1
Estimating Credit Scores with Logit

Typically, several factors can affect a borrower's default probability. In the retail segment, one would consider salary, occupation, age and other characteristics of the loan applicant; when dealing with corporate clients, one would examine the firm's leverage, profitability or cash flows, to name but a few. A scoring model specifies how to combine the different pieces of information in order to get an accurate assessment of default probability, thus serving to automate and standardize the evaluation of default risk within a financial institution.

In this chapter, we will show how to specify a scoring model using a statistical technique called *logistic regression* or simply *logit*. Essentially, this amounts to coding information into a specific value (e.g. measuring leverage as debt/assets) and then finding the combination of factors that does the best job in explaining historical default behavior.

After clarifying the link between scores and default probability, we show how to estimate and interpret a logit model. We then discuss important issues that arise in practical applications, namely the treatment of outliers and the choice of functional relationship between variables and default.

An important step in building and running a successful scoring model is its validation. Since validation techniques are applied not just to scoring models but also to agency ratings and other measures of default risk, they are described separately in Chapter 7.

LINKING SCORES, DEFAULT PROBABILITIES AND OBSERVED DEFAULT BEHAVIOR

A score summarizes the information contained in factors that affect default probability. Standard scoring models take the most straightforward approach by linearly combining those factors. Let x denote the factors (their number is K) and b the weights (or coefficients) attached to them; we can represent the score that we get in scoring instance i as:

$$\text{Score}_i = b_1 x_{i1} + b_2 x_{i2} + \ldots + b_K x_{iK} \tag{1.1}$$

It is convenient to have a shortcut for this expression. Collecting the b's and the x's in column vectors $\mathbf{b}$ and $\mathbf{x}$ we can rewrite (1.1) to:

$$\text{Score}_i = b_1 x_{i1} + b_2 x_{i2} + \ldots + b_K x_{iK} = \mathbf{b}'\mathbf{x}_i, \quad \mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iK} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_K \end{bmatrix} \tag{1.2}$$

If the model is to include a constant b_1, we set $x_{i1} = 1$ for each i.

Assume, for simplicity, that we have already agreed on the choice of the factors $\mathbf{x}$ – what is then left to determine is the weight vector $\mathbf{b}$. Usually, it is estimated on the basis of the

Table 1.1 Factor values and default behavior

Scoring instance i	Firm	Year	Default indicator for year +1	Factor values from the end of year			
			y_i	x_{i1}	x_{i2}		x_{iK}
1	XAX	2001	0	0.12	0.35	...	0.14
2	YOX	2001	0	0.15	0.51	...	0.04
3	TUR	2001	0	−0.10	0.63	...	0.06
4	BOK	2001	1	0.16	0.21	...	0.12
...	...	...	...	...	...	...	...
912	XAX	2002	0	−0.01	0.02	...	0.09
913	YOX	2002	0	0.15	0.54	...	0.08
914	TUR	2002	1	0.08	0.64	...	0.04
...	...	...	...	...	...	...	...
N	VRA	2005	0	0.04	0.76	...	0.03

observed default behavior.[1] Imagine that we have collected annual data on firms with factor values and default behavior. We show such a data set in Table 1.1.[2]

Note that the same firm can show up more than once if there is information on this firm for several years. Upon defaulting, firms often stay in default for several years; in such cases, we would not use the observations following the year in which default occurred. If a firm moves out of default, we would again include it in the data set.

The default information is stored in the variable y_i. It takes the value 1 if the firm defaulted in the year following the one for which we have collected the factor values, and zero otherwise. The overall number of observations is denoted by N.

The scoring model should predict a high default probability for those observations that defaulted and a low default probability for those that did not. In order to choose the appropriate weights $\mathbf{b}$, we first need to link scores to default probabilities. This can be done by representing default probabilities as a function F of scores:

$$\text{Prob}(\text{Default}_i) = F(\text{Score}_i) \tag{1.3}$$

Like default probabilities, the function F should be constrained to the interval from 0 to 1; it should also yield a default probability for each possible score. The requirements can be fulfilled by a cumulative probability distribution function. A distribution often considered for this purpose is the logistic distribution. The logistic distribution function $\Lambda(z)$ is defined as $\Lambda(z) = \exp(z)/(1+\exp(z))$. Applied to (1.3) we get:

$$\text{Prob}(\text{Default}_i) = \Lambda(\text{Score}_i) = \frac{\exp(\mathbf{b}'\mathbf{x}_i)}{1+\exp(\mathbf{b}'\mathbf{x}_i)} = \frac{1}{1+\exp(-\mathbf{b}'\mathbf{x}_i)} \tag{1.4}$$

Models that link information to probabilities using the logistic distribution function are called *logit* models.

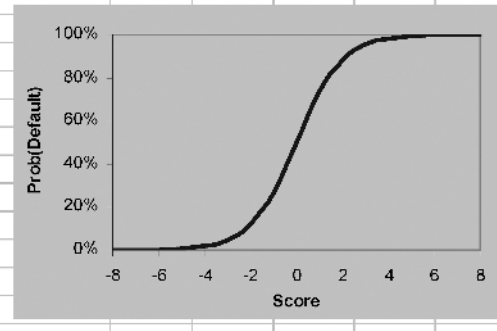
[1] In qualitative scoring models, however, experts determine the weights.

[2] Data used for scoring are usually on an annual basis, but one can also choose other frequencies for data collection as well as other horizons for the default horizon.

In Table 1.2, we list the default probabilities associated with some score values and illustrate the relationship with a graph. As can be seen, higher scores correspond to a higher default probability. In many financial institutions, credit scores have the opposite property: they are higher for borrowers with a lower credit risk. In addition, they are often constrained to some set interval, e.g. 0 to 100. Preferences for such characteristics can easily be met. If we use (1.4) to define a scoring system with scores from -9 to 1, but want to work with scores from 0 to 100 instead (100 being the best), we could transform the original score to $myscore = -10 \times score + 10$.

Table 1.2 Scores and default probabilities in the logit model

	A	B	C	D	E	F	G	H
1	**Score**	**Prob(Default)**						
2	-8	0.03%	=1/(1+EXP(-A2))					
3	-7	0.09%	*(can be copied into B3:B18)*					
4	-6	0.25%						
5	-5	0.67%						
6	-4	1.80%						
7	-3	4.74%						
8	-2	11.92%						
9	-1	26.89%						
10	0	50.00%						
11	1	73.11%						
12	2	88.08%						
13	3	95.26%						
14	4	98.20%						
15	5	99.33%						
16	6	99.75%						
17	7	99.91%						
18	8	99.97%						



Having collected the factors $\mathbf{x}$ and chosen the distribution function F, a natural way of estimating the weights $\mathbf{b}$ is the maximum likelihood method (ML). According to the ML principle, the weights are chosen such that the probability (=likelihood) of observing the given default behavior is maximized. (See Appendix A3 for further details on ML estimation.)

The first step in maximum likelihood estimation is to set up the likelihood function. For a borrower that defaulted ($Y_i = 1$), the likelihood of observing this is

$$\text{Prob}(\text{Default}_i) = \Lambda(\mathbf{b}'\mathbf{x}_i) \tag{1.5}$$

For a borrower that did not default ($Y_i = 0$), we get the likelihood

$$\text{Prob}(\text{No default}_i) = 1 - \Lambda(\mathbf{b}'\mathbf{x}_i) \tag{1.6}$$

Using a little trick, we can combine the two formulae into one that automatically gives the correct likelihood, be it a defaulter or not. Since any number raised to the power of 0 evaluates to 1, the likelihood for observation i can be written as:

$$L_i = (\Lambda(\mathbf{b}'\mathbf{x_i}))^{y_i}(1 - \Lambda(\mathbf{b}'\mathbf{x}_i))^{1-y_i} \tag{1.7}$$

Assuming that defaults are independent, the likelihood of a set of observations is just the product of the individual likelihoods[3]:

$$L = \prod_{i=1}^{N} L_i = \prod_{i=1}^{N} (\Lambda(\mathbf{b}'\mathbf{x}_i))^{y_i} (1 - \Lambda(\mathbf{b}'\mathbf{x}_i))^{1-y_i} \tag{1.8}$$

For the purpose of maximization, it is more convenient to examine ln L, the logarithm of the likelihood:

$$\ln L = \sum_{i=1}^{N} y_i \ln(\Lambda(\mathbf{b}'\mathbf{x_i})) + (1 - y_i) \ln(1 - \Lambda(\mathbf{b}'\mathbf{x_i})) \tag{1.9}$$

This can be maximized by setting its first derivative with respect to **b** to 0. This derivative (like **b**, it is a vector) is given by:

$$\frac{\partial \ln L}{\partial \mathbf{b}} = \sum_{i=1}^{N} (y_i - \Lambda(\mathbf{b}'\mathbf{x}_i))\, \mathbf{x}_i \tag{1.10}$$

Newton's method (see Appendix A3) does a very good job in solving equation (1.10) with respect to **b**. To apply this method, we also need the second derivative, which we obtain as:

$$\frac{\partial^2 \ln L}{\partial \mathbf{b}\, \partial \mathbf{b}'} = -\sum_{i=1}^{N} \Lambda(\mathbf{b}'\mathbf{x_i})(1 - \Lambda(\mathbf{b}'\mathbf{x}_i))\, \mathbf{x}_i \mathbf{x}_i' \tag{1.11}$$

ESTIMATING LOGIT COEFFICIENTS IN EXCEL

Since Excel does not contain a function for estimating logit models, we sketch how to construct a user-defined function that performs the task. Our complete function is called LOGIT. The syntax of the LOGIT command is equivalent to the LINEST command: LOGIT(y, x, [const],[statistics]), where [] denotes an optional argument.

