Getting Started with Credit Scoring

Introduction

Tools and Techniques for Building and Improving Quantitative Risk Ranking Models When Historic Data Is Limited

This technical guide (referred to hereafter as the Guide) provides detailed, step-by-step guidance on how banks with limited historical portfolio data can use simple analytical tools and methods to develop, test and improve their own credit scoring models. To illustrate the concepts in the context of Kenya, the note includes ‘examples from practice’ taken from FSD Kenya’s recent work developing an SME scorecard with a financial institution.

This Guide suggests that practical, useful credit scoring tools do not necessarily require extensive historical data and statistical expertise, but instead require a few simple building blocks:

1. Some understanding of borrowers in the segment for which a credit scoring model is to be used—particularly what borrower characteristics are associated with the risks of not repaying the loan.
2. Some electronic data on past or current clients or the willingness and diligence to systematically collect such data.
3. A software that can organise client data for analysis of cross-tabulations, or counts of the performing and non-performing loans across various borrower characteristics (such as years in business, credit history, financial ratios, etc.).

About GrowthCap

Over the past few years FSDK has been at the forefront of SME banking development through conducting market assessments and studies in areas such as trade finance and SME equity funds, as well as supporting development of the credit reference bureau. Through its partnerships with its Action Research Partners (ARPs), FSDK’s GrowthCap initiative is supporting adoption of SME best practices by individual financial service providers.

This paper is part of a series of Technical Notes and Resource kits that are being developed out of work with the ARPs. These provide detailed information about the best practices and are intended for use by financial service providers and those supporting such institutions which are entering the SME market.

Abstract

This Technical Guide provides detailed, step-by-step guidance on how banks with limited historical portfolio data can use simple analytical tools and methods to develop, test and improve their own credit scoring models.
Getting started with credit scoring

The more a new applicant resembles past clients who did not repay loans, the higher the perceived ‘risk’ of a new loan to that applicant.

Perhaps most important for the successful, long-term use of credit scoring are:

1) The desire to introduce an efficient, quantitative and measurement-based decision making culture for high-volume and lower-amount loan products.

2) The commitment to the ongoing collection and analysis of client data.

Furthermore, at least one person in the bank should be identified (or hired) as the champion for this work and given, as their primary job responsibility, the often tedious task of collecting, preparing and analysing credit risk data. This job is not particularly exciting — it requires perseverance, patience and great attention to detail. However, in practice, this work is critical. It is the quality of client data, and not the mathematical sophistication of its analysis, that drives a scoring model’s predictive power.

In the following sections, this Guide looks at, in turn: 1) the importance of understanding borrowers in the segment scoring will be used for; 2) collection and analysis of data; and 3) building and using statistical credit scoring tools with one key analytical tool — the cross tabulation or contingency table.

UNDERSTAND BORROWERS IN THE TARGET SEGMENT

Each bank needs to understand the clients to whom it sells its financial products. Bankers learn through their own experience and the experience of more senior colleagues to associate various client characteristics and behaviours (such as past credit history, account turnover, balance sheet strength, years of experience in business, etc.) with levels of comfort or confidence that the client will repay the loan.

‘Risk’ in credit scoring is the failure to repay the loan. The more a new applicant resembles past clients who did not repay loans, the higher the perceived ‘risk’ of a new loan to that applicant. Lenders apply such thinking processes in most traditional lending processes. Credit scoring goes further to assign numeric point scores to different types of borrower characteristics. Summing these point scores for a set of borrower characteristics results in a total ‘credit score’ by which clients can be rank ordered in terms of perceived riskiness.

For a bank new to credit scoring, a model is not likely to instantly provide extensive or completely new insights to bankers. Instead, it should help bankers to systematically collect and store electronically what they believe to be the most relevant client data, analyse that data consistently and efficiently over time, and lead to a quantitative and therefore more explicit understanding of how different borrower characteristics are associated with risk. If a bank does not have historical data, it can start using credit scoring conservatively by creating an ‘expert scorecard’ as described in the example from the Practice 1 Box below and in more detail, in the paper “Building Credit Scorecards for Small Business Lending in Developing Markets” (Caire 2004).
Example from Practice 1: Expert scorecard development

In a leading Kenyan commercial bank, a team of experienced SME lenders formed a working group to develop a scorecard without reference to historic data—what is often called an ‘expert’ or ‘judgmental’ scorecard.

The group listed a number of borrower characteristics (further called risk factors) they believed to be associated with credit risk, such as ‘years in operation’, ‘credit history’ and ‘current ratio’. For each risk factor they used their collective expertise to assign points to groups of borrowers where more points indicate lower risk. For example, for the risk factor ‘years in operation’, the working group agreed that risk decreases as years in operation increase. They defined 5 risk groups (less than 1 year, 1–2 years, 3–5 years, 6–10 years and more than 10 years) and assigned points as shown in the table below.

<table>
<thead>
<tr>
<th>Years in operation</th>
<th>Expert scorecard points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 1 year</td>
<td>1</td>
</tr>
<tr>
<td>1 – 2 years</td>
<td>4</td>
</tr>
<tr>
<td>3 – 5 years</td>
<td>6</td>
</tr>
<tr>
<td>6 – 10 years</td>
<td>8</td>
</tr>
</tbody>
</table>

For expert scorecards, it is important to define risk relationships that are simple, straightforward and make sense. Correctly ranking perceived risk for each factor in the model and choosing a set of factors that together create a comprehensive risk profile of the a borrower (along the lines of the popular ‘5 Cs of credit: capacity, capital, collateral, conditions and character) should result in a useful tool that can rank order borrowers by risk in general agreement with an experienced lender.

