Addenda to the Material Covered in the ISLR Book

March 3, 2018

1 A Recap

Let us recall that a continuous random variable X is specified by the cdf

$$F(x) = P(X \leq x)$$

a non-decreasing function with values in $[0, 1]$. For absolutely continuous function (which is what we will consider throughout), we can express the cdf in terms of a non-negative integrable function, the probability density function $f_X(x)$ (if there is no ambiguity we can simply write $f(x)$)

$$F(x) = \int_{-\infty}^{x} f(x)\, dx$$

Also, we define the expected value of X as

$$E(X) = \mu_X = \int_{-\infty}^{\infty} x\, f(x)\, dx$$

Similarly, if we have the function of a random variable $g(X)$ is also a random variable. Its expected value can be computed using the pdf of X

$$E(g(X)) = \int_{-\infty}^{\infty} g(x)\, f(x)\, dx$$

Recall that the variance is

$$Var(X) = E(X^2) - E(X)^2$$

and that the expected values is linear

$$E(\sum_i a_i X_i) = \sum_i a_i E(X_i)$$

Finally, if we have two random variables X, Y, their joint pdf $f_{XY}(x, y)$ can be written as

$$f(x, y) = f_{Y|X}(y|x) f_X(x) = f_{X|Y}(x|y) f_Y(y)$$

the product of the conditional pdf of one of them given the other and the marginal pdf of the latter. If the random variables are independent then

$$f(x, y) = f_X(x) f_Y(y)$$

(similarly for p random variables)

2 Learning

Consider a set T of observations $(y_i, \mathbf{x}_i)$ $i = 1, \ldots, n$, (training set) which we assume to be i.i.d. from a distribution with pdf $f(\mathbf{x}, y)$. The random variable Y is called the dependent variable/response/outcome/output. The random variables $(X_1, \ldots, X_p) = X$ are called the predictors, independent variables/inputs, features. In the terminology it is almost implicit that to obtain a random sample $(\mathbf{x}, y)$, we draw first $\mathbf{x}$ and then draw y from its conditional density given $\mathbf{x}$. Namely,

$$f(\mathbf{x}, y) = f_{Y|X}(y|\mathbf{x}) f_X(\mathbf{x})$$

All this however would be true if we actually knew the pdf's $f_{Y|X}(y|\mathbf{x})$, $f_X(\mathbf{x})$, and this will not be the case in our discussions. We do just have a sample T but no access to the mechanism that generated it. We however assume that the theoretical model behind our observations is of the form

$$y = F(\mathbf{x}) + \epsilon \tag{1}$$

where the error ϵ is a random variable that follows some distribution with (cumulative) density function F_ϵ (or to simplify with pdf f_ϵ), because in the model (1), F is a deterministic function.[1] In other words, we can think of the set T using the joint pdf's of X, Y or that of X, ϵ, to which we now turn. We can write

$$f(\mathbf{x}, \epsilon) = f_{\epsilon|X}(\epsilon|\mathbf{x}) f_X(\mathbf{x})$$

Often ϵ and X are independent (in which case $f(\epsilon|\mathbf{x}) = f_\epsilon(\epsilon)$, but no such assumption in this section). Our training set T is generated as follows. We draw a sample $\mathbf{x}_1 \in \mathbb{R}^p$ from $f_X(\mathbf{x})$, then the error term ϵ_1 from the conditional distribution $\epsilon|X = \mathbf{x}_1$, and we obtain

$$y_1 = F(\mathbf{x}_1) + \epsilon_1$$

This however would be true if we knew F and the density $f_{\epsilon|X}(\epsilon|\mathbf{x})$, $f_X(\mathbf{x})$ and this is what one does when one simulates the data. Then we could draw the rest of the training data (assuming independent sampling, for example).

There is an ambiguity in the model (1) since one can add a constant value to ϵ and subtract the same (constant) value from F and leave all values of y

[1] this F is not the cdf of a r.v- despite the abuse of notation with the above-I'm just trying to follow usual conventions

unchanged. To fix it, one assumes $E(\epsilon) = 0$, and more generally when the error and X are not independent

$$E(\epsilon|\mathbf{x}) = 0 \quad \forall\, \mathbf{x}$$

(which implies $E(\epsilon) = 0$). From this relation, it follows that

$$E(Y|X = \mathbf{x}) = F(\mathbf{x})$$

In learning, the goal is to obtain an estimator $\hat{F}$ of the function F in (1) using the training set T. Two goals are generally aimed at. One is that of finding $\hat{F}$ to predict a new value y^{new} of Y when a sample $\mathbf{x}^{new}$ of X is given. The predicted value is just $y^{new} = \hat{F}(\mathbf{x}^{new})$. Other times, the goal is to try to explain how changes in X, in each X_i, affect changes in Y. This goal is often called explanatory. Often it is actually both goals one is interested in.

If we are interested in prediction, the goodness of an estimate $\hat{F}$ of the function x is evaluated in terms of the risk (the average loss)

$$R(F) \quad = \quad E_{XY}\left(L(y, F(\mathbf{x}))\right) = \int L(y, F(\mathbf{x}))\, f(\mathbf{x}, y)\, d\mathbf{x}\, dy$$

the average with respect to the (often unknown) joint population distribution of X, Y. The best predictor $\hat{F}$ is that which minimizes such risk. Namely one tries to find

$$\hat{F}^* = \arg\min_F R(F)$$

or, equivalently, for each $\mathbf{x}$ one looks for

$$\hat{F}^*(\mathbf{x}) = \arg\min_{F(\mathbf{x})} R(F(\mathbf{x}))$$

where

$$R(F(\mathbf{x})) = \int L(y, F(\mathbf{x})) f(y|\mathbf{x})\ dy = E_{Y|X}\left(L(Y, F(\mathbf{x}))|X = \mathbf{x}\right)$$

The form of the loss L depends on the problem. Generally when Y is continuous (often in this case one speaks of regression) the loss is L_2

$$L(y, w) = (y - w)^2$$

In classification, Y is discrete with its values known as classes, a well used function is the 0/1 loss. Next section gives an example of the learning problem (the Bayes rule).

3 Addendum on Bayes rule

Suppose we have a classification problem, with 2 classes labelled 0 and 1. That is, 0 and 1 are the possible values of the output variable Y. In predictive

learning, we can use the following loss to assess the ability of our classifier $\hat{F}$

$$L(y, \hat{F}(\mathbf{x})) \quad = \quad \begin{cases} L(0,1) = L_0 \\ L(1,0) = L_1 \\ L(1,1) = 0 \\ L(0,0) = 0 \end{cases}$$

where L_0 and L_1 are two numbers. Generally $L_0 = L_1$ is often used in which case one speaks of 0-1 loss, where we equally penalize both types of misclassification (i.e. predict 0 when the true value is 1 and predict 1 when the true value is 0). The above form generalizes the 0-1 loss, when we want to penalize differently those errors. We evaluate the goodness of an estimator by computing the risk, that is the expected value, at each fixed point of $\mathbf{x}$

$$R(F(\mathbf{x})) = E_Y\left(L(y, F(\mathbf{x}))|\mathbf{x}\right) = E_{Y|X}\left(L(y, F(\mathbf{x}))\right)$$

With the above loss, such error is called the miss-classification rate and is equal to

$$R(F(\mathbf{x})) = L_1 \cdot \mathcal{I}(F(\mathbf{x}) = 0)P(Y = 1|X = \mathbf{x}) + L_0 \cdot \mathcal{I}(F(\mathbf{x}) = 1)P(Y = 0|X = \mathbf{x})$$

where the indicator variable $\mathcal{I}(A)$ is 1 if A is true and 0 otherwise. Recall that for a discrete random variable with values $Y = 0, 1$,

$$E(h(Y)|X = \mathbf{x}) = h(Y = 0)P(Y = 0|X = \mathbf{x}) + h(Y = 1)P(Y = 1|X = \mathbf{x})$$

where in our case $h(Y = 0) = L_0\mathcal{I}(F(\mathbf{x}) = 1)$ etc.
The value of the risk is either $L_1P(Y = 1|X = \mathbf{x})$ or $L_0\left(1 - P(Y = 1|X = \mathbf{x})\right)$ since for a given $\mathbf{x}$ the classifier $\hat{F}(\mathbf{x})$ will give us just a single value (0 or 1). [Recall also $1 = P(Y = 0|X = \mathbf{x}) + P(Y = 1|X = \mathbf{x})$] Thus we see the the minimum risk is

$$R(F(\mathbf{x})) \geq \min\{L_1P(Y = 1|X = \mathbf{x}), L_0(1 - P(Y = 1|X = \mathbf{x}))\} = R_B(\mathbf{x})$$

which is called the Bayes risk. Which estimator $\hat{F}$ has the minimum risk? Let us verify that the following one, called the Bayes' rule, has the minimum risk (there may be others)

$$\hat{F}_B(\mathbf{x}) = \begin{cases} 1 & \text{if } P(Y = 1|X = \mathbf{x}) > \frac{L_0}{L_0 + L_1} \\ 0 & \text{otherwise} \end{cases}$$

Indeed, if $\mathbf{x}$ is such that $\hat{F}_B(\mathbf{x}) = 1$, (that is $P(Y = 1|X = \mathbf{x}) > \frac{L_0}{L_0+L_1}$) then

$$R(\hat{F}(\mathbf{x})) = L_0 \cdot P(Y = 0|X = \mathbf{x}) = L_0 \cdot (1 - P(Y = 1|X = \mathbf{x}))$$

but this is smaller than $L_1P(Y = 1|X = \mathbf{x})$. Thus R_B is achieved.