The first argument specifies the range of the dependent variable, which in our case is the default indicator y; the second parameter specifies the range of the explanatory variable(s). The third and fourth parameters are logical values for the inclusion of a constant (1 or omitted if a constant is included, 0 otherwise) and the calculation of regression statistics (1 if statistics are to be computed, 0 or omitted otherwise). The function returns an array, therefore, it has to be executed on a range of cells and entered by [Ctrl]+[Shift]+[Enter].

Before delving into the code, let us look at how the function works on an example data set.[4] We have collected default information and five variables for default prediction: Working Capital (WC), Retained Earnings (RE), Earnings before interest and taxes (EBIT) and Sales (S), each divided by Total Assets (TA); and Market Value of Equity (ME) divided by Total Liabilities (TL). Except for the market value, all of these items are found in the balance sheet and income statement of the company. The market value is given by the number of shares outstanding multiplied by the stock price. The five ratios are those from the widely

[3] Given that there are years in which default rates are high, and others in which they are low, one may wonder whether the independence assumption is appropriate. It will be if the factors that we input into the score capture fluctuations in average default risk. In many applications, this is a reasonable assumption.

[4] The data is hypothetical, but mirrors the structure of data for listed US corporates.

known *Z*-score developed by Altman (1968). WC/TA captures the short-term liquidity of a firm, RE/TA and EBIT/TA measure historic and current profitability, respectively. S/TA further proxies for the competitive situation of the company and ME/TL is a market-based measure of leverage.

Of course, one could consider other variables as well; to mention only a few, these could be: cash flows over debt service, sales or total assets (as a proxy for size), earnings volatility, stock price volatility. Also, there are often several ways of capturing one underlying factor. Current profits, for instance, can be measured using EBIT, EBITDA (=EBIT plus depreciation and amortization) or net income.

In Table 1.3, the data is assembled in columns A to H. Firm ID and year are not required for estimation. The LOGIT function is applied to range J2:O2. The default variable which the LOGIT function uses is in the range C2:C4001, while the factors x are in the range D2:H4001. Note that (unlike in Excel's LINEST function) coefficients are returned in the same order as the variables are entered; the constant (if included) appears as the leftmost variable. To interpret the sign of the coefficient b, recall that a higher score corresponds to a higher default probability. The negative sign of the coefficient for EBIT/TA, for example, means that default probability goes down as profitability increases.

Table 1.3 Application of the LOGIT command to a data set with information on defaults and five financial ratios

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Firm ID	Year	De-fault	WC/ TA	RE/ TA	EBIT/ TA	ME/ TL	S/ TA		CONST	WC/ TA	RE/ TA	EBIT/ TA	ME/ TL	S/ TA
2	1	1999	0	0.50	0.31	0.04	0.96	0.33	b	-2.543	0.414	-1.454	-7.999	-1.594	0.620
3	1	2000	0	0.55	0.32	0.05	1.06	0.33		{=LOGIT(C2:C4001,D2:H4001,1,0)}					
4	1	2001	0	0.45	0.23	0.03	0.80	0.25		*(applies to J2:O2)*					
5	1	2002	0	0.31	0.19	0.03	0.39	0.25							
6	1	2003	0	0.45	0.22	0.03	0.79	0.28							
7	1	2004	0	0.46	0.22	0.03	1.29	0.32							
8	2	1999	0	0.01	-0.03	0.01	0.11	0.25							
9	2	2000	0	-0.11	-0.12	0.03	0.15	0.32							
...															
108	21	1996	1	0.36	0.06	0.03	3.20	0.28							
...															
4001	830	2002	1	0.07	-0.11	0.04	0.04	0.12							

Now let us have a close look at important parts of the LOGIT code. In the first lines of the function, we analyze the input data to define the data dimensions: the total number of observations `N` and the number of explanatory variables (incl. the constant) `K`. If a constant is to be included (which should be done routinely) we have to add a vector of 1's to the matrix of explanatory variables. This is why we call the read-in factors `xraw`, and use them to construct the matrix `x` we work with in the function by adding a vector of 1's. For this, we could use an If-condition, but here we just write a 1 in the first column and then overwrite it if necessary (i.e. if `constant` is 0):

```
Function LOGIT(y As Range, xraw As Range, _
               Optional constant As Byte, Optional stats As Byte)

If IsMissing(constant) Then constant = 1
If IsMissing(stats) Then stats = 0
```

```
'Count variables
Dim i As long, j As long, jj As long

'Read data dimensions
Dim K As Long, N As Long
N = y.Rows.Count
K = xraw.Columns.Count + constant

'Adding a vector of ones to the x matrix if constant=1,
'name xraw=x from now on

Dim x() As Double
ReDim x(1 To N, 1 To K)
For i = 1 To N
 x(i, 1) = 1
 For j = 1 + constant To K
  x(i, j) = xraw(i, j - constant)
 Next j
Next i
…
```

The logical value for the constant and the statistics are read in as variables of type byte, meaning that they can take integer values between 0 and 255. In the function, we could therefore check whether the user has indeed input either 0 or 1, and return an error message if this is not the case. Both variables are optional, if their input is omitted the constant is set to 1 and the statistics to 0. Similarly, we might want to send other error messages, e.g. if the dimension of the dependent variable y and the one of the independent variables x do not match.

In the way we present it, the LOGIT function requires the input data to be organized in columns, not in rows. For the estimation of scoring models, this will be standard, as the number of observations is typically very large. However, we could modify the function in such a way that it recognizes the organization of the data. The LOGIT function maximizes the log likelihood by setting its first derivative to 0, and uses Newton's method (see Appendix A3) to solve this problem. Required for this process are: a set of starting values for the unknown parameter vector **b**; the first derivative of the log-likelihood (the gradient vector $g()$) given in (1.10)); the second derivative (the Hessian matrix $H()$ given in (1.11)). Newton's method then leads to the rule:

$$b_1 = b_0 - \left[\frac{\partial^2 \ln L}{\partial b_0 \, \partial b_0'}\right]^{-1} \frac{\partial \ln L}{\partial b_0} = b_0 - H(b_0)^{-1} g(b_0) \tag{1.12}$$

The logit model has the nice feature that the log-likelihood function is globally concave. Once we have found the root to the first derivative, we can be sure that we have found the global maximum of the likelihood function.