Once an expert scorecard is built by a working group, it can be tested even if the bank has no historic data for the segment. A good way to do this is to score either a sample of recent clients or all new incoming applications with the expert scorecard in parallel to a standard lending procedure. This is sometimes referred to as ‘pilot’ or ‘parallel’ testing. In such a testing process, the loan officer processing the loan or the credit analyst reviewing it should be asked (without reference to the model) to rank the client on a 5-point scale with ‘very low risk’ and ‘very high risk’ at the extremes and ‘average risk’ in the middle, as shown below.

An expert scorecard is ‘validated’ in practice when its rankings based on point scores generally agree with the subjective risk rankings of experienced lenders.

By contrast, if experienced lender assessments frequently disagree with a scorecard’s risk ranking, there is an opportunity to adjust the expert model to reflect particular low or high risk elements it does not currently capture. Often this is done through the introduction of ‘stop factors’ which are not present for the majority of applicants, but signal particularly high risk when they are. Some examples of such stop factors that are commonly found in scorecards are:

- Less than 1 year in business
- Arrears of more than 30 days on currently outstanding loans
- Arrears of more than 90 days on any loans over the past two years
- Any negative listings on a credit bureau report

Each stop factor should be linked to a business rule or policy decision. With reference to the four bullet points above, arrears of more than 90 days or negative credit bureau listings could be grounds for a hard policy decline of the credit application while less than 1 year in business or current arrears exceeding 30 days may call for a different business process or case-by-case review by a particular committee.

Once an expert scorecard has been tested to establish a comfort level in its risk ranking ability, it can be used in the lending process to rank risk and support consistent decision making. It is crucial at this point to also put into place a process to capture all of the data gathered. This point is valid whether the scorecard is implemented in an application processing system or in an Excel template. In fact, it can often be beneficial to briefly test an expert scorecard operationally in Excel before developing
Getting started with credit scoring

COLLECT AND ANALYSE PAST DATA

To move from bankers’ best estimates of risk ranking, based on past personal experience, to quantitative relationships between past borrower characteristics and loan repayment, it is necessary to systematically collect and analyse borrower data over time. For banks that are not already using ‘application processing software’ to capture all of the borrower information collected and analysed in the lending process, the introduction of an expert credit scorecard can prove very useful. In addition to providing a consistent quantitative risk estimate that can be used to support lending decisions, introducing such a scorecard system often catalyses the systematic collection of data for future analysis.

Most banks already have extensive client credit risk data stored electronically. Unfortunately, a lot of it is not gathered in a centralised database. Quite commonly in practice, Excel-based analytical files and Word-based loan memorandum are spread throughout the organisation on the desktops and laptops of loan officers and credit analysts. However, it is a relatively small piece of work to gather these files, extract their data programmatically and thus build a historic database of credit risk indicators. The lowest tech solution is to assign junior staff person to open each file and copy and paste the necessary cells to one central database—not at all glamorous or exciting, but an opportunity to mine meaningful historical data in a situation where the information has been gathered but not aggregated. A more efficient way to do the same thing is to ask an IT person to write a small programme that opens each file in a specified directory, extracts the relevant data and writes it to one database.

Sometimes even the past analytical templates may not contain all of the information needed for the risk modelling. In this case, another way of aggregating the data for analysis is to review hard copy credit files and key in the necessary data into an Excel file containing the expert scorecard. Such a test is sometimes referred to as a ‘back-test’. Back-testing, like pilot-testing, is another way of seeing how well a scorecard’s rankings agree with expert judgment or actual loan performance. As shown in the Example from Practice 2 Box if it is possible to select a batch of past clients made up of both good clients and clients with repayment problems, a back-test can show how well the expert scorecard ranks risk (more of the problem loans should be concentrated in the higher risk score ranges).

Example from Practice 2: Back-testing with historic data

In 2015, a bank back-tested its recently developed expert scorecard for the SME segment. Credit analysts and loan officers reviewed hard-copy credit files for 81 loans originated in 2012 and entered data from the original time of application into an Excel-based tool to score the loans. Of these 81 past clients, 45 were ‘good’ (performing) and 36 were ‘bad’ (non-performing). This ‘back-test’ data allowed the team both to test the expert scorecard’s overall risk ranking ability and to see if the risk relationships expected by the group of experts were evident in the distribution of bad loans in the data.

In this case, 81 total loans is clearly a very small number of loans. Generally, the more historic data that is gathered and analysed, the more confident we can be that patterns in the data are meaningful, or not mainly due to chance. Yet analysing 81 loans is also magnitudes better than analysing no loans (or waiting indefinitely while claiming that credit scoring is not possible until more data is gathered). In this example, the 36 ‘bad’ loans were the entire population of ‘bad’ loans in 2012 (and thus certainly representative of their ‘bad’ clients that year) and there were a similar number of good loans.

Once client data is in a table format, such as an Excel file, with each row a different loan and the data fields to analyse in the columns, there is one other crucial step to take prior to beginning the credit scoring analysis. A new column should be added to the data and named something like ‘Loan Status’. For each loan, it should be classified as ‘good’ or ‘bad’ based on a definition chosen by the bank. A ‘bad’ loan is one that given perfect hindsight the bank would have chosen not to make. A widespread definition of ‘bad’ SME loans for credit scoring is ‘90 or more days in arrears’, but for smaller loans (such as ‘micro’ loans), often more than 30 consecutive days is already ‘bad’. The definition does not need to be number of days — it can also be number of times in arrears, or a combination of days and times — but like most of the modelling process, it is usually a good idea to keep things relatively simple. If the bank has no particular view for a given product, 90 consecutive days in arrears should work well (and is consistent with the Basel Accord risk framework). \(^1\)

1. The Basel Committee on Banking Supervision provides a forum for regular cooperation on banking supervisory matters. Please refer to http://www.bis.org/bcbs/
In summary, a pragmatic approach to kick-starting scoring with limited data is to:

1. Use an expert model and test it against the judgment of experienced analysts;
2. Use as much historic data as possible to back-test the model and examine which expected risk relationships in the expert model are supported by the historic data;
3. Don't worry too much about statistical concepts, but use the cross tabulations explained in the next section to focus on identifying patterns that confirm expected relationships between credit risk and borrower characteristics.