Since we do not know $P(Y = 1|X = \mathbf{x})$ we will have to estimate it. Thus we consider measurements collected on the same sample for both Y and X. the

training set $(y_i, \mathbf{x}_i)$. This training data set is used to learn a classification rule $\hat{F}(\mathbf{x}|T)$ for (future) prediction. We obtain first an estimate of $P(Y = 1|X = \mathbf{x})$, which we plug in to get our classification rule

$$\hat{F}(\mathbf{x}|T) = \begin{cases} 1 & \text{if } \hat{P}(Y = 1|X = \mathbf{x}) > \frac{L_0}{L_0+L_1} \\ 0 & \text{otherwise} \end{cases}$$

4 Linear Regression

In linear regression, Y is assumed to be continuous and the function

$$F(\mathbf{x}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p = (1, \mathbf{x}) \cdot \beta$$

where $\beta^T = (\beta_0, \beta_1, \ldots, \beta_p)$ the $(p+1)$-vector of linear coefficients that completely determines the function F. The minimum assumptions on the error we need are

$$E(\epsilon|X = \mathbf{x}) = \mathbf{0}$$

(vector of length n) and that the errors are uncorrelated

$$Cov(\epsilon_i, \epsilon_j|X) = \sigma^2 \delta_{ij}$$

where $\delta_{ii} = 1$ and $\delta_{ij} = 0$ for $i \neq j$ (the Kronecker delta). In other words, the covariance matrix is the $n \times n$ diagonal matrix with non zero entries σ^2. Define the $n \times (p+1)$ matrix X, which is the matrix with the first column being a column of 1's and the other p columns the training observations, that is $X = (1_n, X(p))$ with length-n vector $1_n^T = (1, \ldots, 1)$ and $X(p)$ the $n \times p$ matrix whose ij entry $X(p)_{ij} = x_{ij}$ is the value of the X_j variable on the i-th observation. Also define $y^t = (y_1, \ldots, y_n)$. Then we can write our model as

$$y_i = \sum_{j=1}^{p+1} X_{ij}\beta_j + \epsilon_i$$

or in matrix form

$$Y = X\beta + \epsilon$$

Our goal is that of minimizing the average error

$$RSS(\beta) = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 = (Y - X\beta)^T(Y - X\beta)$$

with respect to β. That is, we are interested in

$$\hat{\beta} = \arg\min_{\beta}(Y - X\beta)^T(Y - X\beta)$$

which are the solutions to the equations (known as the normal equations)

$$X^T(Y - X\beta) = 0$$

namely

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

assuming the inverse of the $(p+1) \times (p+1)$ matrix $X^T X$ exists. This estimator is called the OLS (Ordinary Least Squares) estimator. We have the following result

Theorem. (Gauss-Markov Theorem)
OLS are BLUE
OLS are the Best Linear Unbiased Estimators of β

The linearity means that any $\beta_i = \sum_{j=1}^{n} a_{ij} y_j$, which is evident in the definition. The unbiasedness also is easy to prove (it is intended as conditional unbiasedness given X, when X are random)

$$E\left(\hat{\beta}|X\right) = E\left((X^T X)^{-1} X^T Y|X\right) = (X^T X)^{-1} X^T E\left(Y|X\right) = (X^T X)^{-1} X^T X \beta = \beta$$

Best means that if there exists another estimator $\tilde{\beta}$ of β that is linear and unbiased, than $Cov(\tilde{\beta}) - Cov(\hat{\beta})$ is non negative definite. In other words, if $p = 1$ this would mean that the OLS is best because it has the least variance among all unbiased linear estimators. We do not prove this result, but just compute the covariance matrix

$$Cov\left(\hat{\beta}|X\right)_{ij} = Cov\left(\beta_i, \beta_j|X\right)$$

Now using the fact that the covariance matrix is bilinear and that we are conditioning on X so that $\beta_i = \sum_j A_{ij} y_j$ in which case the $(p+1) \times n$ matrix $A = (X^T X)^{-1} X^T$ is constant and using the Einstein's convention (so that repeated indices are summed over without the need of using the Σ sign to indicate sums)

$$Cov\left(\hat{\beta}|X\right)_{ij} = Cov\left(\beta_i, \beta_j|X\right) = A_{is} A_{jr} Cov\left(Y_s, Y_r|X\right) = \sigma^2 (AA^T)_{ij} = \sigma^2 (X^T X)^{-1}_{ij}$$

we have found that the covariance matrix is $\sigma^2 (X^T X)^{-1}$ which needs to be estimated since σ^2 is an unknown constant.

A Geometric View of the OLS

Consider $Y \in \mathbb{R}^n$. The column vectors $X_i^t = (x_{1i}, \ldots, x_{ni})$ $i = 0, \ldots, p$, of the data span a sub-space Ω in $\mathbb{R}^n$. That is, $\Omega = \{X\mathbf{x} \text{ for any } \mathbf{x}\}$

We are looking for that point on Ω (the vector) such that $\|Y - X\beta\|$ is minimized. It turns out that such point is identified by $\hat{\beta}$ and is such that $Y - X\hat{\beta}$ is orthogonal to Ω. Because of this, it follows

$$\|Y - X\beta\|^2 = \|Y - X\hat{\beta}\|^2 + \|X\beta - X\hat{\beta}\|^2 \geq \|Y - X\hat{\beta}\|^2$$

and thus that the minimum is achieved when $\beta = \hat{\beta}$. Let us prove the orthogonality, which was used in the above inequality. First write

$$H = X(X^T X)^{-1} X^T$$

this is an $n \times n$ projection matrix, known as the hat matrix, because it transforms the vector of the observed responses Y in the vector of fitted responses $\hat{Y} = HY = X\hat{\beta}$. It is a projection matrix being idempotent $H^2 = H$, symmetric $H = H^T$ (T means transposition) and projects on the space Ω. For example, $HX = X$. Another property is $Tr(H) = tr(I_{p+1}) = p + 1$. Similarly the symmetric idempotent matrix $I - H$, I being $n \times n$, is the matrix that projects on the orthogonal complement of Ω.[2] In fact, $(I - H) \cdot H = 0 = H(I - H)$ Notice that the residual vector (the vector whose entries are the residuals, the differences of the observed y_i and fitted $\hat{y}_i$ values is $\hat{E} = Y - \hat{Y} = (I - H)Y$ is thus orthogonal to the fitted values. Thus we have the decomposition

$$Y = \hat{Y} + \hat{E}$$

into two orthogonal components. Now the covariance matrix for the residual is $Var(\hat{E}|X) = (I - H)\sigma^2$ and for the fitted values is $Var(\hat{Y}|X) = H\sigma^2$. Considering the variance of each element it follows, since the variance is positive that $h_{ii} \in (0, 1)$. Among other things this implies $Var(\hat{y}_i|X) \leq \sigma^2 = Var(Y_i|H)$ thus the fitted values have a smaller error than the observed values, because there is information that comes from the surrounding points and that gives a greater precision in the assessment.

Sum of Squares Decomposition

Notice that we can write

$$\begin{aligned} Y^T Y &= Y^T HY + Y^T (I - H)Y = Y^T H^T HY + Y^T (I - H)^T (I - H)Y \\ &= \hat{Y}^T \hat{Y} + \hat{E}^T \hat{E} \end{aligned}$$

recalling that $H^2 = H$ and $H^T = H$ and the same for $I - H$, which can be written as

$$\begin{aligned} Y^T Y &= \hat{Y}^T \hat{Y} + \hat{E}^T \hat{E} \\ SS_{uncorrect} &= SS_{model} + RSS \end{aligned}$$

the (uncorrected) sum of squares $Y^T Y = \sum_{i=1}^n Y_i^2$ is decomposed into the sum of the sum of squares due to the model being fitted $\hat{Y}^T \hat{Y}$ and the residual sum of squares (RSS) which can't be explained in terms of the model/regression. A similar version of the above decomposition of the total variance of the observations Y is obtained using the (usual) sum of squares

$$TSS = \sum_{i=1}^{n} (Y_i - \bar{Y})^2$$

[2]Notice that this does not mean that the projection matrix is an orthogonal matrix, namely that its transpose is its inverse

removing the mean Y of the observed response values. This is because

$$Y_i - \bar{Y} = Y_i - \hat{Y}_i + \hat{Y}_i - \bar{Y} = \hat{E}_i + \hat{Y}_i - \bar{\hat{Y}}$$

this is because $\sum_i \hat{Y}_i = \sum_i Y_i$ (that is $\sum_{i=1} \hat{E}_i = 0$, the fitted values have the same mean as the responses, when X has full rank) and because the vectors involved are orthogonal

$$\begin{aligned} TSS &= (\hat{Y} - \bar{\hat{Y}})^T (\hat{Y} - \bar{\hat{Y}}) + \hat{E}^T \hat{E} \\ &= SS_{regression} + RSS \end{aligned}$$

One can then define an important statistics, R^2 known as the coefficient of determination

$$R^2 = \frac{SS_{regression}}{TSS} = 1 - \frac{RSS}{TSS} \in [0, 1]$$

which is the fraction of the total variability in the observed values that is explained by the regression models. Its value is in the $[0, 1]$ interval and if all observations fall on the regression line $Y_i = \hat{Y}_i$, for all i, then $R^2 = 1$. A large value of R^2 need not imply that the fitted model is a useful one.

It is thus very important to look at plots to use whether the model is an appropriate one. In which case, the higher it is, the better the model is, but one has also to consider that R^2 can never decrease as additional variables are added in the model. Namely if we compare a model with p regressor and a second model obtained by adding to the p regressors another variable, this latter model will have an R^2 bigger than or equal to that of the p-variable model. Because of this sometimes an adjusted criterion is used

$$R^2_{adj} = 1 - \frac{RSS/(n-p-1)}{TSS/(n-1)}$$

which may be smaller when another variable is added to the model.

It can be shown that R^2 is the estimate of the correlation $Cor(Y, \hat{Y})$ of the observations with the fitted values

Estimate for the Error Variance

The error variance σ^2 is unknown. An estimator is given by the mean squared that is the RSS divided by its degrees of freedom

$$\hat{\sigma}^2 = \frac{RSS}{n-p-1} = \frac{\hat{E}^T \hat{E}}{n-p-1}$$

Let us show that this estimator is unbiased

$$\begin{aligned} E\left(\hat{E}^T \hat{E}\right) &= E\left((Y - \hat{Y})^T (Y - \hat{Y})\right) = E\left((Y - HY)^T (Y - HY)\right) \\ &= E\left(Y^T (I - H) Y\right) \\ &= tr\left((I - H) Cov(Y)\right) + E(Y)^T (I - H) E(Y) \\ &= \sigma^2 \cdot tr(I - H) + \beta^T X^T (I - H) X \beta \\ &= \sigma^2 (n - p - 1) \end{aligned}$$

where we have used the following facts: $Cov(Y|X) = \sigma^2 \cdot I_n$, $(I - H)X = X - X = 0$, the $I - H$ being symmetric and idempotent with a trace equal $n - p - 1$ and that for a bilinear form

$$E\left(Y^T AY\right) = tr(A \cdot Cov(Y)) + E(Y)^T AE(Y)$$

The proof of the latter equation is straightforward.

The residual sum of squares RSS is also called the and $\hat{\sigma}$ is also called the residual standard error (RSE).

Equivariance of the OLS estimates under some transformations

We are going to prove a very important property of the OLS. They are equivariant under: 1) rotations of the coordinate axis (more generally orthogonal transformations), 2) scaling of the coordinates axes and 3) translations of the origin of the coordinates. Equivariance means that if we transform (according to any such transformation) the variable axes, carry out the OLS analysis in this new transformed system and then transform back the solution (that is, we apply the inverse transformation) then we obtain the same result as in the original system of coordinates.

