

CHAPTER 1

Introduction to Financial Modeling

What is a financial model? What is the difference between a financial model and the spreadsheet solutions you create or VBA programs you write all the time to answer financial questions or solve financial problems?

A simple, practical answer is that a financial model is designed to represent in mathematical terms the relationships among the variables of a financial problem so that it can be used to answer “what if” questions or make projections. Some of the spreadsheet solutions that people create capture some of these relationships as well and, therefore, can answer “what if” questions to some extent. But because they are not primarily designed with these objectives in mind, they do not try to capture as many of these interdependencies as possible, and their structures often make it cumbersome to answer “what if” questions or make projections with them.

This may sound a little abstract. So let us look at a simple, concrete example. Suppose you are using a spreadsheet to calculate, based on your taxable income, what your after-tax income was last year. Income tax rates vary in steps (brackets) for different income levels. So you cannot simply calculate your taxes by multiplying your taxable income by one tax rate (30%, for example) and subtracting it from your taxable income to get the after-tax income.

Consider two approaches to setting up a spreadsheet to calculate the after-tax income. In the first approach, you can enter your taxable income in a cell, calculate the tax on the income (using a hand calculator and the tax rates for the different tax brackets), and enter it in the cell below. Then you can write an equation in another cell to calculate your after-tax income by subtracting the tax in the second cell from the taxable income in the first cell.

This spreadsheet solution will give you the answer to your immediate question, but it is not a useful financial model. Why? Because it does not capture the key mathematical relationship between taxable income and taxes. The result is that if you now try to answer the “what if” question, *What would my after-tax income have been if my taxable income were \$10,000 higher?*, you will have to go back to doing the calculations by hand.

However, you can set up your spreadsheet to calculate the taxes on any taxable income (using the different tax brackets and tax rates) and use the computed tax number to calculate your after-tax income. You will then have a financial model, because it will capture the relationship between taxable income and taxes. You also will be able to use this model to answer the “what if” question I posed before.

In creating financial models, you always have to keep in mind that you want to capture as many of the interdependencies among the variables of the model as possible. In addition, you want to structure your models in such a way that it is easy to ask “what if” questions, that is, change the values of the independent variables and observe how they affect the values of the key dependent variables. You also should recognize that some of the relationships, as in the case of taxes, are easy to establish and exact; but many others will be approximate or even unknown. You will have to come up with them based on financial theory, analysis of data, and so on, and coming up with these relationships is one of the major challenges of financial modeling. Generally, the more of these relationships you can come up with and incorporate into your model, the more useful your model will be.

My Assumptions about You and the Users of Your Models

In this book I assume that you know the basics of finance and can solve by hand most of the problems for which you will be creating models. I also assume that you are familiar with the basics of Excel and have experience creating spreadsheet solutions to at least simple problems. You do not need to have knowledge of Excel’s advanced features or of modeling; I will cover both in detail. You also do not need to have any knowledge of VBA. A key objective of the book is to teach you VBA and modeling using VBA from scratch by way of an easy and effective method.

Another important assumption I am making is that you will be developing the models primarily for your own use or for use by people who have some experience with Excel, but not necessarily with VBA. When you create models for use by people who have little or no familiarity with Excel, it requires adding special user interfaces to make the models easy to use. One must build into them special features to make them “bulletproof”—that is, to make sure that the models will not crash or produce wrong results if someone enters inappropriate inputs. I will discuss some design methods and Excel features that make models easier to use and more “bullet-resistant.” Most everyday modelers do not need to go beyond this.

Excel and VBA as Modeling Tools

Even in the mid- to late 1990s, Excel was not considered a powerful enough tool for serious financial modeling, in part because the PCs available at the time had

speed and memory limitations. With advances in PCs and improvements in Excel itself, the table has now turned completely: Excel has become the preferred tool for creating all but the largest and most computationally intensive financial models. The advantages of Excel for financial modeling are so obvious that it is not necessary to go into them. However, for those who have not worked with other programs or programming languages for modeling, it is worthwhile to point out that one of the important advantages of Excel is that with Excel you can create excellent output with very little work. You should learn to take full advantage of Excel's power in this respect.

