcohol"
 [7] "alcind"    "tobacco"   "tobind"    "chd"       "famhist"   "ldl"
```

There is data on 311 male individuals. The variables in the data set include; age of patient (age), systolic bloodpressure (sbp), fat in adipose tissue (beneath the skin, around organs) (adiposity), body mass index (bmi) (obesity), typa A behaviour (aggressive personality) (typea), how much alcohol units consumed per week (alcohol), an indicator if drink alcohol at al (alcind), cumulative tobacco consumption in kg (tobacco), and indicator if patient is /have been a smoker (tobind), an indicator whether patient is diagnosed with heart disease (chd), an indicator if the patient has family member with heart disease (famhist), and finally the cholesterol level of the patient (ldl).
Now, I will use this data set to model cholesterol, though a more natural way of thinking about the data is probably to treat the heart disease as the outcome. We will return to this data set later in the class and do precisely that.

```
> par(mfrow = c(2, 2))
> plot(SA$obesity, SA$ldl, main = "LDL on obesity")
> plot(SA$sbp, SA$ldl, main = "LDL on blood pressure")
> plot(SA$tobacco, SA$ldl, main = "LDL on tobacco usage")
> plot(SA$alcohol, SA$ldl, main = "LDL on alcohol consumption")
> p <- locator()

> par(mfrow = c(2, 2))
> boxplot(SA$ldl ~ SA$chd, main = "LDL on coronary heart disease status")
> boxplot(SA$ldl ~ as.factor(SA$famhist), main = "LDL on family history of same")
> plot(SA$age, SA$ldl, main = "LDL on age")
> plot(SA$typea, SA$ldl, main = "LDL on type A behaviour")
> p <- locator()
```

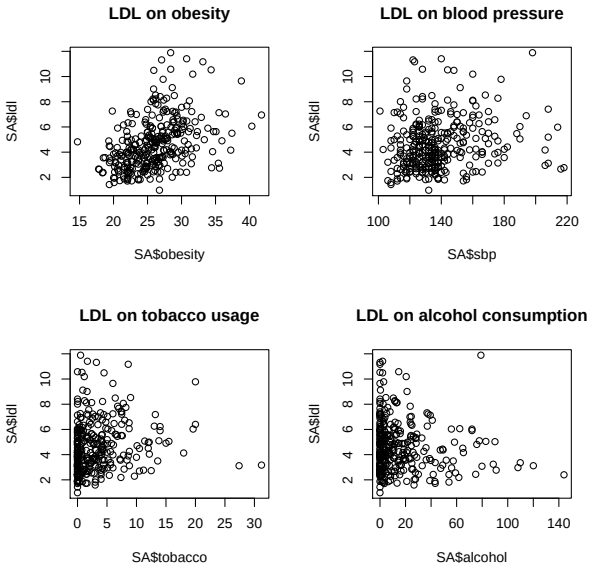

Figure 3: Scatter plots. TL: ldl vs obesity, TR: ldl vs blood pressure. LL: ldl vs tobacco. LR: ldl vs alcohol

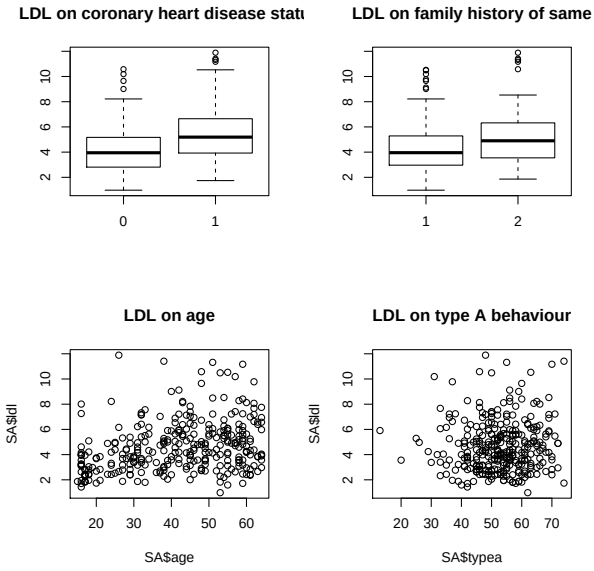

Figure 4: Scatter plots. TL: ldl vs heart disease, TR: ldl vs family history of heart disease. LL: ldl vs age. LR: ldl vs type A

In Figures 3 and4 we see some indication that the error scatter is non-constant, it seems to be higher with higher levels of cholesterol (ldl). I try out a couple of data transformations to deal with the problem (see Figure 5).
Try some other transformations yourself.