A nonsingular affine transformation of the p axes thus is such that the data matrix is transformed

$$X(p)^T \mapsto AX^T(p) + B = \tilde{X}^T(p)$$

where A is an invertible ($det(A) \neq 0$) $p \times p$ matrix and B a $p \times n$ matrix $B_{ij} = b_i$ for some constants b_i $i = 1, \ldots, p$ that is the columns of B are all equal to the vector that defines the translation $(b_1, \ldots, b_p)$. The matrix $X(p)$ is $n \times p$ (that is we are not considering the column of 1's), but we can write the transformation in terms of the usual X matrix $X = (1_n, X(p))$. In particular, we have an orthogonal transformation if A is orthogonal $A^t = A^{-1}$ (the orthogonal group $O(p)$ contains the rotations and the reflections). If $A \in O(p)$ and $B = 0$ we talk of rigid orthogonal transformations (rigid rotations, for short), otherwise of linear-rotation transformation (a rotation followed by a translation).

We are going to prove equivariance with respect to orthogonal transformation (and thus rotations) $A \in O(p)$, $B = 0$, first. Write the transformation as

$$X^T = \begin{pmatrix} 1_n^T \\ X(p)^T \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0_p^T \\ 0_p & A \end{pmatrix} \begin{pmatrix} 1_n^T \\ X(p)^T \end{pmatrix} = \begin{pmatrix} 1_n^T \\ AX(p)^T \end{pmatrix} = \tilde{X}^T$$

where $0_p^T = (0, \ldots, 0)$ the length p vector of zeros and $1_n^T = (1, \ldots, 1)$. Hence

the OLS solution in this coordinate system $\tilde{X}$ is

$$\begin{aligned}
\tilde{\hat{\beta}} &= (\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T Y \\
&= \left[\begin{pmatrix} 1_n^T \\ AX(p)^T \end{pmatrix}\begin{pmatrix} 1_n & X(p)A^T \end{pmatrix}\right]^{-1}\begin{pmatrix} 1_n^T Y \\ AX(p)^T Y \end{pmatrix} \\
&= \left[\begin{pmatrix} 1 & 0_p^T \\ 0_p & A \end{pmatrix}\begin{pmatrix} n & 1_n^T X(p) \\ X(p)^T 1_n & X(p)^T X(p) \end{pmatrix}\begin{pmatrix} 1 & 0_p^T \\ 0_p & A^T \end{pmatrix}\right]^{-1}\begin{pmatrix} n\bar{Y} \\ AX(p)^T Y \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0_p^T \\ 0_p & A^{-T} \end{pmatrix}\begin{pmatrix} n & 1_n^T X(p) \\ X(p)^T 1_n & X(p)^T X(p) \end{pmatrix}^{-1}\begin{pmatrix} 1 & 0_p^T \\ 0_p & A^{-1} \end{pmatrix}\begin{pmatrix} n\bar{Y} \\ AX(p)^T Y \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0_p^T \\ 0_p & A^{-T} \end{pmatrix}\begin{pmatrix} n & 1_n^T X(p) \\ X(p)^T 1_n & X(p)^T X(p) \end{pmatrix}^{-1}\begin{pmatrix} n\bar{Y} \\ X(p)^T Y \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0_p^T \\ 0_p & A^{-T} \end{pmatrix}\hat{\beta}
\end{aligned}$$

where we have used the fact that $1_n^T 1_n = n$, $1_n^T Y = \sum_i y_i = n\bar{Y}$. [Also notice that $1_n^T X(p) = n(\bar{x}_1, \ldots, \bar{x}_p)$ is the vector whose j-th entry is the sum of the values of the j-th predictor $n\bar{x}_j = \sum_{i=1}^n x_{ij}$]. Thus if $A \in O(p)$ we have

$$\tilde{\hat{\beta}} = \begin{pmatrix} 1 & 0_p^T \\ 0_p & A \end{pmatrix}\hat{\beta}$$

and thus applying the inverse transformation we obtain the OLS in the original system (equivariance)

$$\begin{pmatrix} 1 & 0_p^T \\ 0_p & A \end{pmatrix}^{-1}\tilde{\hat{\beta}} = \hat{\beta}$$

We also have a very important corollary that the fitted values are actually invariant for any transformation not necessarily an orthogonal one

$$\hat{\tilde{Y}} = \tilde{X}\tilde{\hat{\beta}} = X\hat{\beta} = \hat{Y}$$

For a scaling transformation, consider $A = diag(a_1, \ldots, a_p)$ which rescales the i-th coordinates by a_i, the above equation gives since $A^T = A$

$$\tilde{\hat{\beta}} = \begin{pmatrix} 1 & 0_p^T \\ 0_p & A^{-1} \end{pmatrix}\hat{\beta}$$

In other words if any regressor X_i is multiplied by a constant a_i, $\tilde{X}_i = a_i X_i$, and we fit a model using $\tilde{X}_i$, then the regression coefficient $\tilde{\hat{\beta}}_i$ multiplied by a_i gives the regression coefficient of the predictor X_i

$$\tilde{\hat{\beta}}_i \cdot a_i = \hat{\beta}_i$$

Another Interpretation of OLS

(To add)

OLS in R

The function lm() constructs a linear model. It is necessary to specify the response and the predictors. For example lm (Y $\sim$ X1+X2). If the data are in a form in which the columns have names one can call them directly, specifying data=Nameofdata in the command. For categorical variables, use as.factor(NameOfVariable).

Suppose we have called the fitted model model. To read the results of the analysis one can use the function summary(model) which contains some information on the distribution of the residuals (minimum value, maximum value, first, second and third quartiles). The values of the OLS estimates $\hat{\beta}$, the standard error $sd(\beta_i)$, the t values $t = \frac{\hat{\beta}_i}{sd(\beta_i)}$ and the associated p value for the following test

$$H_0 : \beta_i = 0 \quad \text{vs} \quad H_a : \beta_i \neq 0$$

assuming all other β's are fixed. This test is a t test on the assumptions that the errors are normal, because this would make the t statistic follow a t distribution with $\nu = n - p - 1$ degrees of freedom when H_0 is true. In addition, the following quantities are summarized: the residual standard error which is what we indicated with $\hat{\sigma}$, and the number of degrees of freedom $n - p - 1$; the R^2 and the $R^2_{adjusted}$. Finally the F-statistic, and the corresponding p value.

The F statistic is generally given by

$$F = \frac{(RSS_{NH} - RSS_{AH})/(df_{NH} - df_{AH})}{RSS_{AH}/df_{AH}}$$

and is used to test two different models

$$H_0 : \text{Null Model } NH \quad \text{vs.} \quad H_a : \text{Alternative Model } AH$$

where the alternative model AH will be bigger and it will *contain as a submodel* the null model NH. RSS_{NH} refers to the residual sum of squares, $\sum_{i=1}^{n} \hat{E}_i^2$, (the sum of the squares of the residuals $\hat{E}_i = Y_i - \hat{Y}_i$) for the null model, that is, the null model is fitted and $\hat{Y}_i$ is the i-th fitted value, $\sum_{j=0}^{p} X_{ij}\hat{\beta}_j^{NH}$ with $\hat{\beta}^{NH}$ the OLS estimates of the coefficients. Similarly, RSS_{AH} is the residual sum of squares for the alternative model. df indicates the number of degrees of freedom, given by $df = n - p'$ where p' is the total number of coefficient estimates (including β_0). The above F statistic, if H_0 is true (the null model is true), follows an F-distribution with the following degrees of freedom $(\nu_1, \nu_2) = (df_{NH} - df_{AH}, df_{AH})$ if the errors are assumed normal. The F statistic and test reported in the summary of lm refer to the test

$$H_0 : \beta_1 = \beta_2 = \ldots = \beta_p = 0 \quad \text{vs} \quad H_a : \text{at least one } \beta_i \neq 0$$

that is, the null model has only the intercept lm(Y $\sim$ 1) and the alternative model is the original fitted model with all regressors. In which case, if H_0 is true, F follows an F-distribution with $(\nu_1, \nu_2) = (p, n - p - 1)$ if the errors are assumed normal.

Sum of squares and tests to compare models can be also obtained using the command anova and the command Anova in the package car. In particular, the command Anova(model, type=2) gives a table in which different submodels of the model with all predictors (full model) are studied, following the marginality principle, while anova compares nested models differing by one term, with terms added sequentially (to match the sequence with which the terms were entered in the model). Read the help in R to see which models the tables are comparing with the two different functions. If two specific nested models (that is one is contained in the other) are studied one can use simply anova(smallermodel, biggermodel).

The OLS estimates can also be found from the command coefficients(model) or model$coef. The command confint(model, level =k) return the $k\%$ CI of the OLS estimates assuming a normal distribution of the errors (the default level is $k = 0.95$, unless a different level is specified).

The residuals are given by residuals(model) (see later). The fitted values by model$fitted.values or also predict(model). The function predict more generally gives predicted values and (prediction intervals, see below) at a specified new set of $\mathbf{x}$ (one needs to add the argument new.data), and if no new data are specified the function predicts the response for the $\mathbf{x}$ in the training set, that is, it returns the fitted values.

Prediction Intervals

Suppose we want to predict the value of the response Y for an observation with covariate values $\mathbf{x}$. What we assume is that this new observation comes from the same family of distributions of the training data. For example if the errors are normal, the predicted value y will be a random sample from a normal distribution. The specific distribution is one that has mean $E(Y|X = \mathbf{x}) = \beta^T\mathbf{x}$ and variance σ^2. Even if we knew exactly σ^2 and β we would not know the value of y but only that is a random sample from the distribution with mean $E(Y|X = \mathbf{x}) = \beta^T\mathbf{x}$ and variance σ^2. If we just chose one single value from this distribution, we would choose its mean. Which is the reason why the predicted value is equal to this mean. Of course we do not know the true β or σ^2 so we estimate it with the training sample and for any new value of X, $\mathbf{x}$ we would consider $\widehat{E(Y|X = \mathbf{x})} = \hat{\beta}^T\mathbf{x}$ the predicted value of the response. To quantify the uncertainty of the prediction we provide an interval called prediction interval. This takes account of two sources of uncertainty. The variation in the possible location of the distribution (that is, in the mean function) $Var(\widehat{E(Y|X = \mathbf{x})})$ and the variation within the probability distribution of Y. The second would always be there even if we knew the true mean (and thus the true β) and variance of the distribution of Y. In other word, if the model were correct, the true value for an observation with covariate values $\mathbf{x}$ would be

$$\beta^T\mathbf{x} + \epsilon$$

with a random error, so the predicted value, even if β and σ were known, would never match the true value because of the error. The variance of the predicted

point when we use the estimate of β is

$$Var(\hat{\beta^T}\mathbf{x}) + Var(\epsilon) = \mathbf{x}^T(X^TX)^{-1}\mathbf{x}\sigma^2 + \sigma^2$$

where the first part is the uncertainty in the location of the distribution and the second the uncertainty of the distribution. This variance is estimated using the estimate σ^2 and it is this estimate that is used to compute prediction interval

$$\hat{\beta^T}\mathbf{x} \pm t_{\alpha/2,n-(p+1)} \cdot \hat{\sigma} \cdot \sqrt{\mathbf{x}^T(X^TX)^{-1}\mathbf{x} + 1}$$

Of course, if n is large enough one will use the z critical value in place of the t as above. If n is small, and the errors are not normal (which is what we are assuming using the t critical value above) we would consider bootstrap CI (to be described later in the course).