If Excel is so good, then, why bother with VBA? VBA is a programming language, and if you do not know anything about programming languages, it will be difficult for you to appreciate the advantages of VBA at this point. Let me touch on only a few key reasons here, and I will answer the question in greater detail when we discuss modeling with VBA.

Despite its power, Excel has many limitations, and there are many financial models—some even relatively simple ones—that either cannot be created in Excel or will be overly complex or cumbersome to create in Excel. What's more, when you create a highly complex model in Excel, it can be difficult to understand, debug, and maintain. VBA generally offers a significant edge in all these respects.

The problem that most people have with VBA is that it is one more thing to learn, and they are somehow afraid of trying to learn a programming language. The reality is that if you follow the right method, learning a programming language is not particularly difficult—especially if you selectively learn what you will really use (as we will do in this book) and not let yourself get lost in all the other things you can do with VBA but probably never will. The truth is that you do not need to learn all that much to be able to create very useful and powerful financial models with VBA. What you will need is a lot of practice, which you will get as you go through this book. VBA offers you the best of both worlds: you can take advantage of all the powers of Excel including its ability to easily create excellent outputs, and supplement them with VBA's additional tools and flexibility.

Independent and Dependent Variables

We can say that the purpose of a model is to calculate the values of certain dependent variables for the values provided for its independent variables. It is therefore important to understand the difference between independent and dependent variables.

Independent variables are also called the input or external variables. The model's user or creator inputs the values of these variables—they are not calculated by the model. These are the variables you change to ask “what if” questions. For example, in our simple model the taxable income is an independent variable.

A model may also include a special type of input variable called a parameter. Parameters are independent variables in that their values are also provided by the

creator or user of the model. The difference is that their values are expected to remain constant or change infrequently within the context of the model. For example, the tax rates and the tax brackets in our simple model can be considered parameters of the model because their values have to be provided for the model to work, but these values are not expected to change frequently. As you create a model, it is useful to keep the parameters together but separate from the other independent variables. They should still be easy to see and change, however.

The variables whose values are calculated by the model are called the calculated or dependent variables. Some of them may be intermediate variables, calculated for use in other calculations. Others are of primary interest to the user and are the output variables of the models. Models are almost always created to observe how the values of the output variables will change with changes in the values of one or more independent variables. Dependent variables are the ones whose values we want to project or determine when we ask “what if” questions.

It is possible to distinguish between intermediate dependent variables and output dependent variables; intermediate dependent variables are used in further calculations, whereas output dependent variables are not. This is generally not a useful distinction, however. It is better to look at the dependent variables of primary interest as the output variables of the model irrespective of whether they are used in further calculations. One must also recognize that, from time to time, some dependent variables that were previously not considered output variables of a model can become so and vice versa.

STEPS IN CREATING A MODEL

Whether you are creating a financial model using Excel or VBA, you must take a systematic approach. A systematic approach always involves planning ahead and this takes some time. Most people do not like to plan and think they can save time by starting to build a model right away without spending time on planning. However, for all but the simplest models, not taking the time upfront to do some planning and not taking a systematic approach ends up being both frustrating and a waste of time.

Here are the key steps you should follow in creating both Excel and VBA models. The details vary somewhat depending on whether you are working with Excel or VBA, and I will discuss them in later chapters. You should keep two other things in mind. First, in practice, you do not have to follow the steps strictly in this order, nor do you have to finish one completely before going onto the next one. Most of the time you will have to go back and forth to some extent. It will depend on the circumstances. Second, over time, you should try to create your own variation on this basic approach and learn to adapt it to different situations.

Excel and VBA are flexible tools and you can usually make changes almost at any stage without a great deal of difficulty. But this still will take more time

than if you do it right the first time, and making changes later increases the chances of missing some of the other changes that have to go with them.

Step 1: Define and Structure the Problem

In real life, problems rarely come neatly defined and structured. Unless you take the time upfront to define and structure the problem and agree on them with the user (your boss, for example), you may end up having to extensively change the model you first create. When your boss asks you a question whose answer requires developing a model, she often has only a vague idea of what she is really seeking. As a finance person and a modeler, you are responsible for putting it all in more concrete terms before proceeding.