A commonly used starting value is to set the constant as if the model contained only a constant, while the other coefficients are set to 0. With a constant only, the best prediction of individual default probabilities is the average default rate, which we denote by $\bar{y}$; it can be computed as the average value of the default indicator variable y. Note that we should not set the constant b_1 equal to $\bar{y}$ because the predicted default probability with a constant

only is not the constant itself, but rather $\Lambda(b_1)$. To achieve the desired goal, we have to apply the inverse of the logistic distribution function:

$$\Lambda^{-1}(\bar{y}) = \ln(\bar{y}/(1-\bar{y})) \tag{1.13}$$

To check that it leads to the desired result, examine the default prediction of a logit model with just a constant that is set to (1.13):

$$\begin{aligned}\text{Prob}(y=1) = \Lambda(b_1) &= \frac{1}{1+\exp(-b_1)} = \frac{1}{1+\exp(-\ln(\bar{y}/(1-\bar{y})))} \\ &= \frac{1}{1+(1-\bar{y})/\bar{y}} = \bar{y}\end{aligned} \tag{1.14}$$

When initializing the coefficient vector (denoted by `b` in the function), we can already initialize the score $\mathbf{b}'\mathbf{x}$ (denoted by `bx`), which will be needed later. Since we initially set each coefficient except the constant to zero, `bx` equals the constant at this stage. (Recall that the constant is the first element of the vector `b`, i.e. on position 1.)

```
'Initializing the coefficient vector (b) and the score (bx)
Dim b() As Double, bx() As Double, ybar As Double
ReDim b(1 To K): ReDim bx(1 To N)

ybar = Application.WorksheetFunction.Average(y)
If constant = 1 Then b(1) = Log(ybar / (1 - ybar))
For i = 1 To N
 bx(i) = b(1)
Next i
```

If the function was entered with the logical value `constant=0`, the `b(1)` will be left zero, and so will be `bx`. Now we are ready to start Newton's method. The iteration is conducted within a `Do While` loop. We exit once the change in the log-likelihood from one iteration to the next does not exceed a certain small value (like 10^{-11}). Iterations are indexed by the variable `iter`. Focusing on the important steps, once we have declared the arrays `dlnl` (gradient), `Lambda` (prediction $\Lambda(b'x)$), `hesse` (Hessian matrix) and `lnl` (log-likelihood) we compute their values for a given set of coefficients, and therefore for a given score `bx`. For your convenience, we summarize the key formulae below the code:

```
'Compute prediction Lambda, gradient dlnl,
'Hessian hesse, and log likelihood lnl
For i = 1 To N
 Lambda(i) = 1 / (1 + Exp(-bx(i)))
 For j = 1 To K
  dlnL(j) = dlnL(j) + (y(i) - Lambda(i)) * x(i, j)
  For jj = 1 To K
   hesse(jj, j) = hesse(jj, j) - Lambda(i) * (1 - Lambda(i)) _
                  * x(i, jj) * x(i, j)
  Next jj
 Next j
 lnL(iter) = lnL(iter) + y(i) * Log(1 / (1 + Exp(-bx(i)))) + (1 - y(i)) _
             * Log(1 - 1 / (1 + Exp(-bx(i))))
Next i
```

$$\text{Lambda} = \Lambda(\mathbf{b}'\mathbf{x}_i) = 1/(1+\exp(-\mathbf{b}'\mathbf{x}_i))$$

$$\text{dlnl} = \sum_{i=1}^{N}(y_i - \Lambda(\mathbf{b}'\mathbf{x}_i))\,\mathbf{x}_i$$

$$\text{hesse} = -\sum_{i=1}^{N}\Lambda(\mathbf{b}'\mathbf{x}_i)(\,1-\Lambda(\mathbf{b}'\mathbf{x}_i))\,\mathbf{x}_i\mathbf{x}_i'$$

$$\text{lnl} = \sum_{i=1}^{N} y_i \ln(\Lambda(\mathbf{b}'\mathbf{x}_i)) + (1-y_i)\ln(1-\Lambda(\mathbf{b}'\mathbf{x}_i))$$

There are three loops we have to go through. The function for the gradient, the Hessian and the likelihood each contain a sum for `i=1 to N`. We use a loop from `i=1 to N` to evaluate those sums. Within this loop, we loop through `j=1 to K` for each element of the gradient vector; for the Hessian, we need to loop twice, so there's a second loop `jj=1 to K`. Note that the gradient and the Hessian have to be reset to zero before we redo the calculation in the next step of the iteration.

With the gradient and the Hessian at hand, we can apply Newton's rule. We take the inverse of the Hessian using the worksheetFunction MINVERSE, and multiply it with the gradient using the worksheetFunction MMULT:

```
'Compute inverse Hessian (=hinv) and multiply hinv with gradient dlnl
hinv = Application.WorksheetFunction.MInverse(hesse)
hinvg = Application.WorksheetFunction.MMult(dlnL, hinv)

If Abs(change) <= sens Then Exit Do
' Apply Newton's scheme for updating coefficients b
For j = 1 To K
 b(j) = b(j) - hinvg(j)
Next j
```

As outlined above, this procedure of updating the coefficient vector `b` is ended when the change in the likelihood, `abs(ln(iter)-ln(iter-1))`, is sufficiently small. We can then forward *b* to the output of the function LOGIT.

COMPUTING STATISTICS AFTER MODEL ESTIMATION

In this section, we show how the regression statistics are computed in the LOGIT function. Readers wanting to know more about the statistical background may want to consult Appendix A4.

To assess whether a variable helps to explain the default event or not, one can examine a *t* ratio for the hypothesis that the variable's coefficient is zero. For the *j*th coefficient, such a *t* ratio is constructed as:

$$t_j = b_j/\text{SE}(b_j) \tag{1.15}$$

where SE is the estimated standard error of the coefficient. We take *b* from the last iteration of the Newton scheme and the standard errors of estimated parameters are derived from the Hessian matrix. Specifically, the variance of the parameter vector is the main diagonal of

the negative inverse of the Hessian at the last iteration step. In the LOGIT function, we have already computed the Hessian `hinv` for the Newton iteration, so we can quickly calculate the standard errors. We simply set the standard error of the jth coefficient to `Sqr(-hinv(j, j)`. t ratios are then computed using equation (1.15).

In the Logit model, the t ratio does not follow a t distribution as in the classical linear regression. Rather, it is compared to a standard normal distribution. To get the p-value of a two-sided test, we exploit the symmetry of the normal distribution:

$$p\text{-value} = 2 * (1 - \text{NORMSDIST}(\text{ABS}(t))) \tag{1.16}$$

The LOGIT function returns standard errors, t ratios and p-values in lines 2 to 4 of the output if the logical value statistics is set to 1.

In a linear regression, we would report an R^2 as a measure of the overall goodness of fit. In non-linear models estimated with maximum likelihood, one usually reports the Pseudo-R^2 suggested by McFadden. It is calculated as 1 minus the ratio of the log-likelihood of the estimated model ($\ln L$) and the one of a restricted model that has only a constant ($\ln L_0$):

$$\text{Pseudo-}R^2 = 1 - \ln L / \ln L_0 \tag{1.17}$$

Like the standard R^2, this measure is bounded by 0 and 1. Higher values indicate a better fit. The log-likelihood $\ln L$ is given by the log-likelihood function of the last iteration of the Newton procedure, and is thus already available. Left to determine is the log-likelihood of the restricted model. With a constant only, the likelihood is maximized if the predicted default probability is equal to the mean default rate $\bar{y}$. We have seen in (1.14) that this can be achieved by setting the constant equal to the logit of the default rate, i.e. $b_1 = \ln(\bar{y}/(1-\bar{y}))$. For the restricted log-likelihood, we then obtain:

$$\begin{aligned} \ln L_0 &= \sum_{i=1}^{N} y_i \ln(\Lambda(\mathbf{b}'\mathbf{x_i})) + (1 - y_i) \ln(1 - \Lambda(\mathbf{b}'\mathbf{x_i})) \\ &= \sum_{i=1}^{N} y_i \ln(\bar{y}) + (1 - y_i) \ln(1 - \bar{y}) \\ &= N \cdot [\bar{y} \ln(\bar{y}) + (1 - \bar{y}) \ln(1 - \bar{y})] \end{aligned} \tag{1.18}$$

In the LOGIT function, this is implemented as follows:

```
'ln Likelihood of model with just a constant(lnL0)
Dim lnL0 As Double
lnL0 = N * (ybar * Log(ybar) + (1 - ybar) * Log(1 - ybar))
```

The two likelihoods used for the Pseudo-R^2 can also be used to conduct a statistical test of the entire model, i.e. test the null hypothesis that all coefficients except for the constant are zero. The test is structured as a likelihood ratio test:

$$\text{LR} = 2(\ln L - \ln L_0) \tag{1.19}$$

The more likelihood is lost by imposing the restriction, the larger the LR statistic will be. The test statistic is distributed asymptotically chi-squared with the degrees of freedom equal to

the number of restrictions imposed. When testing the significance of the entire regression, the number of restrictions equals the number of variables K minus 1. The function CHIDIST(test statistic, restrictions) gives the p-value of the LR test. The LOGIT command returns both the LR and its p-value.

The likelihoods ln L and ln L_0 are also reported, as is the number of iterations that was needed to achieve convergence. As a summary, the output of the LOGIT function is organized as shown in Table 1.4.