**CROSS-TABULATION ANALYSIS AND MODEL BUILDING**

Understanding risk relationships in past performance data is no more complicated than grouping data according to types or levels of borrower characteristics and counting the number of good loans and number of bad loans in each group. The more bad loans as a share of total loans in a group, the more risk.

The cross-tabulation or contingency table, explained in the Example from Practice 3 Box is a simple analytical tool that can be used to build and manage credit scorecards. Step-by-step detailed instructions for how to create cross tabulations in Excel or the open-source ‘R’ statistical software are provided in Annex A: Tools for Cross-Tabulation Analysis.

**Example from Practice 3: Cross-tabulation analysis**

Cross tabulation 1 (or CT1) shows the distribution of 45 good and 36 bad loans from the back-test of 81 past loans.

<table>
<thead>
<tr>
<th>Table CT.1 – Years in operation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row</strong></td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>

Years in operation are divided into 3 groups, less than 6 years, 6-10 years, and more than 10 years. In each group, Table CT.1 presents for each group:

- Row A: the number of good contracts
- Row B: the number of good contracts (row A)/number of total contracts (Row E)
- Row C: the number of bad contracts
- Row D: the number of bad contracts (row C)/number of total contracts (Row E)
- Row E: the total number of contracts
- Row F: the number of total contracts in the group (column) divided by all contracts

The ‘key’ row for credit scoring analysis is row D, or the percentage of bad loans for a group. This percentage (or ‘bad rate’) should rise or fall as we move across the risk categories. Referring back to Example from Practice 1 Box, the expert scorecard expected risk to go down as years in operation increased. The back-testing data confirmed this relationship: 62% of contracts (or 21 (row C) out of 34 (row E), where 21/34 = 62%) were ‘bad’ for companies with less than 6 years in operation, while 47% (7 bad/15 total) were ‘bad’ for companies with 6-10 years, and only 25% (8 bad/32 total) were ‘bad’ for companies operating 10 years or more.
The main work in building a credit scoring model is in understanding the risk relationships between each individual risk factor and loan repayment. Cross-tabulations are one of the best tools to facilitate this analysis. Cross-tabulations also help to put the borrowers into groups with similar risk levels per factor. Grouping (sometimes called ‘binning’) the risk factors in a model is a popular technique in credit scoring because it helps to avoid over-fitting the model to the data set being analysed and avoids some other potential technical problems.

The following guiding principles should be used when grouping variables with reference to historic data:

- Create groups to maximise the differences in the bad rates between groups while distributing the population evenly between the groups. In other words, the greater the differences in bad rates between groups and the more evenly loans are distributed across the groups, the better a given risk factor will rank risk.
- Make sure any relationships identified are easy to explain. For most risk factors, the bad rate either steadily rises or steadily falls without reversals. If there are reversals at any point in the relationship, they should make sense to the working group and be explained in the model documentation. If not, groups should be combined or group boundaries should be re-defined to eliminate the reversal.
- Make sure the working group can explain each relationship in the model. If a relationship conflicts with common sense, it is better to leave it out and use another risk factor in the model while continuing to analyse the factor with the unexpected relationship in order to better understand why the risk trend is different from expectations.

**Combining factors in a risk model**

Once all potential risk factors have been analysed, the analyst should pick a combination of factors that together provide a comprehensive risk profile of the borrower that represents as many independent types of information as possible. A comprehensive risk profile makes the scorecard more stable and less susceptible to changes in one particular area. (Siddiqui, 2006).

For a statistical scorecard, just as for expert scorecards, it is necessary to assign points for each risk factor. In section one on expert scorecards, arbitrarily assigned increasing positive points represented decreasing risk. When historic data is analysed with cross tabulations, a model will rank risk better if the factor points are set with direct reference to the bad rates (total number of ‘bads’ divided by total number of contracts) for the characteristics and their groups.

In banks that use scoring extensively and employ whole departments of analysts to manage their models, logistic regression is most commonly used to develop the statistical model. There are various explanations in the literature of how to turn logistic regression coefficients into positive point weights (see Mays, 2011 and Siddiqi, 2006), but they involve several mathematical transformations of the bad rate information in the data to arrive at point scores.

A more straightforward and transparent way to base the point weights directly on bad rates in the data is to just use the bad rate % numbers as the points numbers. For example, in the cross-tabulation example that has been used throughout this Guide, the points that would be assigned for ‘Years in Operation’ are shown in row P of Table CT2—namely 62 points for less than 6 years in business, 47 points for 6 to 10 years in business, and 25 points for more than 10 years in business.

<table>
<thead>
<tr>
<th>Table CT.2 – Years in operation – bad rate points</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Row</strong></td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>

Using only the bad rates as points and adding the points for each factor to arrive at a total score (where higher points indicate higher risk) will for many data sets result in overall risk ranking results that are very similar to a more complex logistic regression model (Caire and Schreiner, 2013). However, most people expect higher points to indicate lower risk—this is the convention in the best known credit scoring models of major vendors and in general is the convention for most tests and scores people take throughout their lives—higher scores are ‘better’.
An alternative method for capturing the same information in the bad rates and converting it into positive points is to assign points equal to the differences between bad rates in the various groups. In Table CT.3, the points in column P are the differences (expressed in whole numbers) between the maximum bad rate of 62% (for less than 6 years in operation) and the other groups’ bad rates. So the 15 points for 6 to 10 years represents the difference between 62% and 47%.