```
> par(mfrow = c(2, 2))
> plot(log(SA$obesity), log(SA$ldl), main = "log(LDL) on log(obesity)")
> plot(log(SA$age), log(SA$ldl), main = "log(LDL) on log(age)")
> plot(SA$adi, log(SA$ldl), main = "log(LDL) on adiposity")
> plot(SA$sbp, log(SA$ldl), main = "log(LDL) on blood pressure")
> p <- locator()
```

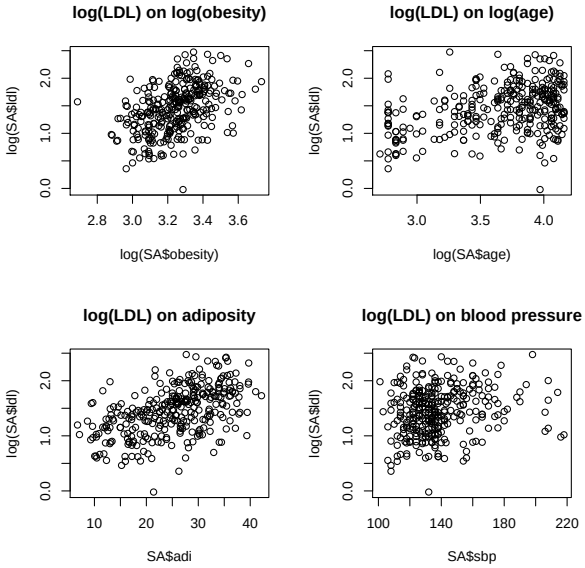

Figure 5: Trying data transformations. TL: log(ldl) vs heart log(obesity), TR: log(ldl) vs log(age) LL: log(ldl) vs adiposity. LR: log(ldl) vs blood pressure

```
> mm1 <- lm(log(ldl) ~ log(age) + log(obesity) + as.factor(chd) +
+     as.factor(famhist) + as.factor(tobind) + as.factor(alcind) +
+     tobacco + alcohol + adiposity + typea + sbp, data = SA)
> print(summary(mm1))

Call:
lm(formula = log(ldl) ~ log(age) + log(obesity) + as.factor(chd) +
    as.factor(famhist) + as.factor(tobind) + as.factor(alcind) +
    tobacco + alcohol + adiposity + typea + sbp, data = SA)

Residuals:
     Min       1Q   Median       3Q      Max
-1.23876 -0.21076  0.03156  0.23059  0.98068

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        -0.0956208  0.6728934  -0.142 0.887093
log(age)           -0.0430960  0.0733132  -0.588 0.557086
```

```
log(obesity)          0.2746311  0.2124959   1.292 0.197211
as.factor(chd)1       0.1560178  0.0479134   3.256 0.001258 **
as.factor(famhist)2   0.0733438  0.0426357   1.720 0.086420 .
as.factor(tobind)1    0.1572254  0.0537512   2.925 0.003707 **
as.factor(alcind)1    0.0178784  0.0509036   0.351 0.725669
tobacco              -0.0008815  0.0053774  -0.164 0.869906
alcohol              -0.0035654  0.0009710  -3.672 0.000285 ***
adiposity             0.0220377  0.0050461   4.367 1.73e-05 ***
typea                 0.0019406  0.0020552   0.944 0.345822
sbp                  -0.0000682  0.0010241  -0.067 0.946950
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.3532 on 300 degrees of freedom
Multiple R-squared: 0.3602,        Adjusted R-squared: 0.3368
F-statistic: 15.36 on 11 and 300 DF,  p-value: < 2.2e-16
```