Table 1.4 Output of the user-defined function LOGIT

b_1	b_2	...	b_K
SE(b_1)	SE(b_2)	…	SE(b_K)
$t_1 = b_1/\text{SE}(b_1)$	$t_2 = b_2/\text{SE}(b_2)$	…	$t_K = b_K/\text{SE}(b_K)$
p-value(t_1)	p-value(t_2)	…	p-value(t_K)
Pseudo-R^2	# iterations	#N/A	#N/A
LR test	p-value (LR)	#N/A	#N/A
log-likelihood (model)	log-likelihood (restricted)	#N/A	#N/A

INTERPRETING REGRESSION STATISTICS

Applying the LOGIT function to our data from Table 1.3 with the logical values for constant and statistics both set to 1, we obtain the results reported in Table 1.5. Let's start with the statistics on the overall fit. The LR test (in J7, p-value in K7) implies that the logit regression is highly significant. The hypothesis 'the five ratios add nothing to the prediction' can be rejected with a high confidence. From the three decimal points displayed in Table 1.5, we can deduce that the significance is better than 0.1%, but in fact it is almost indistinguishable from zero (being smaller than 10^{-36}). So we can trust that the regression model helps to explain the default events.

Table 1.5 Application of the LOGIT command to a data set with information on defaults and five financial ratios (with statistics)

	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Default y	WC/TA	RE/TA	EBIT/TA	ME/TL	S/TA		CONST	WC/TA	RE/TA	EBIT/TA	ME/TL	S/TA
2	0	0.50	0.31	0.04	0.96	0.33	b	-2.543	0.414	-1.454	-7.999	-1.594	0.620
3	0	0.55	0.32	0.05	1.06	0.33	SE(b)	0.266	0.572	0.229	2.702	0.323	0.349
4	0	0.45	0.23	0.03	0.80	0.25	t	-9.56	0.72	-6.34	-2.96	-4.93	1.77
5	0	0.31	0.19	0.03	0.39	0.25	p-value	0.000	0.469	0.000	0.003	0.000	0.076
6	0	0.45	0.22	0.03	0.79	0.28	Pseudo-R² / # iter	0.222	12	#N/A	#N/A	#N/A	#N/A
7	0	0.46	0.22	0.03	1.29	0.32	LR test / p-value	160.1	0.000	#N/A	#N/A	#N/A	#N/A
8	0	0.01	-0.03	0.01	0.11	0.25	lnL / $\ln L_0$	-280.5	-360.6	#N/A	#N/A	#N/A	#N/A
9	0	-0.11	-0.12	0.03	0.15	0.32		{=LOGIT(C2:C4001,D2:H4001,1,1)}					
…	…	…	…	…	…	…		*(applies to J2:O8)*					
108	1	0.36	0.06	0.03	3.20	0.28							
…	…	…	…	…	…	…							
4001	1	0.07	-0.11	0.04	0.04	0.12							

Knowing that the model does predict defaults, we would like to know how well it does so. One usually turns to the R^2 for answering this question, but as in linear regression, setting up general quality standards in terms of a Pseudo-R^2 is difficult to impossible. A simple but often effective way of assessing the Pseudo-R^2 is to compare it with the ones from other models estimated on similar data sets. From the literature, we know that scoring models for listed US corporates can achieve a Pseudo-R^2 of 35% and more.[5] This indicates that the way we have set up the model may not be ideal. In the final two sections of this chapter, we will show that the Pseudo-R^2 can indeed be increased by changing the way in which the five ratios enter the analysis.

When interpreting the Pseudo-R^2, it is useful to note that it does not measure whether the model correctly predicted default probabilities – this is infeasible because we do not know the true default probabilities. Instead, the Pseudo-R^2 (to a certain degree) measures whether we correctly predicted the defaults. These two aspects are related, but not identical. Take a borrower which defaulted although it had a low default probability: If the model was correct about this low default probability, it has fulfilled its goal, but the outcome happened to be out of line with this, thus reducing the Pseudo-R^2. In a typical loan portfolio, most default probabilities are in the range of 0.05% to 5%. Even if we get each single default probability right, there will be many cases in which the observed data (=default) is not in line with the prediction (low default probability) and we therefore cannot hope to get a Pseudo-R^2 close to 1. A situation in which the Pseudo-R^2 would be close to 1 would look as follows: Borrowers fall into one of two groups; the first group is characterized by very low default probabilities (0.1% and less), the second group by very high ones (99.9% or more). This is clearly unrealistic for typical credit portfolios.

Turning to the regression coefficients, we can summarize that three out of the five ratios have coefficients b that are significant on the 1% level or better, i.e. their p-value is below 0.01. If we reject the hypothesis that one of these coefficients is zero, we can expect to err with a probability of less than 1%. Each of the three variables has a negative coefficient, meaning that increasing values of the variables reduce default probability. This is what we would expect: by economic reasoning, retained earnings, EBIT and market value of equity over liabilities should be inversely related to default probabilities. The constant is also highly significant. Note that we cannot derive the average default rate from the constant directly (this would only be possible if the constant were the only regression variable).

Coefficients on working capital over total assets and sales over total assets, by contrast, exhibit significance of only 46.9% and 7.6%, respectively. By conventional standards of statistical significance (5% is most common) we would conclude that these two variables are not or only marginally significant, and we would probably consider not using them for prediction.

If we simultaneously remove two or more variables based on their t ratios, we should be aware of the possibility that variables might jointly explain defaults even though they are insignificant individually. To statistically test this possibility, we can run a second regression in which we exclude variables that were insignificant in the first run, and then conduct a likelihood ratio test.

[5] See, e.g., Altman and Rijken (2004).

Table 1.6 Testing joint restrictions with a likelihood ratio test

	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Default y	WC/TA	RE/TA	EBIT/TA	ME/TL	S/TA	**Model 1**	CONST	WC/TA	RE/TA	EBIT/TA	ME/TL	S/TA
2	0	0.50	0.31	0.04	0.96	0.33	b	-2.543	0.414	-1.454	-7.999	-1.594	0.620
3	0	0.55	0.32	0.05	1.06	0.33	SE(b)	0.266	0.572	0.229	2.702	0.323	0.349
4	0	0.45	0.23	0.03	0.80	0.25	t	-9.56	0.72	-6.34	-2.96	-4.93	1.77
5	0	0.31	0.19	0.03	0.39	0.25	p-value	0.000	0.469	0.000	0.003	0.000	0.076
6	0	0.45	0.22	0.03	0.79	0.28	Pseudo-R^2 / # iter	0.222	12	#N/A	#N/A	#N/A	#N/A
7	0	0.46	0.22	0.03	1.29	0.32	LR test / p-value	160.1	0.000	#N/A	#N/A	#N/A	#N/A
8	0	0.01	-0.03	0.01	0.11	0.25	lnL / $\ln L_0$	-280.5	-360.6	#N/A	#N/A	#N/A	#N/A
9	0	-0.11	-0.12	0.03	0.15	0.32		{=LOGIT(C2:C4001,D2:H4001,1,1)}					
10	0	0.06	-0.11	0.04	0.41	0.29		*(applies to J2:O8)*					
11	0	0.05	-0.09	0.05	0.25	0.34	**Model 2**	CONST	RE/TA	EBIT/TA	ME/TL		
12	0	0.12	-0.11	0.04	0.46	0.31	b	-2.318	-1.420	-7.179	-1.616		
13	0	-0.04	0.27	0.05	0.59	0.21	SE(b)	0.236	0.229	2.725	0.325		
14	0	-0.04	0.25	0.03	0.33	0.21	t	-9.84	-6.21	-2.63	-4.97		
15	0	0.00	0.15	0.00	0.16	0.16	p-value	0.000	0.000	0.008	0.000		
16	0	-0.05	0.02	0.01	0.07	0.16	Pseudo-R^2 / # iter	0.217	11	#N/A	#N/A		
17	0	-0.03	-0.01	0.02	0.10	0.18	LR test / p-value	156.8	0.000	#N/A	#N/A		
18	0	-0.03	-0.04	0.02	0.09	0.19	lnL / $\ln L_0$	-282.2	-360.6	#N/A	#N/A		
19	0	0.02	0.05	0.05	0.55	0.07		{=LOGIT(C2:C4001,E2:G4001,1,1)}					
20	0	0.02	0.08	0.03	0.60	0.09		*(applies to J12:M18)*					
21	0	0.03	0.11	0.04	0.79	0.10							
22	0	0.00	0.12	0.04	0.82	0.09	**LR Test for b(WC/TA)=b(S/TA)=0 in model 1**						
23	0	0.04	0.14	0.02	0.63	0.12	LR	3.39	=2*(J8-J18)				
24	0	-0.05	0.15	0.04	0.89	0.15	DF	2					
25	0	-0.01	0.14	0.04	0.68	0.11	p-value	18.39%	=CHIDIST(J23,J24)				
…	…	…	…	…	…	…							
4001	1	0.07	-0.11	0.04	0.04	0.12							

This is shown in Table 1.6. Model 1 is the one we estimated in Table 1.5. In model 2, we remove the variables WC/TA and S/TA, i.e. we impose the restriction that the coefficients on these two variables are zero. The likelihood ratio test for the hypothesis $b_{\text{WC/TA}} = b_{\text{S/TA}} = 0$ is based on a comparison of the log likelihoods ln L of the two models. It is constructed as:

$$\text{LR} = 2[\ln L(\text{model 1}) - \ln L(\text{model 2})]$$

and referred to a chi-squared distribution with two degrees of freedom because we impose two restrictions. In Table 1.6 the LR test leads to value of 3.39 with a p-value of 18.39%. This means that if we add the two variables WC/TA and S/TA to model 2, there is a probability of 18.39% that we do not add explanatory power. The LR test thus confirms the results of the individual tests: individually and jointly, the two variables would be considered only marginally significant.