<table>
<thead>
<tr>
<th>Row</th>
<th>Group</th>
<th>&lt;6 years</th>
<th>6-10 years</th>
<th>&gt;10 years</th>
<th>Row total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Good</td>
<td>13</td>
<td>8</td>
<td>24</td>
<td>45</td>
</tr>
<tr>
<td>B</td>
<td>% Good</td>
<td>38%</td>
<td>53%</td>
<td>75%</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Bad</td>
<td>21</td>
<td>7</td>
<td>8</td>
<td>36</td>
</tr>
<tr>
<td>D</td>
<td>% Bad</td>
<td>62%</td>
<td>47%</td>
<td>25%</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>Points</td>
<td>0</td>
<td>15</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>Total</td>
<td>34</td>
<td>15</td>
<td>32</td>
<td>81</td>
</tr>
<tr>
<td>F</td>
<td>% of Total</td>
<td>42%</td>
<td>19%</td>
<td>40%</td>
<td>100%</td>
</tr>
</tbody>
</table>

The scores in Table CT.3 have the more intuitive quality of assigning more points for lower risk. Such point weights can be put on any desired point scale (such as zero to 800, which is common to the scoring industry) by multiplying the differences in the bad rate by a ‘scaler’ or whole number constant.

**Example from Practice 4: Building the Statistical Scorecard**

The expert scorecard developed by the bank in our example had 48 expert-weighted risk factors. The back testing data on 81 past borrowers confirmed the risk relationships expected by the experts for 13 of the factors. In each case, the trend was the same as expected by the experts, but the relationship was simpler—in other words, based on differences in bad rates, there were fewer risk groups per risk factor.

Analysing how an expert scorecard performs on examining an expert scorecard’s performance on actual data often creates the opportunity to improve the model’s risk ranking by using fewer total risk indicators with simpler risk ranking patterns (or fewer groups) than initially perceived by experts. This is because some of the relationships initially expected by experts are not proven in practice, and sometimes are even the reverse of what was expected. In such cases, keeping the factors in the model interferes with risk ranking, and removing them can improve it.

The best tool to summarise the risk ranking ability of a credit scoring model and also to set policy around it is also a cross-tabulation—of total scores and past ‘good’ and ‘bad’ loans.

The ‘Policy Table’ presented in CT.4 is taken from a different bank that developed an expert scorecard for SMEs in 2014 and back-tested it with 1,200 SME loans originated in 2012.

The cross-tabulation groups the loans into 10 risk groups, where group 1 is the lowest risk group and group 10 is the highest risk based on total score from a 13-factor statistical model (note this model also reduced a 33-factor expert model to 13 statistically weighted factors, greatly improving risk ranking in the process). The score ranges for the risk groups in column A are shown in columns B and C, and the number of good and bad loans in each group are in columns D and E, respectively.

As the back-test in this bank was also designed to maximise the collection of data on past ‘bad loans, they were significantly overrepresented (at 357/1,200 = ~30%) relative to the percentage of such loans in the outstanding portfolio (estimated to be around 5%). To facilitate decision policy formulation, the Table CT.5 grosses-up the total number of good loans to 6,782 (in column G) with the same percentage of total goods per risk group in order to estimate a more realistic probability of default estimates (in column H) for future clients.

The decision policy recommended for this bank’s pilot test of the scoring model is to auto-approve loans in risk groups 1 to 7 (scoring 300 points or above), review loans in risk group 8 in a branch-based or central risk committee, and reject risk groups 9 and 10.
Table CT.4: Recommended Decision Policy Table by Risk Group

<table>
<thead>
<tr>
<th>RISK GROUP</th>
<th>SCORE RANGE</th>
<th>BACK TEST</th>
<th>Per 5% NPR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt;=</td>
<td>&lt;</td>
<td>Goods</td>
</tr>
<tr>
<td>1</td>
<td>600</td>
<td>29</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>550</td>
<td>115</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>475</td>
<td>124</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>425</td>
<td>128</td>
<td>22</td>
</tr>
<tr>
<td>5</td>
<td>385</td>
<td>104</td>
<td>23</td>
</tr>
<tr>
<td>6</td>
<td>345</td>
<td>111</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>300</td>
<td>117</td>
<td>56</td>
</tr>
<tr>
<td>8</td>
<td>225</td>
<td>89</td>
<td>118</td>
</tr>
<tr>
<td>9</td>
<td>150</td>
<td>22</td>
<td>71</td>
</tr>
<tr>
<td>10</td>
<td>150</td>
<td>4</td>
<td>24</td>
</tr>
<tr>
<td>TOTAL</td>
<td>843</td>
<td>357</td>
<td>1,200</td>
</tr>
</tbody>
</table>

Table CT.5 shows what the bank could expect in terms of acceptance/rejection rates and portfolio bad rates given the recommended decision policy. Based on the model and assuming that decisions are: 1) taken only based on the information reflected in the risk model; and 2) the back-test sample is representative of future target clients in the segment, the bank can expect with a policy of approving loans in risk groups 1 to 7 to have:

1. An acceptance rate of 84% of all clients (column M)
2. A portfolio 'bad' or non-performing loan rate of 2.4% in terms of number of loans (column K).

By rejecting loans in risk groups 9 and 10, the bank can avoid 27% of its past bad clients and lose only 3% of its good clients.