In R the function predict returns also prediction intervals, if one uses the option interval="prediction" and confidence intervals if interval="confidence", where only the uncertainty in the mean function is considered.

Additional Visual Displays

An important question in a model with many regressors ($p > 1$) is to understand the meaning of the regressor coefficients. We can also ask whether the the model

$$y_i = \sum_{j=1}^{p+1} X_{ij}\beta_j + \epsilon_i \quad i = 1, \ldots, n \tag{2}$$

is actually different from the p (marginal) models in which only one regressor at a time is used, namely for $j = 1, \ldots, p$

$$y_i = \beta_0 + X_{ij}\beta_j + \epsilon_i \quad i = 1, \ldots, n$$

In general the estimate of the regression coefficient of a regressor, $\hat{\beta}_i$ is different when one considers such a regressor in a multivariable model and in a univariate model. In the univariate model, $\hat{\beta}_i$ is the change in the mean function for a one unit change in the predictor X_i (assuming it is continuous) ignoring all the other predictors; in the multivariable model it is the change in the mean function for a one unit change in the predictor X_i (assuming it is continuous) when all the values of the other predictors are held fixed (and in the additive model such as the one above, the values of the other predictors are immaterial) or as is said, adjusting or controlling for the other variables.

There is a nice way of looking at the regression coefficient β_i of X_i in the multivariate model (2), of assessing the effect of X_i, having adjusted for the effects of the other predictors: it is via the added-variable plot of X_i. This is the plot of the residuals for the regression of Y vs all other regressors (that is omitting X_i) (these residuals indicate the part of the variability of Y that is not explained by these other regressors $X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_p$) versus

the residuals for the regressors of X_i vs the other regressors. The slope of the regression in this added plot is the estimated coefficient β_i of X_i in the regression with all regressors, thus we have a clear visual estimate. In addition one can compare the added-variable plot of X_i with the (marginal) plot of the fitted line of Y vs X_i, whose slope is the univariate β_i. If the points in the added-variable plot show less variation about the straight fitted line, then all the variables, X_i and the rest, act jointly to explain extra variation in Y.

Notice that also the t-test for testing $\beta_i = 0$ in the complete model is essentially the same as the t-test for testing the slope to the 0 in the added-variable plot.

Added-variable plots are easily obtained using the function avPlots(model) from the package car. They can also be used to identify outliers or highly influential points (see later).

As another way of looking at multivariable regression models vs the univariate ones, consider the model $\mathcal{M}_{12}$

$$y_i = \beta_0 + X_{i1}\beta_1 + X_{i2}\beta_2 + \epsilon_i \quad i = 1, \ldots, n$$

and also the two marginal models: $\mathcal{M}_1$

$$y_i = \beta_0 + X_{i1}\beta_1 + X_{i2}\beta_2 + \epsilon_i \quad i = 1, \ldots, n$$

$\mathcal{M}_2$

$$y_i = \beta_0 + +X_{i2}\beta_2 + \epsilon_i \quad i = 1, \ldots, n$$

One cannot predict the joint relationship of X_1 and X_2 by simply knowing the marginal relationship. For example, we can't say a priori what is the relationship of R^2_{12} the R^2 coefficient of $\mathcal{M}_{12}$ with that of the R^2 of the marginal models R^2_1, R^2_2. If the predictors are uncorrelated then $R^2_{12} = R^2_1 + R^2_2$, but it can be $R^2_{12} < R^2_1 + R^2_2$ if the regressors are partly explaining the same variation of Y. It can however also be $R^2_{12} > R^2_1 + R^2_2$ if the variables X_1 and X_2 act jointly. Marginal model plots are drawn by mmps or (the same function) marginalModelPlots in the package car

Other interesting plots when one is studying the multivariable model (2) are effects plots for the variables X_i, which visualize the role of individual regressors in a fitted model. They do contain the same information as the numeric value of the estimates $\hat{\beta}_i$ and their standard errors actually but give a visual display. They can be obtained using the function plot(Effect(X, Data)) in the package effects

Collinearity

An important question to ask is what happens if some regressors are highly correlated or collinear (collinearity means that at least one regressor can be expressed, approximately if not exactly, as a linear combination of the others $\sum_i^p a_i^T X_i = 0$ for some vector a). In this case, regressors are possibly explaining the same variation in Y. Thus, it is difficult for the least squares method to distinguish the separate effects of the regressors on the response variable Y,

and, accordingly, the regression coefficients cannot be interpreted individually, thus making the identification of a true regressor very difficult. With high collinearity, regression coefficients can have the wrong sign and/or many of the predictor variables are not statistically significant when the overall F-test is highly significant. They are generally unstable. This is because the variance and thus the standard error of the regression coefficient estimates are increased by the high correlation. As a consequence of the increased variance, we also get poor predictions using OLS (unless the new data are the same as in the training set).

Collinearity can be detected by looking at the correlation matrix of the predictors, computed by the function cor . A better measure of collinearity is the variance inflation factor (VIF) defined for the i-th regressor as $v_i = 1/(1-R^2_{X_i,X^{(i)}})$, where $R^2_{X_i,X^{(i)}}$ is the square of the multiple correlation between X_i and the remaining columns of the matrix X (or in other words the R^2 for the regression of X_i onto the other predictors). The higher the collinearity the higher v_i, which is always greater than or equal to 1. The VIF is computed by the function vif in the package car. If the VIF is greater than 5 (as a rule of thumb), then the estimate of the associated regression coefficient should not be trusted. In presence of high collinearity other estimates rather than the OLS should be considered. We will return to this in later chapters.

What to do in presence of collinearity? Use other regression methods (for example ridge regression, see later) or use variable selection (that is, do not use all variables)

Regression Diagnostics

Residual Plots

Plots of residuals versus any function of the regressors should resemble a null plot if all assumptions (linearity of the mean function and homoscedasticity of the errors) are correct. Usual choices for residual plots are plots of the residuals versus any regressor, and versus the fitted values. If only one plot is possible (for example when there are many regressors) then the plot of the residuals versus the fitted values is perhaps the most useful.

The function plot(fit,1) plots the residuals $\hat{E}_i = y_i - \hat{y}_i$ vs the fitted values $\hat{y}_i$, the function plot(model,3) plots the square root of the absolute value of the standardized residuals versus the fitted value ($\sqrt{|r_i|}$ vs $\hat{y}_i$). The package car has more advanced functions for regression diagnostic for example the function residualPlot(model) plots the (ordinary) residuals vs the fitted values and the function residualPlots(model) plots the (ordinary) residuals vs each regressor, where model is the output of a lm() call. There are different kind of residuals. The table below lists the different call in R for different residuals.

(Ordinary) Residuals	$\hat{E}_i = y_i - \hat{y}_i$	residuals(model) resid(model) or model$resid
Standardized residuals (or Internally Studentized residuals)	$r_i = \frac{\hat{E}_i}{\hat{\sigma}(1-h_{ii})}$	rstandard(model)
Studentized residuals (or Externally Studentized Residuals)	$t_i = r_i \sqrt{\frac{n-p-2}{n-p-1-r_i^2}}$	rstudent(model)

The ordinary residuals have a variance that depends on σ^2 and on h_{ii}, so their distribution is scale dependent. The standardized residual are defined to remove such a scale dependence. (The studentized residuals use instead of the estimator $\hat{\sigma}$ which depends on the residuals $\hat{E}$, being proportional to $\sqrt{\hat{E}^T\hat{E}}$, one that does not depend on the residuals and a different estimator $\hat{\sigma}_i$ is used for each point).

What to look for? If the residuals plots have some curvature the assumption on the mean function is probably not correct. If the plots have non constant spread generally this indicates that the hypothesis of constant variance of the errors (homoscedasticity) is not correct. In the case of many regressors, the non-constant spread of the residuals may also indicate a wrong assumption on the mean function (non-linearity).

There are also formal test that check the linearity of the mean function $E(Y|X)$ and the homoscedasticity of the errors.

1) Test to look for curvature in the residual plots
   Together with plotting the residuals vs each regressor X_i and vs the fitted values $\hat{Y}$, if a formal test is needed, the function residualPlots(model) carries out a curvature test, which consists in testing two models: one with the original regressors vs one in which an additional regressor is added to the null model, such regressor having the form U^2 where $U = X_i$ is a regressor or $U = \hat{Y}$ the fitted values. The test is the t-test about the coefficient β_{U^2} of U^2 ($H_0 : \beta_{U^2} = 0$ vs $H_a : \beta_{U^2} \neq 0$. When U is $\hat{Y}$ the test is called Tukey's test for non-additivity. Thus it is a very specific quadratic test.

2) Tests on the non-constant variance of the error
   The Cook Weisberg test, also known as the Breusch-Pagan test, is implemented in car by the function ncvTest(model)

It is however preferable to limit the number of tests.

If a residual plot suggests that a constant variance function is false, one may try the following remedies

1) Transform the response Y (sometimes in this instance one talks of variance-stabilizing transformations)
Common transformation are $\sqrt{Y}$, $log(Y)$ (the base of the logarithm is irrelevant), $1/Y$, which are generally appropriate when the variance increases or decreases with the response. See later the section on transforming variables

2) Use weighted least squares (WLS) with weights that may be determined empirically
In this case we assume

$$Var(\epsilon|X) = \sigma^2 W^{-1} \quad E(Y|X) = X\beta$$

with the weight matrix $W = diag(w_1, \ldots, w_n)$, $w_i > 0$. The OLS estimator in this case is

$$\hat{\beta}_W = (X^T W X)^{-1} X^T W Y$$

which is unbiased, and has the following variance

$$Var(\hat{\beta}_W|X) = \sigma^2 (X^T X)^{-1} (X^T W^{-1} X)(X^T X)$$

The corresponding hat matrix is $H_W = W^{1/2} X (X^T W X)^{-1} X^T W^{1/2}$. To carry out WLS in R, one needs to specify inside the function lm() the weight option lm(.., weights=) with the explicit indication of the vector of weights. The problem in this case is that of having an estimate for the weights.