Where do we go from there? In model building, one often follows simple rules based on stringent standards of statistical significance, like 'remove all variables that are not significant on a 5% level or better'. Such a rule would call to favour model 2. However, it is advisable to complement such rules with other tests. Notably, we might want to conduct an out-of-sample test of predictive performance as it is described in Chapter 7.

PREDICTION AND SCENARIO ANALYSIS

Having specified a scoring model, we want to use it for predicting probabilities of default. In order to do so, we calculate the score and then translate it into a default probability (cf. equations (1.1) and (1.4))[6]:

$$\text{Prob}(\text{Default}_i) = \Lambda(\text{Score}_i) = \Lambda(\mathbf{b}'\mathbf{x}_i) = \frac{1}{1+\exp(-\mathbf{b}'\mathbf{x}_i)} \tag{1.20}$$

In Table 1.7, we calculate default probabilities based on the model with all five ratios. For prediction, we just need the coefficients, so we can suppress the statistics by setting the associated logical value in the LOGIT function to zero.

Table 1.7 Predicting the probability of default

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Default y	WC/ TA	RE/ TA	EBIT/ TA	ME/ TL	S/ TA		CONST	WC/ TA	RE/ TA	EBIT TA	ME/ TL	S/ TA		Default probability
2	0	0.50	0.31	0.04	0.96	0.33	b	-2.543	0.414	-1.454	-7.999	-1.594	0.620		1.16%
3	0	0.55	0.32	0.05	1.06	0.33		{=LOGIT(C2:C4001,D2:H4001,1,0)}							0.91%
4	0	0.45	0.23	0.03	0.80	0.25		*(applies to J2:O2)*							1.75%
5	0	0.31	0.19	0.03	0.39	0.25									3.24%
6	0	0.45	0.22	0.03	0.79	0.28									1.76%
7	0	0.46	0.22	0.03	1.29	0.32									0.82%
8	0	0.01	-0.03	0.01	0.11	0.25									7.10%
9	0	-0.11	-0.12	0.03	0.15	0.32		=1/(1+EXP(-(J$2+SUMPRODUCT(K$2:O$2,D9:H9))))							6.25%
...	...	...	...	...	...	...		*(can be copied into Q2:Q4001)*							...
108	1	0.36	0.06	0.03	3.20	0.28									0.05%
...	...	...	...	...	...	...									...
4001	1	0.07	-0.11	0.04	0.04	0.12									6.74%

We need to evaluate the score $\mathbf{b}'\mathbf{x}_i$. Our coefficient vector $\mathbf{b}$ is in J2:O2, the ratio values contained in $\mathbf{x}_i$ can be found in columns D to H, with each row corresponding to one value of i. However, columns D to H do not contain a column of 1's which we had assumed when formulating Score $= \mathbf{b}'\mathbf{x}$. This is just a minor problem, though, as we can multiply the ratio values from columns D to H with the coefficients for those ratios (in K2:O2) and then add the constant given in J2. The default probability can thus be computed via (here for row 9):

$$= 1/(1+\text{EXP}(-(\text{J\$2}+\text{SUMPRODUCT}(\text{K\$2:O\$2, D9:H9}))))$$

The formula can be copied into the range Q2:Q4001 as we have fixed the reference to the coefficients with a dollar sign. The observations shown in the table contain just two defaulters (in row 108 and 4001), for the first of which we predict a default probability of 0.05%. This should not be cause for alarm though, for two reasons: First, a borrower can

[6] Note that in applying equation (1.20) we assume that the sample's mean default probability is representative of the population's expected average default probability. If the sample upon which the scoring model is estimated is choice-based or stratified (e.g. overpopulated with defaulting firms) we would need to correct the constant b_0 before estimating the PDs, see Anderson (1972) or Scott and Wild (1997).

default even if its default probability is very low. Second, even though a model may do a good job in predicting defaults on the whole (as evidenced by the LR test of the entire model, for example) it can nevertheless fail at predicting some individual default probabilities.

Of course, the prediction of default probabilities is not confined to borrowers that are included in the sample used for estimation. On the contrary, scoring models are usually estimated with past data and then applied to current data.

As already used in a previous section, the sign of the coefficient directly reveals the directional effect of a variable. If the coefficient is positive, default probability increases if the value of the variable increases, and vice versa. If we want to say something about the magnitude of an effect, things get somewhat more complicated. Since the default probability is a non-linear function of all variables and the coefficients, we cannot directly infer a statement such as 'if the coefficient is 1, the default probability will increase by 10% if the value of the variable increases by 10%'.

One way of gauging a variable's impact is to examine an individual borrower and then to compute the change in its default probability that is associated with variable changes. The easiest form of such a scenario analysis is a *ceteris paribus* (c.p.) analysis, in which we measure the impact of changing one variable while keeping the values of the other variables constant. Technically, what we do is change the variables, insert the changed values into the default probability formula (1.20) and compare the result to the default probability before the change.

In Table 1.8, we show how to build such a scenario analysis for one borrower. The estimated coefficients are in row 4, the ratios of the borrower in row 7. For convenience, we include a 1 for the constant. We calculate the default probability (cell C9), very similar to the way we did in Table 1.7.

Table 1.8 Scenario analysis – how default probability changes with changes in explanatory variables

	A	B	C	D	E	F	G	H
1		CONST	WC/TA	RE/TA	EBIT/TA	ME/TL	S/TA	
2								
3	**Estimated model**							
4	Coefficients	-2.543	0.414	-1.454	-7.999	-1.594	0.620	
5								
6	**Data for borrower under analysis**							
7	Ratio values	1	0.50	0.31	0.04	0.96	0.33	
8								
9	=> Default prob:		1.16%	=1/(1+EXP(-SUMPRODUCT(B4:G4,B7:G7)))				
10								
11	**Scenario analysis: Default prob's for c.p. changes of individual variables**							
12	Scenario values for variables							
13	better		0.40	0.40	0.08	1.00	0.20	
14	worse		0.60	0.20	-0.02	0.50	0.40	
15								
16	Scenario default probability							
17	better		1.11%	1.01%	0.87%	1.08%	1.07%	
18	worse		1.21%	1.35%	1.91%	2.37%	1.21%	
19			C18: =1/(1+EXP(-(SUMPRODUCT($B$4:$G$4,$B$7:$G$7)+ C$4*(C14-C$7))))					
20			*(can be copied into C17:B18)*					

In rows 13 and 14, we state scenario values for the five variables, and in rows 17 and 18 we compute the associated default probabilities. Recall that we change just the value of one variable. When calculating the score $\mathbf{b}'\mathbf{x}_i$ by multiplying $\mathbf{b}$ and $\mathbf{x}_i$, only one element in $\boldsymbol{x}_i$ is affected. We can handle this by computing the score $\mathbf{b}'\mathbf{x}_i$ based on the status quo, and then correcting it for the change assumed for a particular scenario. When changing the value of the second variable from x_{i2} to x_{i2}^*, for example, the new default probability is obtained as:

$$\text{Prob}(\text{Default}_i) = \Lambda(b' x_i{}^*) = \Lambda(b' x_i + b_2(x_{i2}{}^* - x_{i2})) \tag{1.21}$$

In cell C18, this is implemented via:

$$= 1/(1 + \text{EXP}(-(\text{SUMPRODUCT}(\$B\$4:\$G\$4, \$B\$7:\$G\$7) + C\$4^*(C14 - C\$7))))$$

We can directly copy this formula to the other cells C17:G17. For example, if the firm manages to increase its profitability EBIT/TA from –2% to 8%, its default probability will move from 1.91% to 0.87%. We could also use the Goal Seek functionality or the Solver to find answers to questions like 'what change in the variable ME/TL is required to produce a default probability of 1%?'.

An analysis like the one conducted here can therefore be very useful for firms that want to reduce their default probability to some target level, and would like to know how to achieve this goal. It can also be helpful in dealing with extraordinary items. For example, if an extraordinary event has reduced the profitability from its long-run mean to a very low level, the estimated default probability will increase. If we believe that this reduction is only temporary, we could base our assessment on the default probability that results from replacing the currently low EBIT/TA by its assumed long-run average.