```
> par(mfrow = c(2, 2))
> plot(mm1)
> p <- locator()
```

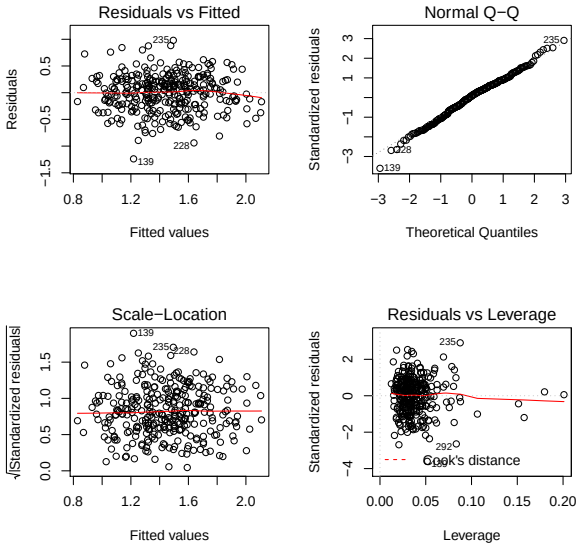

Figure 6: Diagnostic plots

The diagnostic plots in Figure 6 include:

- Top left: a residual vs fitted value plot - look for an even spread around the horizontal axis
- Top right: a QQplot - checking to see if the residuals follows a near normal distribution
- Bottom left: absolute values of residuals vs fitted values - check to see that the variance does not vary with the fitted value
- Bottom right: residuals vs leverage - look for extremes (R will mark some for you)

Let's try some other diagnostic plots.

```
> plot(cooksd, main = "Cook's Distance", type = "h")
> abline(h = qf(0.95, 1, mm1$df), lty = 2)
> if (max(id$ind) != -Inf) {
+     text(id$ind, cooksd[id$ind], id$ind, pos = id$pos)
+ }

> plot(lm1$hat, main = "Leverage")
> abline(h = 3 * length(mm1$coef)/dim(SA)[1])
> if (max(idlev$ind) != -Inf) {
+     text(idlev$ind, lm1$hat[idlev$ind], idlev$ind, pos = idlev$pos)
+ }
```

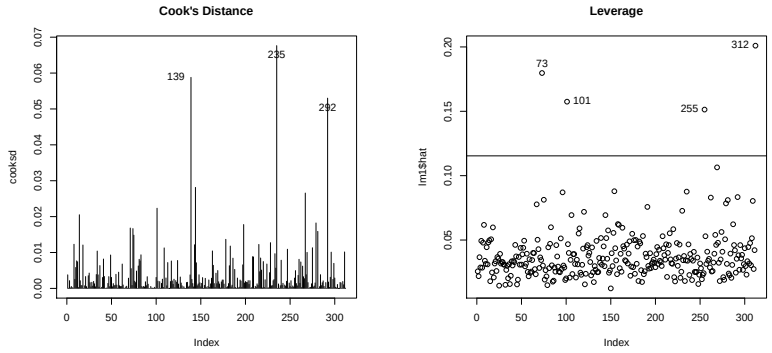

Figure 7: Diagnostic plots: Cook's Distance and Leverage plots

```
> plot(lm1$coeff[, 4], main = "change in slope 4")
> if (max(id4$ind) != -Inf) {
+     text(id4$ind, lm1$coeff[id4$ind, 4], id4$ind, pos = id4$pos)
+ }

> plot(lm1$coeff[, 6], main = "change in slope 6")
> if (max(id6$ind) != -Inf) {
+     text(id6$ind, lm1$coeff[id6$ind, 6], id6$ind, pos = id6$pos)
+ }

> plot(lm1$coeff[, 9], main = "change in slope 9")
> if (max(id9$ind) != -Inf) {
+     text(id9$ind, lm1$coeff[id9$ind, 9], id9$ind, pos = id9$pos)
+ }