3) Do nothing but then try to correct the misspecified variances
In this case one accommodates for the misspecified variance, by replacing the usual estimate of the variance $Var(\hat{\beta})$ which can be easily computed (by the function vcov(lm()) for example), with a different estimate. A common estimate, known as the HC3 estimate, is

$$Var(\hat{\beta}|X) = (X^T X)^{-1} X^T D X (X^T X)^{-1}$$

where $D = diag(d_1, \ldots, d_n)$ with $d_i = \hat{E}_i/(1 - H_{ii})^2$. This is computed by the function hccm(model, type="hc3") in car. This estimate can also be used to carry out the testing for the regression coefficients (the F test or t-tests). There are some libraries that do this

4) Use generalized linear models that account for the non constant variance that is a function of the mean. We will consider this later

If the linearity of the mean function is questionable one can transform (some of) the regressors and use such transformed variables as the regressors, hoping that in the transformed scale the mean function is linear. It is often necessary to

transform *both* the predictors (or some predictors) and the response to obtain a linear mean function. There are two empirical rules often helpful in regression: 1) the log-rule: if the values of a variable range over more than one order of magnitude and are strictly positive, a log-transformation is likely useful; 2) the range rule: if the range of a variable is considerably less than one order of magnitude, then any transformation of that variable is unlikely to be helpful.

Transformation of Variables

If the variables are categorical, then they are not transformed. Thus the rest of the discussion below applies to continuous variables.

An important first step is to examine the scatter plots of all predictors and variables plot(Data). This is to find indications on the possible transformations to use, although this helps only up to a point, but can suggest for example that no transformation will be helpful (by the range rule), or that a log-transformation may be helpful. One thing one would like to see is that the 2d plots of a predictor X_i vs another predictor X_j have a linear mean function (or at least that they are really not too curved). This at first may seem surprising. However, there are some theoretical works that have shown the connection between the overall goal of finding transformation in which the multiple linear regression matches the data and choosing transformations that make the 2-d plots of predictors have linear mean function. Thus a useful procedure for multiple linear regression analysis will consist in first transforming the predictors so that the plots of any two transformed regressors are not too curved. [3]

When that is achieved (when the plots of pairs of regressors are linear) [4] one can transform the variable Y. The assumption most methods make is that

$$E(Y|X = \mathbf{x}) = g(\beta^t \mathbf{x})$$

for some g which should be determined. One way to determine g consists in looking at the 2d scatterplot of Y vs $\hat{\beta}^T \mathbf{x}$ and inferring the form of g. The methods we describe below for transforming Y however achieve the goal of finding g in a different way: one method looks at the inverse plot of $\hat{\beta}^T \mathbf{x}$ vs Y and the other method (Box Cox) selects the transformation in a family to achieve normality.

Transformation of the Predictors

Assume first that $p = 1$. The transformation of the regressor is generally chosen from a one-parameter family. Scaled power transformations (introduced by Tukey) are often used

$$X \mapsto \psi(X, \lambda) = \begin{cases} (X^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \\ \log(X) & \text{if } \lambda = 0 \end{cases}$$

[3] The assumption is really that the predictors should be linearly related, which is stronger than the property that the 2-d plots of any two regressors should have a linear mean function, but this latter condition is easier to check

[4] when the regressors are approximated linearly related, in fact

which preserve the direction of association (in the sense that if (X, Y) are positively related so are $(\psi(X, \lambda), Y)$. This class is very flexible. For $\lambda \geq 1$ $\psi(X, \lambda)$ is convex and for $\lambda \leq 1$ is concave. It is increasing in X and λ. They are useful for inducing approximate symmetry when X is skewed. More often, a modified family is used which is known as the modified power family of transformations or Box-Cox power family and is obtained multiplying the above function $\psi(X, \lambda)$ by the geometric mean of the variable to the power $1 - \lambda$

$$\psi_{BC}(X, \lambda) = \psi(X, \lambda) \cdot \exp(\sum_{i=1}^{n} \log(x_i)/n)^{1-\lambda}.$$

The geometric mean term guarantees the units of the transformed variables are the same for all values of λ so that the RSS is in the same scale. This is important because the parameter λ is chosen by minimizing the RSS for the λ-dependent models

$$E(Y|X) = \beta_0 + \beta_1 \psi_{BC}(X, \lambda)$$

The function invTranPlot(x,y) in car draws a two-dimensional scatterplot of the training data (x_i, y_i), along with the OLS fit from the regression of Y on the transformed X ($\psi_{BC}(X, \lambda)$ or $\psi(X, \lambda)$ or other transformations) for different values of λ (very rarely are the values chosen outside the $[-2, 2]$ interval) to be specified with an option (lambda=c(-1.2,1,1), for example, some default values are used if not specified) and for the value of λ that minimizes the RSS. This value is computed by the program. It is a useful method for the case of $p = 1$ regressor. The function invTranEstimate(x,y) also computes the minimizer value λ and gives a CI but does not plot. The default family used to transform the regressor is the modified power family above, family=bcPower, but one can choose another class of transformations, which, unlike the above, are valid also for non-strictly positive variables or even specify other one-parameter families of transformations and the program will determine the best such parameter by minimizing the RSS. Generally if the scaled power transformations or the modified power family of transformations are used once λ is determined one can fit the model using a simple power transformation X^{λ} (rather than the curve in the family used) with the best λ found, since this simplifies the interpretability.

If there are many predictors, $p > 1$, and one needs to transform many predictors simultaneously, one can use a generalization of a method known as the Cox-Box method or actually a modification thereof since the Box-Cox model was developed for the response (in fact when applied to the X, there is no regression model Y onto X at all, since no use is made in such a method of the Y when the X are transformed). The details of such a method and of its generalizations are beyond the scope of our course, but the underlying principle is that of transforming the regressors in the model (or a chosen subset) according to a one-parameter family of transformations, and the parameter is computed by an optimization method (by maximizing the log-likelihood function). The powerTransform(X) function in car implements this method, where X is the matrix of predictors. It can also be called also as powerTransform(cbind(X_1, X_2, ...) $\sim$ 1, data=NameofData). Its summary gives the estimated values of the

λ for each regressor and the CI. If the CI contains zero, then we can take the $\hat{\lambda} = 0$ and simply considering taking the logarithm of that predictor. Do try to take values that are easy to interpret even if those estimated are a little bit different.

One can think of the Box-Cox method as an empirical method, as there is no guarantee that the transformed regressors have a linear mean function. Thus, it is important to check that the plots of the transformed regressors have a linear mean function. Notice also that outliers affect the transformations. In particular, pay attention at high values of λ (5, say) as this is an indication that something is not quite right. One should consider the presence of outliers and see how the transformations suggested by the Box-Cox methods change when outliers are present or are removed.

Transformation of the Response Y

After the transformation of the regressors, if those were in need of transformation, one should modify, if it is necessary, the response. Two methods that can be used are: the inverse response plot method and the Box Cox Method. The inverse response plot is an application of the methodology for transforming a single predictor, described above, but while there one considers regression of Y onto the transformed regressor, here one considers regression of the fitted value $\hat{Y}$ in the model with the untransformed Y) onto the transformed Y.

Notice that the two methods need not give rise to the same transformation for Y (the same value of λ for example).

In the inverse response method one plots $\hat{Y}$ (the fitted value of the regression of Y onto the predictors, that is of the model whose response we are trying to transform) against Y and several curves that are the OLS estimates of the regression of $\hat{Y}$ on $\psi(Y, \lambda_y)$ (hence the name inverse transform). In other word the starting point is

$$E(\hat{Y}|Y) = \alpha_0 + \alpha_1 \psi(Y, \lambda_y)$$

where ψ is a one-parameter transformation among the class of curves considered above, although the function inverseResponsePlot(model), which implements this method, actually only uses simple power functions $\psi(Y, \lambda_y) = Y^{\lambda_y}$. By inspecting the plots for different λ_y one chooses the best curve. In addition to values of λ_y specified by the user, the λ_y that minimizes the RSS is also computed.

A second method is the Box-Cox method applied to Y implemented powerTransform(model), with possibly specifying the family of transformations, the default ones being the Box-Cox power family but others can be selected.

One will then fit the model with the transformed variables. It can be useful to try to apply again the inverse response method, which should result in a plot with a linear mean function, indicating that no other transformation is necessary.

It should be pointed out that the methods only suggest transformations for the variables X and/or Y, but one needs to verify that the linear model built on the transformed variables does make sense. It may not be possible to fit a linear model no matter how the variables are transformed (especially if we are just using one regressor), as we may not have included in the model important

predictors which interact with others we have considered.

Outliers

An observation whose response Y_i may seem not to correspond to the model fitted to the bulk of the data is called an outlier. In other words, an outlier is a point for which Y_i differs greatly from $\hat{Y}_i$. Graphically an outlier can be seen in the plots of the response Y vs a predictor or the fitted value. Generally candidates for outliers are points with large residuals (although not all such points are outliers). As a rule of thumb, one should consider as outliers those points such that $|t_i| > 2$, where t is the studentized residual. It is also to consider outliers points whose standardized residuals r are outside $[-2, 2]$ for small data sets or outside $[-4, 4]$ for large data sets.

There are also some formal tests that can be carried out. One such test is based on the fact that the studentized residuals t_i has a central t-distributions with $n - (p + 1)$ degrees of freedom (under the assumptions that the errors are normal). Thus the value i with the largest $|t_i|$ can be analysed. Doing so actually corresponds to carry out not one test but n tests, so if n is large there are problems of multiple testing. The function outlierTest(lm()) in the library car carry out such a test correcting for multiple testing (Bonferroni correction). It gives a corrected p value, which when small suggests that the point is an outlier.

High Leverage Points

If the i-th observation is such that h_{ii} is large (h_{ii} is the ii elements of the hat matrix H and is known also as the leverage statistic), is said to be a high-leverage point or simply a leverage point. Such points are generally outliers in the X space (have large X values). Generally it is suggested to look carefully at points with $h_{ii} > 2(p+1)/n = 2\sum_j h_{jj}/n$, especially if they have a high residual. The command hatvalues(model) outputs the leverage values $(h_{11}, \ldots, h_{nn})$

Influential Points

Influence analysis is concerned with studying how a model (assumed to be valid) is robust to perturbations. One thing one might consider is how the OLS estimate changes when a point is removed. In particular, there is a quantity that is used in such analysis called Cook's distance. For each point/observation i, it is defined as

$$D_i = \frac{(\hat{\beta}^{(i)} - \hat{\beta})^T X^T X (\hat{\beta}^{(i)} - \hat{\beta})}{(p+1)\hat{\sigma}^2}$$

where $\hat{\beta}$ is the OLS estimate computed using the entire data and $\hat{\beta}^{(i)}$ is the OLS estimate computed removing the i-th observation from the data set. The $D_i = const.$ sets are ellipsoids that can be thought of defining the distance of $\hat{\beta}^{(i)}$ from $\hat{\beta}$. The higher the D_i the more influential the i-th point is. As a rule of thumb points with $D_i > 1$ should be studied in more details (for example one

should redo the analysis without them and see the changes in the results), but in practice, it is important to look for gaps in the values of Cook's distance and not just whether a value exceeds the suggested cut-off. It can be shown that the Cook's distance can be written in terms of the standardized residuals r_i

$$D_i = \frac{1}{p+1} r_i^2 \frac{h_{ii}}{1-h_{ii}}$$

The R function cooks.distance(model) computes the vector $(D_1, \ldots, D_n)$. Notice that points with a large Cook's distance have either high values of h_{ii} (they are high-leverage points), large values of r_i (outliers) or both.