TREATING OUTLIERS IN INPUT VARIABLES

Explanatory variables in scoring models often contain a few extreme values. They can reflect genuinely exceptional situations of borrowers, but they can also be due to data errors, conceptual problems in defining a variable or accounting discretion.

In any case, extreme values can have a large influence on coefficient estimates, which could impair the overall quality of the scoring model. A first step in approaching the problem is to examine the distribution of the variables. In Table 1.9, we present several descriptive statistics for our five ratios. Excel provides the functions for the statistics we are interested in: arithmetic means (AVERAGE) and medians (MEDIAN), standard deviations (STDEV), skewness (SKEW) and excess kurtosis (KURT),[7] percentiles (PERCENTILE) along with minima (MIN) and maxima (MAX).

A common benchmark for judging an empirical distribution is the normal distribution. The reason is not that there is an *a priori* reason why the variables we use should follow a normal distribution but rather that the normal serves as a good point of reference because it describes a distribution in which extreme events have been averaged out.[8]

[7] Excess kurtosis is defined as kurtosis minus 3.

[8] The relevant theorem from statistics is the central limit theorem, which says that if we sample from any probability distribution with finite mean and finite variance, the sample mean will tend to the normal distribution as we increase the number of observations to infinity.

Table 1.9 Descriptive statistics for the explanatory variables in the logit model

	H	I	J	K	L	M	N	O	P
1	S/ TA			WC/ TA	RE/ TA	EBIT/ TA	ME/ TL	S/ TA	*Formulae for column O, can be copied in columns K to L*
2	0.33		Average	0.14	0.21	0.05	1.95	0.30	=AVERAGE(H$2:H$4001)
3	0.33		Median	0.12	0.22	0.05	1.14	0.26	=MEDIAN(H$2:H$4001)
4	0.25		Stdev	0.17	0.33	0.03	2.99	0.21	=STDEV(H$2:H$4001)
5	0.25		Skewness	-1.01	-2.55	-4.84	7.75	4.48	=SKEW(HS2:H$4001)
6	0.28		Kurtosis	17.68	17.44	86.00	103.13	71.22	=KURT(H$2:HS4001)
7	0.32		Extreme values / Percentiles						
8	0.25		Min	-2.24	-3.31	-0.59	0.02	0.04	=MIN(H$2:H$4001)
9	0.32		0.50%	-0.33	-1.72	-0.05	0.05	0.06	=PERCENTILE(H$2:H$4001,$J9)
10	0.29		1%	-0.17	-0.92	-0.02	0.08	0.07	
11	0.34		5%	-0.06	-0.25	0.02	0.22	0.10	
12	0.31		95%	0.44	0.65	0.09	5.60	0.68	
13	0.21		99%	0.58	0.90	0.12	14.44	1.05	
14	0.21		99.50%	0.63	0.94	0.13	18.94	1.13	
15	0.16		Max	0.77	1.64	0.20	60.61	5.01	=MAX(H$2:H$4001)

A good indicator for the existence of outliers is the excess kurtosis. The normal distribution has excess kurtosis of zero, but the variables used here have very high values ranging from 17.4 to 103.1. A positive excess kurtosis indicates that, compared to the normal, there are relatively many observations far away from the mean. The variables are also skewed, meaning that extreme observations are concentrated on the left (if skewness is negative) or on the right (if skewness is positive) of the distribution.

In addition, we can look at percentiles. For example, a normal distribution has the property that 99% of all observations are within ± 2.58 standard deviations of the mean. For the variable ME/TL, this would lead to the interval $[-5.77, 9.68]$. The empirical 99% confidence interval, however, is [0.05, 18.94], i.e. wider and shifted to the right, confirming the information we acquire by looking at the skewness and kurtosis of ME/TL. Looking at WC/TA, we see that 99% of all values are in the interval $[-0.33, 0.63]$, which is roughly in line with what we would expect under a normal distribution, namely $[-0.30, 0.58]$. In the case of WC/TA, the outlier problem is thus confined to a small subset of observations. This is most evident by looking at the minimum of WC/TA: it is -2.24, which is very far away from the bulk of the observations (it is 14 standard deviations away from the mean, and 11.2 standard deviations away from the 0.5 percentile).

Having identified the existence of extreme observations, a clinical inspection of the data is advisable as it can lead to the discovery of correctable data errors. In many applications, however, this will not lead to a complete elimination of outliers; even data sets that are 100% correct can exhibit bizarre distributions. Accordingly, it is useful to have a procedure that controls the influence of outliers in an automated and objective way.

A commonly used technique applied for this purpose is *winsorization*, which means that extreme values are pulled to less extreme ones. One specifies a certain winsorization level α; values below the α percentile of the variable's distribution are set equal to the α percentile, values above the $1-\alpha$ percentile are set equal to the $1-\alpha$ percentile. Common values for α are 0.5%, 1%, 2% or 5%. The winsorization level can be set separately for each variable in accordance with its distributional characteristics, providing a flexible and easy way of dealing with outliers without discarding observations.

Table 1.10 exemplifies the technique by applying it to the variable WC/TA. We start with a blank worksheet containing only the variable WC/TA in column A. The winsorization level is entered in cell E2. The lower quantile associated with this level is found by applying the PERCENTILE() function to the range of the variable, which is done in E3. Analogously, we get the upper percentile for 1 minus the winsorization level.

Table 1.10 Exemplifying winsorization for the variable WC/TA

	A	B	C	D	E	F
1	WC/TA	WC/TA winsorized				
2	0.501	0.501		Level	2%	
3	0.548	0.521		Lower bound	-0.113	=PERCENTILE(A2:A4001,E2)
4	0.451	0.451		Upper bound	0.521	=PERCENTILE(A2:A4001,1-E2)
5	0.307	0.307				
6	0.447	0.447	=MAX(MIN(A6,E$4),E$3)			
7	0.458	0.458	*(can be copied to B2:B4001)*			
8	0.006	0.006				
9	-0.115	-0.113				
10	0.061	0.061				
11	0.051	0.051				
...	...	....				
4001	0.066	0.066				

The winsorization itself is carried out in column B. We compare the original value of column A with the estimated percentile values; if the original value is between the percentile values, we keep it. If it is below the lower percentile, we set it to this percentile's value; likewise for the upper percentile. This can be achieved by combining a maximum function with a minimum function. For cell B6, we would write

$$= \text{MAX}(\text{MIN}(\text{A6}, \text{E\$4}),\ \text{E\$3})$$

The maximum condition pulls low values up, the minimum function pulls large values down.

We can also write a function that performs winsorization and requires as arguments the variable range and the winsorization level. It might look as follows:

```
Function WINSOR(x As Range, level As Double)

Dim N As Integer, i As Integer
N = x.Rows.Count

'Obtain percentiles
Dim low, up
low = Application.WorksheetFunction.Percentile(x, level)
up = Application.WorksheetFunction.Percentile(x, 1 - level)

'Pull x to percentiles
Dim result
ReDim result(1 To N, 1 To 1)
For i = 1 To N
```

```
    result(i, 1) = Application.WorksheetFunction.Max(x(i), low)
    result(i, 1) = Application.WorksheetFunction.Min(result(i, 1), up)
Next i

WINSOR = result

End Function
```

The function works in much the same way as the spreadsheet calculations in Table 1.10. After reading the number of observations N from the input range x, we calculate lower and upper percentiles and then use a loop to winsorize each entry of the data range. WINSOR is an array function that has as many output cells as the data range that is inputted into the function. The winsorized values in column B of Table 1.10 would be obtained by entering

$$= \text{WINSOR}(\text{A2:A4002}, 0.02)$$

in B2:B4001 and confirming with [Ctrl] + [Shift] + [Enter].

If there are several variables as in our example, we would winsorize each variable separately. In doing so, we could consider different winsorization levels for different variables. As we saw above, there seem to be fewer outliers in WC/TA than in ME/TA, so we could use a higher winsorization level for ME/TA. We could also choose to winsorize asymmetrically, i.e. apply different levels to the lower and the upper side.

Here we present skewness and kurtosis of our five variables after applying a 1% winsorization level to all variables:

	WC/TA	RE/TA	EBIT/TA	ME/TL	S/TA
Skewness	0.63	−0.95	0.14	3.30	1.68
Kurt	0.01	3.20	1.10	13.48	3.42

Both skewness and kurtosis are now much closer to zero. Note that both statistical characteristics are still unusually high for ME/TL. This might motivate a higher winsorization level for ME/TL, but there is an alternative: ME/TL has many extreme values to the right of the distribution. If we take the logarithm of ME/TL, we also pull them to the left, but we don't blur the differences between those beyond a certain threshold as we do in winsorization. The logarithm of ME/TL (after winsorization at the 1% level) has skewness of −0.11 and kurtosis of 0.18, suggesting that the logarithmic transformation works for ME/TL in terms of outliers.