> plot(lm1$coeff[, 10], main = "change in slope 10")
> if (max(id10$ind) != -Inf) {
+     text(id10$ind, lm1$coeff[id10$ind, 10], id10$ind, pos = id10$pos)
+ }
```

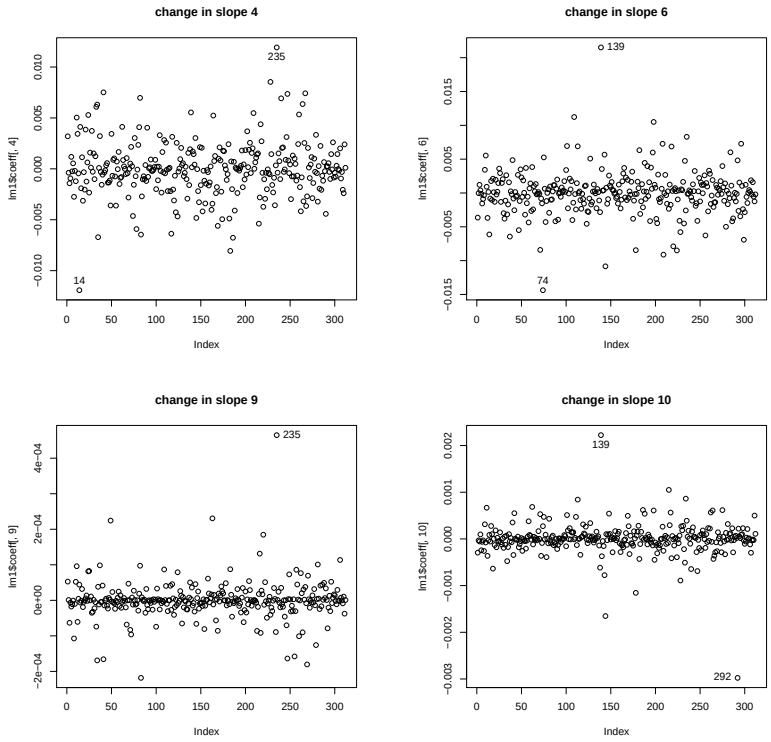

Figure 8: Change in slope in coefficients 4,6, 9 and 10

```
> indvec <- c(id$ind, idlev$ind, id4$ind, id6$ind, id9$ind, id10$ind)
> print(table(indvec))

indvec
 14  73  74 101 139 235 255 292 312
  1   1   1   1   3   3   1   2   1

> indout <- unique(sort(indvec))[sort.list(table(indvec), decreasing = T)[1]]
```

I identify the most commonly present outliers as observations 139. I rerun the analysis without this observation.

```
> mm1b <- lm(log(ldl) ~ log(age) + log(obesity) + as.factor(chd) +
+     as.factor(famhist) + as.factor(tobind) + as.factor(alcind) +
+     tobacco + alcohol + adiposity + typea + sbp, data = SA, subset = -indout)
> print(summary(mm1b))

Call:
lm(formula = log(ldl) ~ log(age) + log(obesity) + as.factor(chd) +
    as.factor(famhist) + as.factor(tobind) + as.factor(alcind) +
    tobacco + alcohol + adiposity + typea + sbp, data = SA, subset = -indout)

Residuals:
     Min       1Q   Median       3Q      Max
-0.93056 -0.22058  0.03154  0.22218  1.01322

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        -0.3935346  0.6642453  -0.592 0.553995
log(age)           -0.0019872  0.0726955  -0.027 0.978210
```

```
log(obesity)          0.3446144  0.2090664   1.648 0.100331
as.factor(chd)1       0.1504759  0.0469685   3.204 0.001503 **
as.factor(famhist)2   0.0638829  0.0418526   1.526 0.127973
as.factor(tobind)1    0.1357134  0.0529882   2.561 0.010922 *
as.factor(alcind)1    0.0029381  0.0500394   0.059 0.953218
tobacco              -0.0013189  0.0052700  -0.250 0.802546
alcohol              -0.0035687  0.0009514  -3.751 0.000211 ***
adiposity             0.0198151  0.0049808   3.978 8.72e-05 ***
typea                 0.0023561  0.0020168   1.168 0.243647
sbp                  -0.0001148  0.0010034  -0.114 0.908999
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

Residual standard error: 0.346 on 299 degrees of freedom
Multiple R-squared: 0.3645,        Adjusted R-squared: 0.3411
F-statistic: 15.59 on 11 and 299 DF,  p-value: < 2.2e-16
```