There is also a graphical diagnostic for influence given by the added-variable plots. Points at the left or right of an added-variable plots that do not match the general trend of the plot are likely to be influential.

Other influence measures are DFBETAS, DFFIT, COVRATIO. The following table summarize their meaning and how to compute them in R. In addition influence.measures(model), outputs a table with n rows (one per observation) and the following $p+5$ columns (in order): DFBETAS ($p+1$ columns), DFFIT (dffit, 1 column), COVRATIO (cov.r, 1 column), Cook's distance (cook.d, 1 columns), leverage H_{ii} (hat, 1 column)

Name	measures the effect of the i-th observation	R command
Cook's distance D_i	on $\hat{\beta}$ (entire vector)	cooks.distance
$DFBETAS_j(i)$	on $\hat{\beta}_j$ $j = 0, 1, \ldots, p$	dfbetas ($n \times (p+1)$ matrix)
$DFFITS_i$	on $\hat{Y}_i$	dffits
$COVRATIO_i$	on the variance-covariance matrix of the parameter estimates	covratio

There is also the possibility of plotting some of these quantities. For example the function influenceIndexPlot(model) in car plots the Cook's distances D_i, the h_{ii} values, the studentized residuals t_i and the Bonferroni tests for the outlier test described above.

The simple plot command (used earlier for the residuals) also plots the Cook's distance in the fourth plotplot(model,4) and the fifth plot plot(model,5) is the standardized residuals (r_i) vs the leverage (h_{ii}) and in addition the contours corresponding to some values of the Cook's distance.

Model Selection for Discovery

Unless we have many variables (and in this case we will consider different regression methods in the later part of the course) generally the choice of which

variables to insert in a regression model is based on our knowledge of the problem at hand. Namely, if we are really interested in studying how changes in some specific variables X_1, X_2, etc, which we think are important, affect changes in the variable Y or in predicting Y in terms of some specific variables, we need to use such variables in the model. However there are instances in which we have many variables (p is quite large) as potential regressors and we wonder whether all variables should be kept in the model. This is a problem of model selection. Among all possible 2^p models we want to decide which one is the best, and this need not include all p predictors. Is there an advantage in considering a model with a smaller number of variables over a model with a larger number of variables? A smaller model with fewer variables has the advantage of being simpler and easier to interpret. This is a very important property. However if the goal of our analysis is prediction (of new values of Y), a smaller model may or may not give a better prediction than a larger one. Generally, as one adds more variables to the model, the bias of the predictions decreases and the variance increases. Too few variables give rise to a high bias (underfitting) and too many to high variance (overfitting). A balance between bias and variance is thus important to achieve a prediction.

In this section we will consider some traditional methods to select a regression model, to do model selection (which in this instance is variable selection). The problem of selecting a model will consist in assigning a score to each model that measures the adequacy of the model and then in searching through all possible models that with the best score.

Scoring a model

The criterion to use to assess the adequacy on the model is the risk. The risk depends on the aim of the analysis. If one is considering prediction, then the prediction error (the prediction risk) should be used. Since the risk is not known, as it depends on the joint distribution of X and Y, we need to resort to an estimate of the prediction risk. We will introduce later on an important estimate of the predictor error (based on cross-validation). For this chapter we will consider the following criteria to assess the adequacy of the model, which are mainly used for selecting models for discovery rather than for prediction (even if it is possible to establish a connection with the prediction risk).

1) Mallow's C_p statistic
   This is infact an estimate of the prediction risk of the form

$$R(\mathcal{M}) = \hat{R}_{tr}(\mathcal{M}) + 2|\mathcal{M}|\hat{\sigma}^2 \frac{1}{n}$$

   for a model $\mathcal{M}$, where $|\mathcal{M}|$ is the number of the variables in the model, $\hat{\sigma}^2$ the estimate of σ^2 obtained from the model with all covariates and

$$\hat{R}_{tr}(\mathcal{M}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i(\mathcal{M}))^2$$

is the training error, with $\hat{y}(\mathcal{M})$ being the fitted values for the model $\mathcal{M}$. The training error is an estimate of the prediction error, which is very biased. The C_p statistic therefore corrects the training error by a term that depends on the complexity of the model. If we were to use just the training error as a criterion to select the model, then the largest model would be selected (the more terms one adds, the smaller the training error will be). The additional complexity term is a bias correction term that penalizes larger (more complex model) models. Thus the C_p statistics has the form a term that measures the lack of fit of the model added to a complexity penalty term. The smaller is C_p the better the model.

For a regression model with p' variables (so $p'+1$ is the number of parameters), if p is the maximum number of variables we are fitting, Mallow's statistic becomes

$$C_{p'} = \frac{RSS(p')}{n} + \frac{2(p'+1)}{n}\frac{RSS(p)}{(n-p-1)}$$

Often another definition is used for this statistics. In some books, the C_p statistics is defined differently: it is given by $C^*_{p'}$ where

$$C_{p'} = \frac{\hat{\sigma}^2}{n}\left(C^*_{p'} + n\right)$$

Since the two definitions C_p and C^*_p only differ in a scaling and translation by a quantity that is the same for all models, then the ranking of the models is the same.

2) The Akaike information criterion (AIC), the Bayesian information criterion (BIC).
These criteria assume a distribution for the errors. AIC is a likelihood based criterion. BIC has a Bayesian interpretation. There are different definitions of AIC/BIC differing by an overall constant. A standard definition is

$$AIC(\mathcal{M}) = \mathcal{L}(\mathcal{M}) - |\mathcal{M}| \quad BIC(\mathcal{M}) = \mathcal{L}(\mathcal{M}) - |\mathcal{M}| \cdot \log(n)/2 \qquad (3)$$

where $\mathcal{L}(\mathcal{M})$[5] measures the goodness of fit of the model and the other term the complexity of the model. Thus, the best models are those that have the highest AIC or BIC. If the errors are Gaussian, it can be shown that maximizing the AIC is equivalent to minimizing C_p while BIC puts a more severe penalty for complexity and thus leads one to choose a smaller model than the other methods. The function extractAIC(model, k) computes for $k = 2$ (the default value) $-2AIC$ and for $k = \log(n) = \log(length(residuals(model)))$ computes $-2 \cdot BIC$. There are also other functions, AIC(model) and BIC(model) for example, which

[5]the loglikelihood of the model evaluated at the MLE of the estimates, see Appendix

give values often different from those of extractAIC(model, k), but AIC and BIC as computed in R by the functions above have an *opposite* sign with respect to the definition (3), so when using the values output by R, you should know that the *lower* the value the better the model.

3) The adjusted R^2_{adj} is also used as a score for models
Of course one uses the adjusted R^2_{adj} and not the unadjusted R^2 since the latter can never decrease as additional variables are added in the model, so it will select the model with the largest number of predictors.

Searching the best model

If p is small, one can choose to go through all 2^p models, an exhaustive search. There is also an algorithm that finds the best model without actually computing all possible models. It is the leaps and bound algorithm implemented in the function leaps in the package leaps. It does not work well in presence of interactions, factors or polynomials, and it will work only when p is small ($p < 30$).

However when p is large enough, the exhaustive search and the leaps and bounds algorithm are not feasible, and thus one can resort to a search on a subspace of all models. There are different procedures that differ on where the search for the best model is carried out. We consider the following three procedures

1) Backward Elimination. The starting model is that with all regressors. At each step a model is considered that has one fewer regressor than the current model. If the best of these models is better than the current model than it is set to be the current model, and the procedure is reiterated. Otherwise, the procedure stops with the current model being the accepted model.

2) Forward Selection. The procedure starts with the model consisting of the intercept only. At each step a model is considered that has one more regressor than the current model. If the best of these models is better than the current model than it is set to be the current model, and the procedure is reiterated. Otherwise, the procedure stops with the current model being the final model.

3) The Stepwise method. At each step, the set of candidate models consists of all subsets obtained from the current model by either adding or deleting a term. The model with the best score is accepted as the new candidate model

It is important to note that by term in the above one means a variable or a factor or an interaction term. The set of candidate models is obtained from the current model removing or adding a term but subject to a general principle called the marginality principle. That is, an interaction is never added unless all the lower order terms in the interaction are already present

in the model, and only the highest order effect of an interaction can be removed. For example, if the interaction is made of three variables $A : B : C$, the lower order terms are A, B, C (main effects) $A : B$, $A : C$ (two-factor interactions). Also if at some step the number of terms is already large enough only a subset of the set of candidate models is actually considered.

Notice that there is no guarantee that the best model is indeed among all those that are examined.

The function steps implements the 3 methods above. The option direction is set to "both" (default value), "backward", "forward" to implement one of the three routines above and the starting model should also be specified: e.g. step(model, direction="forward").

5 Generalized Linear Models

6 Logistic Regression

In this section we will consider logistic regression as a method to predict the class for observations that can belong to two classes, which we can label as 0 and 1. The exposition here considers logistic regression as an instance of generalized linear models in which the dependent variables Y are binomial distributions. I actually suggest reading section 4.4 in Hastie, Tibshirani, and Friedman's book for an introduction which can perhaps be felt more natural. There is really no need in this course to consider the binomial distribution, although this is standard.

When we are studying classifiers the goal is in general that of finding an estimate of the (posterior) probability that an observation whose values on a set of variables $X = (X_1, \ldots, X_p)$ are $\mathbf{x}$ belongs to the class k

$$P(Y = k|X = \mathbf{x}) = \pi_k(\mathbf{x})$$

In the binary case, which is what we consider, $k = 0, 1$. Then one can use this estimate in a Bayes classifier in which the class assigned to the values $\mathbf{x}$ is the class with the highest $\pi_k(\mathbf{x})$. Clearly $\sum_k \pi_k(\mathbf{x}) = 1$. Thus in the 2-class case we are considering we just need to estimate the probability of one class, say, of class 1. The Bayes classifier for the 2-class theorem thus implies that an observation is classified as class 1 if $p_1 > 0.5$, but once can also change this value, for example classify as class 1 cases for which $p_1 > 0.7$, and as class 2 cases for which $p_1 < 0.3$ and leave unclassified the cases if their p_1 is outside those ranges.