Table CT.5: Acceptance Rate and Bad-Rate by Risk Group

<table>
<thead>
<tr>
<th>RISK GROUP</th>
<th>Adj. Goods</th>
<th>Bads</th>
<th>Total</th>
<th>PD est- mate</th>
<th>POLICY</th>
<th>% of Goods</th>
<th>Cum % Goods</th>
<th>% of Bads</th>
<th>Cum % Bads</th>
<th>Bad Rate at Cut-off</th>
<th>% Bads Avoided at Cut-Off</th>
<th>Acceptance Rate At Cut-Off</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>233</td>
<td>0</td>
<td>233</td>
<td>0%</td>
<td>APPROVE</td>
<td>3%</td>
<td>3%</td>
<td>0%</td>
<td>0%</td>
<td>0.00%</td>
<td>100%</td>
<td>3%</td>
</tr>
<tr>
<td>2</td>
<td>925</td>
<td>4</td>
<td>929</td>
<td>0%</td>
<td>APPROVE</td>
<td>14%</td>
<td>17%</td>
<td>1%</td>
<td>1%</td>
<td>0.34%</td>
<td>99%</td>
<td>16%</td>
</tr>
<tr>
<td>3</td>
<td>998</td>
<td>14</td>
<td>1,012</td>
<td>1%</td>
<td>APPROVE</td>
<td>15%</td>
<td>32%</td>
<td>4%</td>
<td>5%</td>
<td>0.83%</td>
<td>95%</td>
<td>30%</td>
</tr>
<tr>
<td>4</td>
<td>1,030</td>
<td>22</td>
<td>1,052</td>
<td>2%</td>
<td>APPROVE</td>
<td>15%</td>
<td>47%</td>
<td>6%</td>
<td>11%</td>
<td>1.24%</td>
<td>89%</td>
<td>45%</td>
</tr>
<tr>
<td>5</td>
<td>837</td>
<td>23</td>
<td>860</td>
<td>3%</td>
<td>APPROVE</td>
<td>12%</td>
<td>59%</td>
<td>6%</td>
<td>18%</td>
<td>1.54%</td>
<td>82%</td>
<td>57%</td>
</tr>
<tr>
<td>6</td>
<td>893</td>
<td>25</td>
<td>918</td>
<td>3%</td>
<td>APPROVE</td>
<td>13%</td>
<td>72%</td>
<td>7%</td>
<td>25%</td>
<td>1.76%</td>
<td>75%</td>
<td>70%</td>
</tr>
<tr>
<td>7</td>
<td>941</td>
<td>56</td>
<td>997</td>
<td>6%</td>
<td>APPROVE</td>
<td>14%</td>
<td>86%</td>
<td>16%</td>
<td>40%</td>
<td>2.40%</td>
<td>60%</td>
<td>84%</td>
</tr>
<tr>
<td>8</td>
<td>716</td>
<td>118</td>
<td>834</td>
<td>14%</td>
<td>REVIEW</td>
<td>11%</td>
<td>97%</td>
<td>33%</td>
<td>73%</td>
<td>3.83%</td>
<td>27%</td>
<td>96%</td>
</tr>
<tr>
<td>9</td>
<td>177</td>
<td>71</td>
<td>248</td>
<td>29%</td>
<td>REJECT</td>
<td>3%</td>
<td>100%</td>
<td>20%</td>
<td>93%</td>
<td>4.70%</td>
<td>7%</td>
<td>99%</td>
</tr>
<tr>
<td>10</td>
<td>32</td>
<td>24</td>
<td>56</td>
<td>43%</td>
<td>REJECT</td>
<td>0%</td>
<td>100%</td>
<td>7%</td>
<td>100%</td>
<td>5.00%</td>
<td>0%</td>
<td>100%</td>
</tr>
<tr>
<td>TOTAL</td>
<td>6,782</td>
<td>357</td>
<td>7,139</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CONCLUSION

This Guide has sought to explain, with as little jargon as possible, how a bank with limited historic data can introduce credit scoring by understanding its borrowers, collecting and analysing what historic (or current) data it does have and then building the best model it can with either expertise, historic data, or most likely some of both. The cross-tabulation, or contingency table, is presented as the single most useful analytical tool used throughout this process, and remains the key tool needed to manage scorecards and improve their risk ranking in the future.

In fact, once any scorecard has been developed, whether fully in-house or with the assistance of a third party, the long-term success of a credit scoring project will depend not only on how well the scorecard ranks risk or estimates a given applicant’s probability of default (PD), but will also be a function of some or all of the following factors:

- Scoring’s role in the business process, and the business process itself;
- Software used to implement and administer the scorecard, including its links to other process-management software;
- Training, support, and communication with “front-line” users;
- “Ownership” of the scorecard by sufficiently senior people in the organisation;
- Regular monitoring of scorecard performance, along with readiness to adjust or re-develop the scorecard as appropriate;
- Clear documentation of scorecard development and the scorecard validation process (Caire 2004).

Hopefully this Guide can serve as a useful introduction to the topic for banks new to credit scoring, and for a handy and practical reference for banks with some experience. The books and articles included in the references section at the very end of this document are also a good place to start further reading on the topic of credit scoring.

REFERENCES


Mark Schreiner. Scoring: the Next Breakthrough in Microcredit? http://www.microfinance.com/English/Papers/Scoring_Breakthrough_CGAPpdf

ANNEX A: TOOLS FOR CROSS-TABULATION ANALYSIS

Introduction
This annex is meant to be used as a step-by-step guide and reference while working at the computer with the programmes mentioned. It explains how to create and analyse cross tabulations in Excel and the ‘R’ software.

The cross-tabulation, or contingency-table is a simple analytical tool that can be created with a pivot table in Excel, manually in Excel (using the COUNTIFS function in version 2007 or higher is helpful for this), or using any statistical analysis software.

METHOD 1: CROSS-TABULATIONS USING EXCEL PIVOT TABLES

A 3-minute video on the internet (https://www.screenr.com/uuv8) explains how to generate cross-tabulations using Excel Pivot Tables. The below ‘offline’ example uses the data for the 81 back-test cases from a real example and walks through 7 steps to create a cross-tabulation of the Years in Operation as it was originally grouped in the expert scorecard.