The proof of the pudding is in the regression. Examine in Table 1.11 how the Pseudo-R^2 of our logit regression depends on the type of data treatment.

Table 1.11 Pseudo-R^2s for different data treatments

	Pseudo-R^2
Original data	22.2%
Winsorized at 1%	25.5%
Winsorized at 1% + log of ME/TL	34.0%
Original but log of ME/TL	34.9%

For our data, winsorizing increases the Pseudo-R^2 by three percentage points from 22.2% to 25.5%. This is a handsome improvement, but taking logarithms of ME/TL is much more important: the Pseudo-R^2 subsequently jumps to around 34%. And one can do even better by using the original data and taking the logarithm of ME/TL rather than winsorizing first and then taking the logarithm.

We could go on and take the logarithm of the other variables. We will not present details on this, but instead just mention how this could be accomplished. If a variable takes negative values (this is the case with EBIT/TL, for example), we cannot directly apply the logarithm as we did in the case of ME/TL. Also, a variable might exhibit negative skewness (an example is again EBIT/TL). Applying the logarithm would increase the negative skewness rather than reduce it, which may not be what we want to achieve. There are ways out of these problems. We could, for example, transform EBIT/TA by computing $-\ln(1-\text{EBIT/TA})$ and then proceed similarly for the other variables.

As a final word of caution, note that one should guard against data mining. If we fish long enough for a good winsorization or similar treatment, we might end up with a set of treatments that works very well for the historical data that we optimized it on. It may not, however, serve to improve the prediction of future defaults. A simple strategy against data mining is to be restrictive in the choice of treatments. Instead of experimenting with all possible combinations of individual winsorization levels and functional transformations (logarithmic or other), we might restrict ourselves to a few choices that are common in the literature or that seem sensible, based on a descriptive analysis of the data.

CHOOSING THE FUNCTIONAL RELATIONSHIP BETWEEN THE SCORE AND EXPLANATORY VARIABLES

In the scoring model (1.1) we assume that the score is linear in each explanatory variable x: $\text{Score}_i = \mathbf{b}'\mathbf{x}_i$. In the previous section, however, we have already seen that a logarithmic transformation of a variable can greatly improve the fit. There, the transformation was motivated as an effective way of treating extreme observations, but it may also be the right one from a conceptual perspective. For example, consider the case where one of our variables is a default probability assessment, denoted by p_i. It could be a historical default rate for the segment of borrower i, or it could originate from models like those we discuss in Chapters 2 and 4. In such a case, the appropriate way of entering the variable would be the logit of p_i, which is the inverse of the logistic distribution function:

$$x = \Lambda^{-1}(p) = \ln(p/(1-p)) \quad \Rightarrow \quad \Lambda(x) = p \tag{1.22}$$

as this guarantees that the default prediction equals the default probability we input into the regression.

With logarithmic or logit transformations, the relationship between a variable and the default probability is still monotonic: for a positive coefficient, a higher value of the variable leads to a higher default probability. In practice, however, we can also encounter non-monotonic relationships. A good example is sales growth: low sales growth may be due to high competition or an unsuccessful product policy, and correspondingly indicate high default risk; high sales growth is often associated with high cash requirements (for advertising and inventories), or may have been bought at the expense of low margins. Thus, high sales growth can also be symptomatic of high default risk. All combined, there might be a U-shaped

relationship between default risk and sales growth. To capture this non-monotonicity, one could enter the square of sales growth together with sales growth itself:

$$\text{Prob}(\text{Default}_i) = \Lambda\left(b_1 + b_2\,\text{Sales growth}_i + b_3(\text{Sales growth}_i)^2 + \ldots + b_K x_{iK}\right) \quad (1.23)$$

Similarly, we could try to find appropriate functional representations for variables where we suspect that a linear relation is not sufficient. But how can we guarantee that we detect all relevant cases and then find an appropriate transformation? One way is to examine the relationships between default rates and explanatory variables separately for each variable. Now, how can we visualize these relationships? We can classify the variables into ranges, and then examine the average default rate within a single range. Ranges could be defined by splitting the domain of a variable into parts of equal length. With this procedure, we are likely to get a very uneven distribution of observations across ranges, which could impair the analysis. A better classification would be to define the ranges such that they contain an equal number of observations. This can easily be achieved by defining the ranges through percentiles. We first define the number of ranges M that we want to examine. The first range includes all observations with values below the $(100/M)$th percentile; the second includes all observations with values above the $(100/M)$th percentile but below the $(2 \times 100/M)$th percentile and so forth.

For the variable ME/TL, the procedure is exemplified in Table 1.12. We fix the number of ranges in F1, then use this number to define the alpha values for the percentiles (in D5:D24). In column E, we use this information and the function PERCENTILE(x, alpha) to determine the associated percentile value of our variable. In doing so, we use a minimum condition to ascertain that the α value is not above 1. This is necessary because the summation process in column L can yield values slightly above 1 (Excel rounds to 15 digit precision).

The number of defaults within a current range is found recursively. We count the number of defaults up to (and including) the current range, and then subtract the number of defaults that are contained in the ranges below. For cell F5, this can be achieved through:

$$= \text{SUMIF(B\$2:B\$4001, "<="\&E5, A\$2:A\$4001)} - \text{SUM(F4:F\$4)}$$

where E5 contains the upper bound of the current range; defaults are in column A, the variable ME/TL in column B. Summing over the default variable yields the number of defaults as defaults are coded as 1. In an analogous way, we determine the number of observations. We just replace SUMIF by COUNTIF.

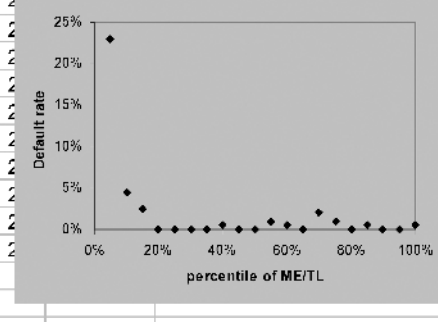
What does the graph tell us? Apparently, it is only for very low values of ME/TL that a change in this variable impacts default risk. Above the 20th percentile, there are many ranges with zero default rates, and the ones that see defaults are scattered in a way that does not suggest any systematic relationship. Moving from the 20th percentile upward has virtually no effect on default risk, even though the variable moves largely from 0.5 to 60. This is perfectly in line with the results of the previous section where we saw that taking the logarithm of ME/TL greatly improves the fit relative to a regression in which ME/TL entered linearly. If we enter ME/TL linearly, a change from ME/TL = 60 to ME/TL = 59.5 has the same effect on the score as a change from ME/TL = 0.51 to ME/TL = 0.01, contrary to what we see in the data. The logarithmic transformation performs better because it reduces the effect of a given absolute change in ME/TL for high levels of ME/TL.

Table 1.12 Default rate for percentiles of ME/TL

	A	B	C	D	E	F	G	H	I
1	Default	ME/TL		Number of ranges:		20			
2	0	0.96							
3	0	1.06		alpha	Range bound	Defaults	Obs	Default rate	
4	0	0.80							
5	0	0.39		5%	0.22	46	200	23.0%	
6	0	0.79		10%	0.33	9	200	4.5%	
7	0	1.29		15%	0.44	5	200	2.5%	
8	0	0.11		20%					
9	0	0.15		25%					
10	0	0.41		30%					
11	0	0.25		35%					
12	0	0.46		40%					
13	0	0.59		45%					
14	0	0.33		50%					
15	0	0.16		55%	1.32	2	2		
16	0	0.07		60%	1.50	1	2		
17	0	0.10		65%	1.71	0	2		
18	0	0.09		70%	1.95	4	2		
19	0	0.55		75%	2.24	2	2		
20	0	0.60		80%	2.65	0	2		
21	0	0.79		85%	3.21	1	2		
22	0	0.82		90%	3.89	0	2		
23	0	0.63		95%	5.60	0	2		
24	0	0.89		100%	60.61	1	2		
25	0	0.68							
26	0	0.58							
...	...	...							
4001	1	0.04							

D5: =D4+1/FS1
E5: =PERCENTILE(BS2:B$4001,MIN(1,D5))
F5: =SUMIF(BS2:B$4001,"<="&E5,A$2:A$4001)-SUM(F4:FS4)
G5: =COUNTIF(B$2:B$4001,"<="&E5)-SUM(G4:G$4)
H5: =F5/G5

Can be copied down into range J5:N24



Thus, the examination of univariate relationships between default rates and explanatory variables can give us valuable hints as to which transformation is appropriate. In case of ML/TE, it supports the logarithmic one; in others it may support a polynomial representation like the one we mentioned above in the sales growth example.