```
> par(mfrow = c(2, 2))
> plot(mm1b)
> p <- locator()
```

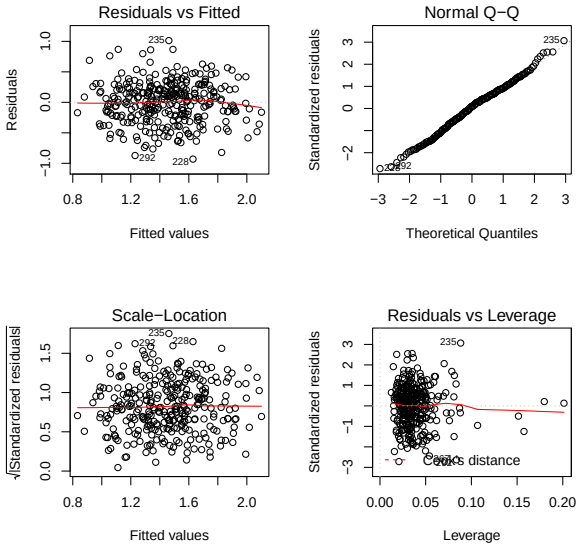

Figure 9: Diagnostics after removing observation 139

In the regression summary and diagnostic figures (Figure 9), we see that the fit is improved after removing the outlier.

```
> SA2 <- SA
> SA2$obesity <- log(SA$obesity)
> SA2$age <- log(SA$age)
> SA2$ldl <- log(SA$ldl)
> distmat <- 1 - cor(SA2[, -12])
> library(gplots)
> hh <- heatmap.2(as.matrix(distmat), col = redgreen(75), cexRow = 0.5,
+     key = TRUE, symkey = FALSE, density.info = "none", trace = "none")
```

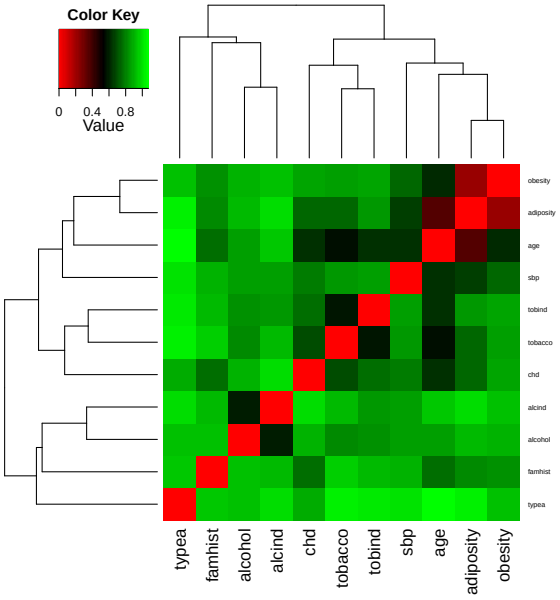

Figure 10: Clustering of explanatory variables.

In Figure 10 I depict a clustering of the explanatory variables. Which variables cluster together? What does that mean in terms of interpretational value of the regression model?

```
> step(mm1b, directions = "backward")

Start:  AIC=-648.35
log(ldl) ~ log(age) + log(obesity) + as.factor(chd) + as.factor(famhist) +
    as.factor(tobind) + as.factor(alcind) + tobacco + alcohol +
    adiposity + typea + sbp

                     Df Sum of Sq    RSS     AIC
- log(age)            1   0.00009 35.798 -650.35
- as.factor(alcind)   1   0.00041 35.798 -650.35
- sbp                 1   0.00157 35.800 -650.34
- tobacco             1   0.00750 35.806 -650.29
- typea               1   0.16340 35.961 -648.93
<none>                            35.798 -648.35
- as.factor(famhist)  1   0.27894 36.077 -647.94
- log(obesity)        1   0.32530 36.123 -647.54
- as.factor(tobind)   1   0.78537 36.583 -643.60
- as.factor(chd)      1   1.22888 37.027 -639.85
- alcohol             1   1.68455 37.483 -636.05
- adiposity           1   1.89486 37.693 -634.31

Step:  AIC=-650.35
log(ldl) ~ log(obesity) + as.factor(chd) + as.factor(famhist) +
    as.factor(tobind) + as.factor(alcind) + tobacco + alcohol +
    adiposity + typea + sbp