1. Make sure the data you are analysing is in an Excel worksheet where the first row is a header and one column contains loan status coded 0 (for good) and 1 (for bad). The example in the screen shots will contain 81 rows of data, but only three columns – loan status, years in operation points and years in operation.

2. Select the entire range containing the data as shown in the Screen Shot 1:

3. Press the ‘Pivot Table’ button on the INSERT tab. Notice in Screen Shot 2 that the range of cells from A1 to C82 is selected and that the Pivot Table will be created on a new worksheet. Click ‘OK’.
4. A new worksheet is inserted and the Pivot Table Wizard view appears, the right half of which is shown in Screen Shot 3.
Drag ‘loan status’ into the ‘ROWS’ quadrant and into the ‘values’ quadrant. Drag ‘years in operation points’ in to the columns quadrant. The result is shown in Screen Shot 4.

**Screen Shot 4**

Now, in the VALUES quadrant, click on Sum of Loan’ and select ‘Value Field Settings’, as shown in Screen Shot 5.

**Screen Shot 5**
6. Change the selection from ‘Sum’ to ‘Count’ by selecting Count, as shown in Screen Shot 6, and press ‘OK’.

The Pivot Table created by Excel will now look like Screen Shot 7 where the two categories ‘0’ (for good loans) and ‘1’ (for bad Loans) are shown in column A and in columns B to F the count of good loans for each possible point score (1, 4, 6, 8 or 10) are shown in in row 5, while the count of ‘bad’ loans is shown in row 6.
Recall, the expert point scores for this factor were:

<table>
<thead>
<tr>
<th>Years in operation</th>
<th>Expert scorecard points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 1 Year</td>
<td>1</td>
</tr>
<tr>
<td>1 – 2 Years</td>
<td>4</td>
</tr>
<tr>
<td>3 – 5 Years</td>
<td>6</td>
</tr>
<tr>
<td>6 – 10 Years</td>
<td>8</td>
</tr>
<tr>
<td>More than 10 Years</td>
<td>10</td>
</tr>
</tbody>
</table>

7. In order to view the share, or percentage of bads in the Pivot Table, right click inside the Pivot Table and select ‘Show Values As’ and then ‘% of Column Values’, as shown in screen shot 8.

The resulting table is shown below in screen shot 9, where the ‘bad rate’, the measure or ‘risk’ is now shown in row 6. From column D to column F we can see that the ‘bad rate’ in row 6 steadily decreases from 65% for loans that scored 6 points (3–5 years in Business), to 55% for loans scoring 8 points (6–10 years in business) to only 25% for loans scoring 10 points (more than 10 years in business).

In summary, the expected relationship, that risk decreases as Years in Operation increases is evidenced by the 81 loans in the back test. The expected pattern is not
evidenced for the loans scoring 1 (less than one year in business) or 4 points (1-2 years in business).

However, looking at the earlier screen shot with ‘counts’ of ‘good’ and ‘bad’ loans, it is clear that only 1 loan (a ‘good’ one) in the back-test had less than 1 year in operation and only 5 were 1-2 years in business. If these 6 loans are added into the category scoring 6 points (by replacing the values ‘1’ and ‘4’ in the relevant column with the value ‘6’) and we change the name of the first category to ‘Loans less than 5 years’, the expected pattern holds in the data (with bad rates of 58%, 55% and 25%).

Pivot Tables in Excel are handy in that with a little practice they are easy to make and require no additional ‘programming’, however they are not the most appropriate presentation format; because it useful to look not only at the ‘bad rate’, but also at the percentage of total observations in each group. The next method, of creating cross-tabulations using formulae in Excel addresses this issue, but requires additional work.

**METHOD 2: CREATING ‘HAND-MADE’ CROSS-TABLES IN EXCEL USING THE “COUNTIFS” FUNCTION**

The COUNTIFS function can count frequencies for more than one logical argument. For example in credit scoring, it is necessary to count cases where the Loan Status column equals ‘1’ and where the other variable of interest, such as years in operation points takes a specified value range.

Screen shot 10 shows three columns of data: Loan status of 0 or 1 in column a, years in operation points in column b and years in operation in column c.

The cross table in columns E to I follows the presentation format of Table CT.1 shown earlier. It is created using the COUNTIFS function, with the Excel formulae rather than the results displayed. For example, to count the number of good loans that scored 10 points (and thus were more than 10 years in operation), the formula is:

\[=\text{COUNTIFS}(A:A,0,B:B,10)\]

This counts the number of contracts that have 0 in column A and ‘10’ in column B of the data set. In the screen shot dollar signs are added to the references to the columns A and B (this can be done with the keyboard shortcut F4) which enables the formulae to be copied across columns whilst references to the columns A and B remain fixed.

To make the same counts with reference to column C, the final argument should be stated as a logical argument and put in quotation marks, as follows:

\[=\text{COUNTIFS}(A:A,0,C:C,”>10”)\]

Looking at the results rather than the formulae, the hand-made cross tabulation shown below as Table CT.1 presents the same results described in the previous section on Pivot Tables, but now shows the column-wise percentage of good and bad loans and shows the row-wise % of total contracts (which sum to 100% in the far right column). Thus, the advantage of hand-making a table is that it is possible to control the format of presentation without relying on features in the Excel Pivot Table wizard. On the other hand, it requires a little more tedious work by the analyst. For the minor investment of learning to bring Excel data into the open source statistical analysis software ‘R’, cross-tabulations can be created easily by editing the name of the desired analysis variable in a line of code as is illustrated in the next section.
Table CT.1 – Years in Operation