In binomial regression, one has the variables $Y_i, (X_1, \ldots, X_p)_i$ where the distribution of Y_i is assumed to be $\mathsf{Bin}(n_i, p_i)$. We assume n_i known and we want to estimate p_i, which we think depends on a linear predictor, that is $\beta_0 + \sum_j x_{ij}\beta_j$. We stated at the beginning we are considering only observations that belong to two classes, why is it then that we consider an observation's

dependent variable Y_i as binomial and not just Bernoulli (that is, $n_i = 1$)? Why, in other words, can Y_i have a value such as, say, 3, if the class we are predicting is just 0 or 1? Before answering the question, I suggest to set $n_i = 1$ in all the rest of the discussion, in case of confusion, that is assume that Y_i are Bernoulli. Recall that a Binomial distribution is built from Bernoulli random variables. It is these underlying random variables that are binary and whose values define the classes. p_i refers in fact to the probability that this observation is in class 1.

Suppose we had a set of N (independent) observations that can belong to either of two classes $0, 1$ and that each of these observations were also measured on the set of covariates $X_1, \ldots, X_p$. That is, an observation is $(c_i, \mathbf{x}_i)$ for $i = 1, \ldots, N$ and $c_i = 0, 1$. Among these N observations there may be groups of observations that share the same values of covariates, often called covariate classes. Instead of considering the original N observations one considers just the covariate classes, and to each such group, one associates a value of a new variable Y equal to the sum of the class values of the original observations that belong to such a group, that have the same values of X. Namely, now we think of a different representation of the data as $y_j, \mathbf{x}_j$, where $j = 1, \ldots, n$ and y_j assumes a value in the set $\{0, \ldots, n_j\}$, where n_j is equal to the number of the original observations $(c_i, \mathbf{x}_i)$ that have the same X. Thus Y_i is the number of "successes" (that is of class-1 observations) among the n_i of the original representation of the data that share the same values of X. That is, $Y_i \sim \mathsf{Bin}(n_i, p_i)$, with p_i a function of the covariates $\mathbf{x}_i$. When we have this representation of the data we often speak of grouped data, as opposed to the other representation, the ungrouped data. In particular, we assume that the probability p_i are a function of the linear predictor $\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}$ which I will write as $\beta^T \mathbf{x}_i$, indicating henceforth with $\mathbf{x}$ the vector $(1, x_{i1}, \ldots, x_{ip})$ that also includes 1. Since p_i is a probability, the function that maps the linear predictor to the probability p_i, which is the mean of Y_i/n_i, must be a function with values in $[0, 1]$. The function that takes the linear predictor and gives the mean of the function Y (or a rescaled Y, as in this case) it is called kernel function and its inverse is the link function. In logistic regression we have

$$m(\beta^T \mathbf{x}_i) = \frac{\exp(\beta^T \mathbf{x}_i)}{1 + \exp(\beta^T \mathbf{x}_i)}$$

where $m(x) = (1 + \exp(-x))^{-1}$ is known as the logistic function (hence the name logistic regression [6]). The link function (its inverse) is given by

$$g(p(x)) = \log \frac{p(x)}{1 - p(x)}$$

is called the log-odds function (since it is the logarithm of the odds[7]) or logit

[6] one can also use the term binomial regression because of the distribution of the Y, however, with binomial data, one can also use other link functions, the term logistic regression leaves no ambiguity on which link function is being used

[7] properly we are talking of the odds in favor of success, that is the probability of success over the probability of failure

function. The distribution of Y (or equivalently of the errors) and the form of the link function are what identifies the logistic regression among the other generalized linear models (glms), which are characterized by other distributions for Y and other link functions. If one wants to see the usual linear regression as a generalized linear model, Y is assumed normal and the link function is the identity. Notice that we have not really made any assumptions on the distribution of Y (or of the errors) in our approach to linear regression except when the tests were discussed.

The other assumptions made in the linear regression model are also assumed here and in all glms. That is, that the errors have constant variance and are independent (or at least uncorrelated). We assume independence of the errors and thus of Y_i in this approach to logistic regression. The problem therefore is that of finding estimates of $p_i = P(S|X = \mathbf{x_i})$ is the probability of success in a trial if we think of the binomial as the number of successes in a sequence of independent Bernoulli trials. Estimating p_i requires the estimation of the coefficients of the linear predictor which are determined via maximum likelihood. The likelihood function (see the Appendix) is obtained by looking at the density function of the data as a function of the parameters with the data considered fixed. For a logistic model, the density is

$$f_{Y_1,\ldots,Y_n}(y_1,\ldots,y_n) = \prod_{i=1}^{n} \binom{n_i}{y_i} p_i^{y_i} \left(1-p_i\right)^{n_i - y_i}$$

because of independence of the y_i, $i = 1, \ldots, n$, and hence the likelihood function, or better its log, the loglikelihood function is

$$\begin{aligned} l(\beta_0,\ldots,\beta_p) = l(\beta) &= \log L(\beta_0,\ldots,\beta_p) \\ &= f \sum_{i=1}^{n} y_i \log p(\beta^T \mathbf{x}_i) + (n_i - y_i)\log(1 - p(\beta^T \mathbf{x}_i)) \\ &= \sum_{i=1}^{n} \left(y_i \cdot \beta^T \mathbf{x}_i - n_i \log\left(1 + \exp(\beta^T \mathbf{x}_i)\right)\right) \end{aligned}$$

The estimates for β are the maximizers of the log-likelihood functions. An iterative procedure is required here. Most computer packages use the Fisher scoring algorithm but also Newton-Raphson can be used. They differ in the different Hessian matrix used. A relation exists between weighted least squares estimation and using Fisher scoring to find ML estimates. We will not discuss this.

A question we need to ask is how to assess the model fit. In GLM models, one uses the deviance, which is the logarithm of ratio of likelihoods. In particular, the deviance of a model is the ratio of the likelihood of the so called saturated model over that of the model under consideration which is estimated at values of the parameters equal to the MLE. The saturated model is the model that has a number of parameters equal to the number of observations (so for us equal to n, $n > p$) and thus for which the mean is estimated by the observed

values $\hat{\mu}_i = y_i$. The log-likelihood of the saturated model is the maximum achievable log-likelihood. The saturated model is not really useful, but it serves as a baseline for comparison with other model fits.

It is important to make clear that by parameters we mean here the parameters that define the mean $(\mu_1, \ldots, \mu_p)$ not other parameters on which the mean does not depend (such as the dispersion parameter σ^2 when it exists for the family of models under consideration). To be precise, if μ is the mean of the variables Y_i, $\hat{\mu}_i = y_i$ in the saturated model, and one can define the the scaled deviance as a function of the mean μ

$$D^*(y;\mu) = 2l(y;y) - 2l(\mu;y)$$

For a specific model, the scaled deviance of a model is $D^*(y;\hat{\mu})$ is obtained by plugging in the estimate of the mean function using the model itself. It is more common to use the deviance D instead of the scaled variance, related, as the names suggests, by a scale factor in the form of the dispersion parameter of the model. When Y_i are normal and the link function is the identity (linear regression model) one obtains

$$D(y;\mu) = \sigma^2 D^*(y;\mu) = \sum_{i=1}^{n}(y_i - \mu_i)^2$$

since $l(y;y) = -n/2\log(\sigma^2)$, where $\mu_i = \sum_j x_{ij}\beta_j$. Thus the deviance is the RSS. To compute the deviance of a particular model one plugs in the MLE estimates. The deviance as a function of the parameters that define the mean (β) can also be used as the goodness-of-fit criterion whose minimization can be employed to find estimates of such parameters. Infact, the minimization of the deviance is the same principle as the maximization of the loglikelihood, since the first term in the deviance does not depend on the parameters. From the above expression one can thus see (set $\mu_i = \sum_j x_{ij}\beta_j$ and think of the deviance as function of β) the MLE estimates of β (or equivalently the β that minimize the deviance $D(y;\beta)$) are those that minimize the RSS, that is the OLS estimates.

For binomial Y_i, the scaled deviance is thus the deviance, since there is no dispersion parameter, which is the scaling term. From the formula above we obtain the following expression for the deviance, often indicated in logistic regression as G^2,

$$D = 2\sum_{i=1}^{n}\left(y_i \log\frac{y_i}{\hat{y}_i} + (n_i - y_i)\log\frac{n_i - y_i}{n_i - \hat{y}_i}\right)$$

where $\hat{y}_i = n_i\hat{p}_i = n_i \exp(\hat{\beta}^T\mathbf{x}_i)/(1+\exp(\hat{\beta}^T\mathbf{x}_i))$ (the estimated expected value of Y_i). And again if we see it as a function of β maximizing the deviance is the same as maximizing the likelihood/loglikelihood. The associated number of degrees of freedom is $df = n - d$, where d is the number of parameters estimated $p+1$. D is reduced by adding further covariates to a model. It is the deviance that is reported in the output of the logistics regression analysis and it

is the deviance that is used in general to test models. Namely, to compare two models (a null and a bigger alternative model) one considers the difference of the deviance computed under the two models $D_{NH} - D_{AH}$ and compares it with a χ^2 distribution with $df_{NH} - df_{AH}$ d.o.f. For example, the equivalent of the overall test in linear models, is that in which the model with only intercept (that is the hypothesis that the probability of success is constant) is compared with the full model (the probability depends on all the regressors). The R output gives the deviance D for the model with only the intercept $\beta_i = 0$ when $i > 0$ called the null deviance, and the deviance for the complete model called the residual deviance.

In addition one can consider a goodness-of-fit test by using the residual deviance (of the complete model). That is, one uses the residual deviance to test whether the complete model is appropriate (null hypothesis) or not. Small values of the residual deviance indicate that the model is appropriate (a good fit). One can compare the residual deviance with a a χ^2 distribution with $n - d$ d.o.f. ($d = p + 1$, number of parameters in the model and n number of binomial samples). If the p-values is large there is no evidence to reject the null hypothesis that the model is appropriate, that is no lack of fit, the fit is good. Notice however that if all or nearly all $n_i = 1$ (in this case we speak of sparseness) the deviance D does *not* provide a lack of fit tests and is uninformative. It depends only on the loglikelihood of the model. In other words in logistic regression when $Y_i = 0, 1$ only (Bernoulli data), one does not rely on the deviance as a measure of the goodness-of-fit. Furthermore, the approximate chi-squared distribution for the deviance occurs for n_i large (for grouped data) but not for Bernoulli data (the ungrouped data). However, the test comparing models does not depend on whether the data file has grouped or ungrouped form, because the difference between deviances for two unsaturated models does not depend either on the form of the data. The model comparison statistic often has an approximate chi-squared null distribution even when separate deviances do not.