Often, however, which transformation to choose may not be clear; and we may want to have an automated procedure that can be run without us having to look carefully at a set of graphs first. To such end, we can employ the following procedure: we first run an analysis as in Table 1.12. Instead of entering the original values of the variable into the logit analysis, we use the default rate of the range to which they are assigned. That is, we use a data-driven, non-parametric transformation. Note that before entering the default rate in the logit regression, we would apply the logit transformation (1.22) to it.

We will not show how to implement this transformation in a spreadsheet. With many variables, it would involve a lot of similar calculations, making it a better idea to set up a user defined function that maps a variable into a default rate for a chosen number of ranges. Such a function might look like this:

```
Function XTRANS(defaultdata As Range, x As Range, numranges As Integer)
Dim bound, numdefaults, obs, defrate, N, j, defsum, obssum, i
```

```
ReDim bound(1 To numranges), numdefaults(1 To numranges)
ReDim obs(1 To numranges), defrate(1 To numranges)

N = x.Rows.Count

'Determining number of defaults, observations and default rates for ranges
For j = 1 To numranges

  bound(j) = Application.WorksheetFunction.Percentile(x, j / numranges)

  numdefaults(j) = Application.WorksheetFunction.SumIf(x, ''<='' & _
                   bound(j), defaultdata) - defsum
  defsum = defsum + numdefaults(j)
  obs(j) = Application.WorksheetFunction.CountIf(x, ''<='' & bound(j))_
           - obssum
  obssum = obssum + obs(j)

  defrate(j) = numdefaults(j) / obs(j)
Next j

'Assigning range default rates in logistic transformation
Dim transform
ReDim transform(1 To N, 1 To 1)

For i = 1 To N
  j = 1
  While x(i) - bound(j) > 0
    j = j + 1
  Wend
  transform(i, 1) = Application.WorksheetFunction.Max(defrate(j), _
                    0.0000001)
  transform(i, 1) = Log(transform(i, 1) / (1 - transform(i, 1)))
Next i

XTRANS = transform
End Function
```

After dimensioning the variables, we loop through each range, `j=1` to `numranges`. It is the analogue of what we did in range D5:H24 of Table 1.12. That is why we see the same commands: SUMIF to get the number of defaults below a certain percentile, and COUNTIF to get the number of observations below a certain percentile.

In the second loop over $i = 1$ to N, we perform the data transformation. For each observation, we search through the percentiles until we have the one that corresponds to our current observation (`Do While … Loop`) and then assign the default rate. In the process, we set the minimum default rate to an arbitrarily small value of 0.0000001. Otherwise, we could not apply the logit transformation in cases where the default rate is zero.

To illustrate the effects of the transformation, we set the number of ranges to 20, apply the function XTRANS to each of our five ratios and run a logit analysis with the transformed ratios. This leads to a Pseudo-R^2 of 47.8% – much higher than the value we received with the original data, winsorization, or logarithmic transformation (Table 1.13).

Table 1.13 Pseudo-R^2 for different data treatments and transformations

	Pseudo-R^2
Original data	22.2%
Winsorized at 1%	25.5%
Winsorized at 1% + log of ME/TL	34.0%
Original but log of ME/TL	34.3%
Transformation based on default rates	47.8%

The number of ranges that we choose will depend on the size of the data set and the average default rate. For a given number of ranges, the precision with which we can measure their default rates will tend to increase with the number of defaults contained in the data set. For large data sets, we might end up choosing 50 ranges while smaller ones may require only 10 or less.

Note that the transformation also deals with outliers. If we choose M ranges, the distribution of a variable beyond its $(100/M)$th and $(100-100/M)$th percentiles does not matter. As in the case of outlier treatments, we should also be aware of potential data-mining problems. The transformation introduces a data-driven flexibility in our analysis, so we may end up fitting the data without really explaining the underlying default probabilities. The higher the number of ranges, the more careful we should be about this.

CONCLUDING REMARKS

In this chapter, we addressed several steps in building a scoring model. The order in which we presented them was chosen for reasons of exposition; it is not necessarily the order in which we would approach a problem. A possible frame for building a model might look like this:

1. From economic reasoning, compile a set of variables that you believe to capture factors that might be relevant for default prediction. To give an example: the Factor 'Profitability' might be captured by EBIT/TA, EBITDA/TA, or Net Income/Equity.
2. Examine the univariate distribution of these variables (skewness, kurtosis, quantiles…) and their univariate relationship to default rates.
3. From step 2 determine whether there is a need to treat outliers and non-linear functional forms. If yes, choose one or several ways of treating them (winsorization, transformation to default rates,…).
4. Based on steps 1 to 3, run regressions in which each of the factors you believe to be relevant is represented by at least one variable. To select just one variable out of a group that represents the same factor, first consider the one with the highest Pseudo-R^2 in univariate logit regressions.[9] Run regressions with the original data and with the treatments applied in step 3 to see what differences they make.
5. Rerun the regression with insignificant variables from step 4 removed; test the joint significance of the removed variables.

[9] For each variable, run a univariate logit regression in which default is explained by only this variable; the Pseudo-R^2's from these regressions give a good indication on the relative explanatory power of individual variables.

Of course, there is more to model building than going through a small number of steps. Having finished step 5, we may want to fine tune some decisions that were made in between (e.g. the way in which a variable was defined). We may also reconsider major decisions (like the treatment of outliers). In the end, model building is as much an art as a science.

NOTES AND LITERATURE

In the econometrics literature, the Logit models we looked at are subsumed under the heading of 'binary response or qualitative response models'. Statisticians, on the other hand, often speak of generalized linear models. Expositions can be found in most econometrics textbooks, e.g. Greene, W.H., 2003, *Econometric Analysis*, Prentice Hall. For corrections when the sample's mean probability of default differs from the population's expected average default probability see Anderson, J.A., 1972, Separate sample logistic discrimination, *Biometrika* 59, 19–35 and Scott, A.J. and Wild, C.J., 1997, Fitting regression models to case-control data by maximum likelihood, *Biometrika* 84, 57–71.

For detailed descriptions of scoring models developed by a rating agency see: Falkenstein, E., 2000, *RiskCalc for Private Companies. Moody' Default Model*. Moody's Investor Service; Sobehart, J., Stein, R., Mikityanskaya, V. and Li, L., 2000, *Moody's Public Firm Risk Model*: *A Hybrid Approach to Modeling Short-Term Default Risk*. Moody's Investor Service; Dwyer, D., Kocagil, A. and Stein, R., 2004, *Moody's KMV RiskCalc v3.1 model*. Moody's KMV.

Two academic papers that describe the estimation of a logit scoring model are Shumway, T., 2001, Forecasting bankruptcy more accurately: A simple hazard model, *Journal of Business* 74, 101–124 and Altman, E. and Rijken, H., 2004, How rating agencies achieve rating stability, *Journal of Banking and Finance* 28, 2679–2714. Both papers make use of the financial ratios proposed by Altman, E., 1968, Financial ratios, discriminant analysis and the prediction of corporate bankruptcy, *Journal of Finance* 23, 589–609.

APPENDIX

Logit and Probit

We have described the estimation of scoring model with logit. A common alternative choice is the probit model, which replaces the logistic distribution in equation (1.4) by the standard normal distribution. Experience suggests that the choice of the distribution is not crucial in most settings; predicted default probabilities are fairly close. Note, however, that the estimated coefficients differ significantly because the two distributions have different variances. When comparing logit and probit models estimated on the same data set, you should compare default probability estimates or other information which is not affected by scaling.

Marginal effects

Scenario analysis is an intuitive way of understanding the impact of individual variables. An analytical approach would be to calculate the *marginal effect* of a variable. In linear

models the marginal effect is equal to the coefficient. In the logit model, however, life is more difficult. The marginal effect is given by the coefficient multiplied by a scale factor:

$$\text{Marginal effect}_i = \text{Scale factor}_i \times b_i = \Lambda(b'x_i)(1 - \Lambda(b'x_i)) \times b_i \tag{1.24}$$

This scale factor varies with each observation – that is, for each row of our data set we have a different scale factor. To make a statement about average marginal effects, we can use the mean of the x variables to calculate (1.24). Alternatively, we can calculate the scale factor for every observation and then take the average of that.