<table>
<thead>
<tr>
<th>Category</th>
<th>5 years or less</th>
<th>6-10 years</th>
<th>More than 10 years</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Goods</td>
<td>12</td>
<td>9</td>
<td>24</td>
<td>45</td>
</tr>
<tr>
<td>% of Goods</td>
<td>41.4%</td>
<td>45.0%</td>
<td>75.0%</td>
<td></td>
</tr>
<tr>
<td>Number of Bads</td>
<td>17</td>
<td>11</td>
<td>8</td>
<td>36</td>
</tr>
<tr>
<td>% of Bads</td>
<td>58.6%</td>
<td>55.0%</td>
<td>25.0%</td>
<td></td>
</tr>
<tr>
<td>Total Contracts</td>
<td>29</td>
<td>20</td>
<td>32</td>
<td>81</td>
</tr>
<tr>
<td>% of Total</td>
<td>35.8%</td>
<td>24.7%</td>
<td>39.5%</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

**METHOD 3: THE CROSSTABLE FUNCTION IN ‘R’ SOFTWARE**

The advantage of performing cross-tabulation analysis using ‘coding’, or, in other words, by writing a programme, is that the analysis is then documented and can be repeated at any time by ‘running’ the programme. This means if a mistake is found, it can be corrected in the programme and the analysis can be re-generated in moments. By contrast, if a large set of data is analysed ‘by hand’ in Excel and a mistake is found (and finding the mistake will also likely be more difficult), the analyst may need to re-do that analysis starting from the point of the mistake in the original data file (if a unique version had been saved at that point). Fortunately, only a few steps and a few lines of code are needed to use the CrossTable function in R.

**Getting familiar with the ‘R’ software**

Some general comments about using the ‘R’ software examples presented below:

- Each example suggests one way to complete a given task. In fact, there are very many possible ways to complete these tasks in R, and the reader is encouraged to use the internet to learn more about R syntax and explore how to do things most efficiently for the analytical task at hand.
- All names of variables and R objects are assigned arbitrarily by the analyst. Certain names are used in the example below, but could be named differently at the discretion of the analyst. What is important is to correctly use R syntax (language) and to correctly name any R functions used.
- A hash mark (#) in R is used to “comment out” or deactivate text — it is used both to leave notes in our programmes that explain the use of different lines of code and to toggle various lines of code “on” or “off” as we analyse our data.

To use R:

I. The ‘R’ software can be downloaded free of charge at the weblink, or by ‘Google’-ing ‘R Software download’: http://cran.r-project.org/bin/windows/base/ (accessed March 31, 2015)

II. After installing ‘R’, open the programme (by double-clicking on the ‘R’ icon on the desktop) and select New script from the File menu, as shown in the Screen Shot 11.
III. In the ‘R’ programme, the programme code, or script, should be written in the ‘R-Editor’ window (on the right hand side of the the Screen Shot 12 below).

Screen Shot 11

Screen Shot 12
Getting started with credit scoring

IV. However, in order to carry out cross-tab analysis in ‘R’, several ‘R packages’ should be first downloaded to its ‘R library’ Select ‘Install package(s)’ from the Packages menu as shown in Screen Shot 13:

Screen Shot 13

V. The ‘R’ programme will require selection of a ‘mirror’ to be selected — the server from where the packages are downloaded to the user’s computer—as shown in screen shot 14. Selecting any location (including ‘0-Cloud’) and pressing ‘OK’ should work fine.

VI. Next, a window listing all possible libraries will appear, as shown in Screen Shot 14. The following packages should be downloaded (one at a time):

- caTools
- gmodels
- Hmisc

Screen Shot 14
Once these libraries have been downloaded to the user's computer, they will be available for all future sessions—in other words, downloading the 'R' software and libraries only needs to be done once.

VII. Finally, before starting to write the programme in R, save the Excel data file as a .txt Tab delimited file by selecting the Excel worksheet containing the data and choosing SaveAs from the file menu in Excel, then choosing the 'Text (Tab delimited) (*.txt)' option.

VIII. Once the data has been saved as a .txt Tab delimited file, return to the R-Editor window in the R programme and type in the following code:

```r
#LOAD R LIBRARIES TO USE IN SESSION
library(gmodels)
library(Hmisc)
library(caTools)
library(base)
```

This code will load the libraries into the active R session. Next, type in

```r
##SET DIRECTORY WHERE DATA FILE IS STORED ON YOUR COMPUTER
```

In Windows Explorer, navigate to the location of the Excel data file that you have saved as a .txt Tab delimited file on your computer. Then right click on the file and select properties, as shown in the Screen Shot 16.
Use Ctrl+C to copy (to the clipboard) the ‘Location’, or path to the .txt file, from the window that appears, as shown in the Screen Shot 17:
Paste the path name into the R-Editor window.

```r
# SET DIRECTORY WHERE DATA FILE IS STORED ON YOUR COMPUTER
C:\Users\Documents\
```

Then, type `setwd()` in front of the path and close the quotes and parentheses at the end of the path for:

```r
setwd("C:\Users\Documents")
```

Next, change the forward slashes in the path name to backward slashes, so the resulting code is:

```r
setwd("C:/Users/Documents/
```

Next, type in:

```r
# READ IN THE DATA FILE AS A 'DATA TABLE'
flat <- read.delim(""
```

Return to the Properties Window in Windows Explorer and copy the name of the .txt file to be analysed (here it is called ‘Bank_data.txt’), as shown in Screen Shot 18.

```
 Screen Shot 18
```

Paste this name in immediately after the open quote mark, type: `”`

so the code reads:

```r
# READ IN THE DATA FILE AS A 'DATA TABLE'
flat <- read.delim("Bank_data.txt")
```
“flat” is the name of a data table object created for analysis. It could in fact be named anything (“round”, “Elton”, etc.). The next line of code, the ‘fix’ function, will display the data table in a tabular format visually similar to a spreadsheet. Type into the R-Editor:

```r
fix(flat)
```

To see the data table (and check for any typing errors made in the code up to this point, it is necessary to execute the code by selecting all of the code in the R editor (see Screen Shot 19).