Start by discussing and defining why the model is needed and what decisions, if any, will be made based on its output—that is, what questions the model is supposed to answer. Then establish how accurate or realistic the outputs need to be. As we discussed, all models have to capture the relationships among their variables, and discovering and quantifying these can take a lot of time. How much effort you put into doing this should depend on how important the project is and how accurate or realistic the outputs need to be.

Step 2: Define the Input and Output Variables of the Model

Make a list of all the inputs the model will need and decide who will provide them or where they will come from. This is crucial. For example, if you are creating a model to do the business plan for your company, the inputs must come from the business managers. You cannot just guess what sales growth rates they will be able to achieve, how much they will have to spend on plants and equipment to support those sales growths, and so forth. You may not need the actual numbers upfront, but the list of inputs should be established based on your discussions with the business managers so that you can make them independent variables in your model. Otherwise you may have to go back later on and change a lot of things in the model.

Make a list of the tabular, graphical, and other outputs the model needs to create. To some extent, these should be driven by the decisions that will be made based on them. One advantage of Excel is that a lot of the output can be just printouts of your spreadsheets, provided the spreadsheets have been laid out properly. If you plan ahead and lay out your spreadsheets with the outputs in mind, you will save yourself a lot of time later on.

Step 3: Decide Who Will Use the Model and How Often

Who will use the model and how often it will be used make a lot of difference. In this book, I am assuming that you are developing the models either for your

own use or for use by others who are familiar with Excel and understand the model, at least to some extent. When you create models for others' use, it involves much more work. You have to make sure that these people cannot enter data that do not make sense, they cannot accidentally damage parts of the model, and they can get the necessary outputs automatically and so forth. These are collectively called the user interface, and the more elegant, more easy to use, and more robust you want to make a model, the more work it is. You also have to plan for many of these features ahead of time.

How frequently a model will be used is another important issue. If a model is going to be used only once in a while, then it does not matter if it takes a long time to run or if it takes some extra work every time to create the outputs. A model that will be used frequently, however, should be designed differently.

Step 4: Understand the Financial and Mathematical Aspects of the Model

It is important to remember that the computer cannot do any thinking; you have to tell it exactly how all the calculations in the model will have to be done. In most situations, if you do not know how you would do the calculations by hand, you are not going to be able to write the necessary formulas or instructions for the computer to do it. It does not pay to start building the model until you are sure you could solve the problem by hand.

It usually takes beginners a lot of time to create a model and they often think that it is their Excel or VBA skills that are slowing things down. This may be partly true, but at least as often the problem is in their understanding of the finance and mathematics of the model they are trying to create. You will save a lot of time if you do not even sit down in front of the computer to create a model until you are sure that you know how to solve the problem.

Step 5: Design the Model

There are two aspects to designing a model. One is to sketch the steps that Excel or VBA will have to follow to solve the problem. For simple models, you may want to write down only the broad steps or perhaps even do it in your head. For more complex problems, however, you should work on paper and use a degree of detail that suits your level of experience and the complexity of the problem. The less experience you have, the more detailed the sketch should be. Once again, remember that this may seem like a waste of time, but ultimately it will save you time compared to plunging into your spreadsheet or VBA program without such a sketch of the model.

The other aspect of design is planning how the model will be laid out in Excel or VBA. Are you going to do the entire model in one spreadsheet (or VBA module) or split it into several spreadsheets (or VBA modules or procedures)? Editing an Excel or VBA model is easy. So you do not have to decide every detail ahead

of time, but you need to have an overall design in mind or on paper depending on the complexity of the problem and your level of experience.

As I discussed before, you also need to think about the kind of user interface you want to create and the reports you want the model to produce.

Step 6: Create the Spreadsheets or Write the VBA Codes

For most models, this is the big step. Most of this book covers the details of this step, so there is no need to get into them here.

Step 7: Test the Model

Almost no model works correctly the first time it is used; you have to find the problems (bugs) and fix them. The bugs that prevent the model from working at all or produce obviously wrong answers are generally easier to find and fix. However, models often include hidden bugs that create problems only for certain values or certain combinations of values for the input variables. To find them, you have to test a model extensively with a wide range of input variables.