Thus we can use the deviance to compare a model to a larger one with additional terms (for example a quadratic term, an interaction term) even for Bernoulli regression. By comparing a model with other models that have some additional terms, one is also checking the goodness of fit in a different way. If more complex models do not fit better, this provides some assurance that the model chosen is reasonable. Using the residual deviance instead is a goodness of fit check based on comparing the model at hand with the saturated one (and this is problematic for Bernoulli data as stated above).

Notice also that while the computed values of the deviance are very different (as the number of parameters to be estimated can be very different) in the grouped data and ungrouped data, the grouping of the data should have no effect on the estimates of the regression parameters and their standard errors.

As for testing individual regressors, the statistic used is the ratio of the estimates and the standard error [8]. Logistic regression does not have a variance

[8] the estimated standard error is computed on the basis of the iteratively reweighted least square approximation to the MLE

parameter and the large sample normal approximation is used, which is why the test are called z-tests. This is consequence of the asymptotic properties of MLE estimators (*i/e/* the test is a Wald test). Confidence intervals are more informative than tests, so you may want to consider those, rather than looking at the p-values.

Notice that unlike the linear models, where the t-tests for a coefficient and the F-tests (obtained by comparing the model with or without the corresponding regressor) are identical, in logistic regression the z-tests and the corresponding χ^2 tests based on comparing the deviance of two models are only identical for large sample sizes and in small samples they can give conflicting results. Tests based on deviance are preferred.

The residuals play an important part in the linear model, as they can be used to check the adequacy of a fit. In generalized linear model, a different definition of residuals is required that can be used for the same purpose as the standard residuals. We only consider the Pearson residuals and the deviance residuals. The Pearson residuals are

$$r_P = \frac{y_i - \hat{y}_i}{\sqrt{\widehat{Var(\hat{y}_i)}}} = \frac{y_i - \hat{y}_i}{\sqrt{\hat{y}_i(1 - \hat{y}_i/n_i)}}$$

with $\hat{y}_i = n_i m(\hat{\beta}^T \mathbf{x}_i)$. There is also a standardized version of such residuals that takes account of the variance of $\hat{p}_i$. The deviance residuals are based on using the deviance as a measure of discrepancy in a model. If one thinks that each observation contributes d_i to the deviance, $D = \sum_i d_i$

$$\begin{aligned} r_D &= sign(y_i - \hat{y}_i)\sqrt{d_i} \\ &= sign(y_i - \hat{y}_i)\sqrt{2\left(y_i \log\frac{y_i}{\hat{y}_i} + (n_i - y_i)\log\frac{n_i - y_i}{n_i - \hat{y}_i}\right)} \end{aligned}$$

and similarly there is a standardized version.

Residuals are difficult to interpret for sparse data ($n_i = 1$ always or almost always). If one plots them they appear to have very a non-random pattern, because they are essentially either $1 - \hat{Y}$ (one minus the fitted values) for $Y = 1$ or $-\hat{Y}$ for $Y = 0$, that is consist simply of two parallel lines of dots. Different methods other than residual plots are needed to check the validity of logistic regression models based on binary data. Marginal model plots are a good method to see the adequacy of the model (see below). Notice that we always must check the adequacy of the model. You may have tests that indicate all predictors are significant in a given model, but this does not guarantee the model accurately describes the data (this is of course true also for the linear model, where we use residual plots as diagnostics). One could also group data if possible and compute residuals for the grouped data than for individual subjects, which is better.

As in linear model, one can consider transformation of variables, but in the logistic case, one transforms the predictors and the methodology we have

discussed for the linear model to do so can be applied. Similarly, model selection for validation can be carried out.

With binary data, we never have outliers since there are only two possible values. With grouped data however, we can. As for influence analysis, it should be carried out.

Interpretation of coefficients

Suppose that the model we fit has p regressors. If the covariate X_j increases by one unit and all other covariates are fixed, the odds of success $\hat{p}/(1-\hat{p})$ will be multiplied $\exp(\hat{\beta}_j)$. For example, for $\beta = 0.1615$, $\exp(0.1615) = 1.175$the odds of success will increase by 17.5 percent for each unit increase in X.

Logistic Regression in R

The function glm fits generalized linear model. The syntax inside the function is the same as for lm , except that one needs also to specify which glm one is fitting. This is done by the parameter family. For logistic regression, write family="binomial". For example, to model the probability of an event with the regressors X_1, X_2, X_3 (the latter being categorical) we would write
glm(Y $\sim$ X1+X2+ factor(X3), data=NameofData, family="binomial")
Notice however that if the data are binary $Y_i = 0, 1$ for all i (or Y_i="red" or "blue"), then the Y is just a vector, but if the Y_i are more generally binomial, Y in glm must be actually a matrix with two columns, the first being the number of successes and the second the number of failures.

The output of a glm object includes some similar objects as the lm: coefficients estimates $\hat{\beta}$ (coefficients() or coef, or model$coef, and confint for the CI's), their estimated standard deviations (errors), the fitted values ($\hat{p}_i = m(\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j \mathbf{x}_{ij})$ with m the logistic function), the results of the Wald tests on the coefficients, the residuals. The residuals can also be computed by the usual function: for Pearson residuals residuals(model, "pearson"), while the deviance residuals are the default choice residuals(model) or residuals(model, "deviance"). Notice that the function predict returns the predicted linear predictor $\beta^T \mathbf{x}$ as default value and if one wants instead $m(\beta^T \mathbf{x})$, the predicted probability, one needs to specify the option type="response". In addition, the null deviance null.deviance, the deviance of the model deviance (which, as stated above, in the summary of the output, is called residual variance) and some other quantities about the algorithm used to find the estimates (the steps implemented and if it converged was reached). The plots output by plot(glm()) are the same *mutatis mutandis*. The residuals are computed by the usual function: for Pearson residuals residuals(model, "pearson"), while the deviance residuals are the default choice residuals(model) or residuals(model, "deviance"). The function residualPlot(model) plots the Pearson residuals vs the fitted values and the function residualPlots(model) plots the Pearson residuals vs each regressor. The function anova gives a table of deviances values (gives the values of the deviances of the resulting model) as the terms are added in the model sequen-

tially, or anova(smallermod, biggemodel) gives the residual deviance of the two models, and the difference which can be used for testing by comparing it to a χ^2 distribution. Also Anova in car gives a deviance table with corresponding log-likelihood ratio (deviance) tests to compare models according to the marginality principle.

The usual approach to the analysis (at least for case with a not too large number of regressors) should be do look at the data. It could be helpful to plot $p_i = y_i/n_i$ of the logit of it $\log(p_i/(1-p_i))$, versus the regressors. The latter quantity diverges when $p_i = 0$ or it is not defined when $n_i = 0$. Thus in these cases, when Y has only two values, one could draw boxplots for the predictors for the different levels of the response can be useful. More useful are marginal model plots of Y vs each predictor X_i or the linear predictor. They are useful tools to decide if a logistic regression model is adequate or not. They consist in plotting the observed values of Y vs U (U being a predictor X_i or linear predictor), a non-parametric curve of Y vs U_i (a data curve, that is, it does not use the model fit), and a smooth curve of the points $\hat{Y}$ vs U_i (which are computed using the fitted model). One checks whether there is a reasonable fit of the two smooth curves. If so, the model is adequate in the U "direction". If the two curves do not match, one might consider modifying the model, for example by adding terms (including interactions involving the regressor along which the model is inadequate).

This approach works for continuous predictors. For discrete predictors, one can use contingency tables and cell-by-cell comparisons of observed and fitted counts can help assess the adequacy of the model or suggest a better model.

The function mmps also called with marginalModelPlots in car draws such plots. The smooth curves are drawn generally using a smother (loess, which you can also try out plot(loess(x, y)) for a set of points of with coordinates specified in the vector x and y). The smooth curve of binary response data looks like straight in the middle and flattens out at the end.

Effect plots are also useful to better understand the model.

LDA and QDA

I suggest reading this part in Hastie, Tibshirani, Friedman (ESL).

Appendix. Likelihood Function and MLE

Consider some data Y and let $f(Y|\theta)$ be the density function, where θ are parameters. For example, let $y_i \sim N(\mu, \sigma^2)$ for $i = 1, \ldots, n$ and let Y_i be independent so that the density of the data $Y = (y_1, \ldots, y_n)$ factorizes

$$f(Y|\mu, \sigma^2) = \prod_{i=1}^{n} f(y_i|\mu, \sigma^2) = \prod_{i=1}^{n} \exp\left((y_i - \mu)^2/2\sigma^2\right)/\sqrt{2\pi\sigma^2}$$

The likelihood function is the sample density considered as function of the parameters so that the data are fixed to their observed values (in a sense it is the sample density re-written in a proper order) in which the parameters are unknown and y_i are constant (the observed values)

$$L(\mu, \sigma^2|Y) = f(Y|\mu, \sigma^2)$$

Properly, the equality is up to constant terms (that is all functions of y_i for example can be ignored). Thus for our previous example, if we, as is customary, consider the logarithm of the likelihood function we obtain

$$l(\mu, \sigma^2) = \log L(\mu, \sigma^2|Y) = -\frac{n}{2}\log\sigma^2 + \sum_{i=1}^{n}\frac{(y_i - \mu)^2}{2\sigma^2}$$

Thus methods that are based on the likelihood function requires that the density function is specified. In our derivation, we have only used distributional assumptions on the error and thus on the response conditional on X only when we considered tests. The rest of our derivations made no such assumptions. If we assume the errors to be $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ independent (independence will make the density of the data equal to the density of each observation)), so that

$$y_i = \sum_j x_{ij}\beta_j + \epsilon_i \sim \mathcal{N}(\sum_j x_{ij}\beta_j, \sigma^2)$$

we can talk of likelihood function of the parameters. For example the log-likelihood function of the parameters β, σ^2

$$\log L(\beta, \sigma^2|X, Y) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \sum_j x_{ij}\beta_j\right)^2 - \frac{n}{2}\log\sigma^2$$

Some methods in statistics are likelihood-based. An important method to obtain estimators of parameters is the maximum likelihood estimation (MLE) method, according to which the estimators of parameters θ are the maximizers of the likelihood function or equivalently the maximizers of the log-likelihood function

$$\hat{\theta}_{MLE} = \arg\min_{\theta} L(\theta|data) = \arg\min_{\theta} \log L(\theta|data)$$

MLE need not exist and need not be unique. The MLE method is however widely used, especially because of strong asymptotic properties (consistency and efficiency) of the estimators. In addition, the MLE are parameterization invariant. If $\hat{\theta}$ is the MLE estimator of θ then $h(\hat{\theta})$ is the MLE estimator of $h(\theta)$.