                     Df Sum of Sq    RSS     AIC
- as.factor(alcind)   1   0.00042 35.799 -652.35
- sbp                 1   0.00176 35.800 -652.34
- tobacco             1   0.00831 35.806 -652.28
- typea               1   0.16487 35.963 -650.92
<none>                            35.798 -650.35
- as.factor(famhist)  1   0.28483 36.083 -649.89
```

```
- log(obesity)        1   0.34268 36.141 -649.39
- as.factor(tobind)   1   0.82401 36.622 -645.27
- as.factor(chd)      1   1.24607 37.044 -641.71
- alcohol             1   1.68852 37.487 -638.02
- adiposity           1   2.51981 38.318 -631.20

Step:  AIC=-652.35
log(ldl) ~ log(obesity) + as.factor(chd) + as.factor(famhist) +
    as.factor(tobind) + tobacco + alcohol + adiposity + typea +
    sbp

                     Df Sum of Sq    RSS     AIC
- sbp                 1   0.00160 35.800 -654.33
- tobacco             1   0.00834 35.807 -654.27
- typea               1   0.16461 35.963 -652.92
<none>                            35.799 -652.35
- as.factor(famhist)  1   0.28675 36.085 -651.87
- log(obesity)        1   0.34694 36.145 -651.35
- as.factor(tobind)   1   0.83388 36.632 -647.19
- as.factor(chd)      1   1.24729 37.046 -643.70
- alcohol             1   1.94543 37.744 -637.89
- adiposity           1   2.53784 38.336 -633.05

Step:  AIC=-654.33
log(ldl) ~ log(obesity) + as.factor(chd) + as.factor(famhist) +
    as.factor(tobind) + tobacco + alcohol + adiposity + typea

                     Df Sum of Sq    RSS     AIC
- tobacco             1   0.00840 35.809 -656.26
- typea               1   0.16552 35.966 -654.90
<none>                            35.800 -654.33
- as.factor(famhist)  1   0.28656 36.087 -653.85
- log(obesity)        1   0.34661 36.147 -653.34
- as.factor(tobind)   1   0.83228 36.632 -649.19
- as.factor(chd)      1   1.25398 37.054 -645.63
- alcohol             1   1.97882 37.779 -639.60
- adiposity           1   2.59393 38.394 -634.58

Step:  AIC=-656.26
log(ldl) ~ log(obesity) + as.factor(chd) + as.factor(famhist) +
    as.factor(tobind) + alcohol + adiposity + typea

                     Df Sum of Sq    RSS     AIC
- typea               1   0.16902 35.978 -656.80
<none>                            35.809 -656.26
- as.factor(famhist)  1   0.29475 36.103 -655.71
- log(obesity)        1   0.35973 36.168 -655.15
- as.factor(tobind)   1   0.88219 36.691 -650.69
- as.factor(chd)      1   1.26426 37.073 -647.47
- alcohol             1   2.04131 37.850 -641.02
- adiposity           1   2.62351 38.432 -636.27

Step:  AIC=-656.8
log(ldl) ~ log(obesity) + as.factor(chd) + as.factor(famhist) +
    as.factor(tobind) + alcohol + adiposity

                     Df Sum of Sq    RSS     AIC
<none>                            35.978 -656.80
```

```
- as.factor(famhist)  1   0.30162 36.279 -656.20
- log(obesity)        1   0.44901 36.427 -654.94
- as.factor(tobind)   1   0.84488 36.822 -651.58
- as.factor(chd)      1   1.42731 37.405 -646.70
- alcohol             1   1.97752 37.955 -642.15
- adiposity           1   2.48021 38.458 -638.06

Call:
lm(formula = log(ldl) ~ log(obesity) + as.factor(chd) + as.factor(famhist) +
    as.factor(tobind) + alcohol + adiposity, data = SA, subset = -indout)

Coefficients:
        (Intercept)         log(obesity)      as.factor(chd)1
          -0.403954             0.388258             0.154690
as.factor(famhist)2   as.factor(tobind)1              alcohol
           0.065295             0.127694            -0.003525
          adiposity
           0.018669
```