![Screen Shot 19](image)

and pressing either Ctrl+R or pushing the third icon from the left (of an arrow between two planes) to ‘Run line or selection’ of the code as shown in screen shot 20.

![Screen Shot 20](image)

Running this code will load the named libraries, set the working directory to the folder that contains the .txt (data) file that will be analysed, read that .txt file into an R data table and display that data table in the R ‘Data Editor’, as shown in the Screen Shot 21.

At this point, it is a good idea to scroll down through the data editor and be sure there are 81 lines of data.
Leaving the data editor open, click on the column header reading ‘Years.in.Operation.Points’. The following Variable editor window will appear. Be sure that variables you expect to be numeric are indeed numeric, as otherwise ‘R’ will not be able to perform numeric operations on them. In case there is accidentally text mixed in with numeric values in the original (Excel) data set, it is easiest to open the data in Excel, fix the error there, and then re-save the worksheet as a .txt tab delimited file and import it back into ‘R’. As shown in Screen Shot 22, highlight the variable name and copy it to the clipboard using Ctrl+C.

Place a hash mark (#) on the next free line in the R-editor and paste in the variable name as shown below:

```r
#Years.in.Operation.Points
```

Do the same two steps to copy the name of the Loan Status field into the R-editor:

```r
##Loan.Status
```

The reason to cut and paste rather than re-type the names is to avoid typographic errors, as any slight difference from the actual text as it appears in the data set will cause it to be read incorrectly in ‘R’ and result in an error.

Now type in the following code to use the CrossTable function in R:

```r
CrossTable(flat$, flat$ ,prop.chisq=FALSE,prop.r=FALSE,prop.t=FALSE)
```
After the first ‘flat$’, paste in the name of the Loan.Status variable (without the hash tag), and after the second ‘flat$’ copy and paste the name of of ‘Years.in.Operation.Points’ (also without the hash tag). This ‘R’ naming convention is that the data table is named ‘flat’ and the the ‘$’ precedes the name of the variable in the data table that will be analysed.

The result should be:

CrossTable(flat$Loan.Status,flat$Years.in.Operation.Points,prop.chisq=FALSE,prop.r=FALSE,prop.t=FALSE)

Highlight this text in the R-editor and press Ctrl+R. The result shown in the ‘R’ console will look like the below Screen Shot 23.

Screen Shot 23

This is the same result as the table built in Excel by-hand with COUNTIF formulae. The advantage is that rather than needing to copy and paste tables and alter several formulae to make additional cross-tabulations, all subsequent cross-tabulations can be generated by only copying over the previous analysis variable with the next variable of interest and running the line of code.

If the variable of interest is not already grouped but a continuous numeric value is (such as a financial statement variable that can range from zero to millions), one extra line of code can be added to call the ‘cut2’ function and create a number of proportionate groups selected by the user. For example, for the variable ‘Years.in.Operation’, which is the number of years rather than the expert points for those years, the following code groups the data into 3 proportional groups. The number of groups can be increased or decreased by changing the number before the closed parentheses (for example to ‘g = 2’, ‘g = 5’ or ‘g = 7’).

flat$Years.in.Operation_bin <- cut2(flat$Years.in.Operation,g=3)

This code creates a new column in the R data table named ‘flat’, and this new column is named ‘Years.in.Operation_bin’, but it could be named anything chosen by the analyst. The word ‘bin’ is chosen to indicate this is a grouping variable.

To see a cross tabulation of the grouped variable, copy and paste the CrossTable code and paste in the name ‘Years.in.Operation_bin’ as the new analysis variable, as follows:

CrossTable(flat$Loan.Status,flat$Years.in.Operation_bin,prop.chisq=FALSE,prop.r=FALSE,prop.t=FALSE)

Now execute the two lines of code by highlighting them and pressing Ctrl+R, and the result shown in the R Console is shown in Screen Shot 24.
The ranges of each group are indicated by the numbers in brackets in the top row: in this case, the first group is 3 to 6 years, the second is 7 to 16, the last is 17 to 53 (it is not clear from the displayed output that the upper bound of each range is not included in that group, but can be checked by looking in the data frame). The groups are made so that the 3 groups are roughly equal, as the row-wise Column Total indicates. The cut2 function is useful for looking for naturally occurring risk patterns in the data with continuous variables.

Information on how to do anything in R can be found on the internet using Google. This technical note Guide has limited R coding examples to the creation of cross tabulations, but as noted at the beginning of this section, using R facilitates writing an entire analysis programme that can be easily adjusted and repeated (or refreshed) given new data or in case errors in the data are discovered at any point during the analysis.

---

**About the authors:** Dean Caire, CFA, has developed over 100 scorecards in over 50 countries, with more than half of those projects involving very limited historic data. The simpler methods advocated here have been developed over time to transfer the necessary scorecard development and management skills to a wide range of bankers with different skill sets. Any questions can be directed to: dean_caire@hotmail.com.

The Kenya Financial Sector Deepening (FSD) programme was established in early 2005 to support the development of financial markets in Kenya as a means to stimulate wealth creation and reduce poverty. Working in partnership with the financial services industry, the programme’s goal is to expand access to financial services among lower income households and smaller enterprises. It operates as an independent trust under the supervision of professional trustees, KPMG Kenya, with policy guidance from a Programme Investment Committee (PIC). In addition to the Government of Kenya, funders include the UK’s Department for International Development (DFID), the World Bank, the Swedish International Development Agency (SIDA), Agence Française de Développement (AFD) and the Bill and Melinda Gates Foundation.