You have to take somewhat different approaches to testing and debugging a model depending on whether you are working with Excel or VBA. Both Excel and VBA provide some special tools for this purpose; I will discuss these tools and provide suggestions on how to debug models in Excel and VBA in later chapters. Here are a few helpful hints that apply to both:

- There is no standard approach to testing and debugging a model. You almost always have to use your ingenuity to figure out what will be the best way to test and debug a particular model. Your ability to do so will improve with experience.
- The better you understand a problem and a model, the easier it will be to debug it. If you understand how changes in certain independent variables affect the values of certain dependent variables, then you can change the values of the independent variables to see if the dependent variables are changing in the right direction and by the right orders of magnitude. This is one of the best tools, especially for debugging large models, and you should do a lot of testing using this approach. You can also use this approach to hunt down the sources of the problems: Starting from a value that looks wrong, backtrack through the values of the intermediate dependent variables to see where the problem may be originating. This approach may sound somewhat vague and abstract, but with experience you will find that you can locate and fix most bugs rapidly using this approach.
- Checking a model's output against hand-calculated answers is a common and effective approach to debugging. In some situations, doing hand calculations may not be practical, but you may be able to use Excel itself to do some side calculations to test individual parts of the model.

Step 8: Protect the Model

Once you have completed a model, and especially if you are going to give it to others to use, you should consider protecting it against accidental or unauthorized changes. In addition, you may also want to hide parts of the model so that others cannot see certain formulas, data, and so on. Excel provides several flexible tools that you can use to hide and protect parts or all of your model. A good strategy is to cluster and color code all the input cells of a model and protect and hide everything else in the workbook.

There is less need to protect VBA modules because most users do not even know how to open them. Nonetheless, if you think it is necessary, you can protect parts of your VBA models as well.

Step 9: Document the Model

Documenting a model means putting in writing, diagrams, flowcharts, and so on, the information that someone else (or you yourself in the future) will need to figure out what it does, how it is structured, and what assumptions are built into it. One can then efficiently and effectively make changes to (update) the model if necessary.

For large systems (for example, the reservation systems for airlines), the amount of necessary documentation can be enormous; it is often put on CDs for easy access and use. Professional system development organizations have elaborate standards for documentation, because different pieces of large systems are developed by different people—many of whom may not be around for very long. Also, it is almost certain that the systems will have to be constantly updated.

Over time, anyone who creates models develops his own system of documentation. As long as you keep in mind the objectives I mentioned before, you have a lot of leeway to come up with your own system as well. Both Excel and VBA offer a number of features that let you easily do a lot of the documentation as you work on your model. You should take full advantage of them and do as much of your documentation as possible while creating the model.

This is important for two reasons. First, if you write your documentation when things are fresh in your mind, it will save you time later and you will be less likely to forget to document important things. Second, everyone hates (or learns to hate) documentation. It is no fun at all, especially if you try to do it all at once at the end of the project. If you do not work on the documentation until the end, chances are you will never do it. Then, if you have to use the model again a few months later or have to update it, you will end up spending hours or even days trying to figure out what you did. Do your documentation as you go along and finish it immediately after your model is done.

You have to take somewhat different approaches to when you document Excel and VBA models. I will discuss how in the appropriate later chapters.

Step 10: Update the Model as Necessary

This is not a part of the initial model development, but almost all models require updating at some point, either because some things have changed or because you want to adapt it to do something else. This is where the documentation becomes useful. Depending on how much updating is involved, you may want to go through all of the above steps again. You should also thoroughly update the documentation and include in it the information on who updated it, when and why, and what changes were made.

HOW THIS BOOK IS ORGANIZED

To use this book effectively, it will be helpful to understand some of my thinking behind its organization. Modeling is a skill that you can develop only by creating a variety of models. Once you have developed the basic skill, you should be able to create increasingly complex models—as long as you understand the finance and mathematics of the underlying problems. In some ways, it is like learning a language, especially when you are working with VBA. Once you learn the language, you can say new things in that language—you can make up sentences that you have never heard before. I therefore emphasize learning the language by exposing you to a variety of models in different areas of finance instead of concentrating on one particular area like derivatives or trying to anticipate and include every model you may need to develop.

For both Excel and VBA, I have provided one part that covers the “grammar” of the language (Parts One and Three) and then a second part that provides the examples of its use (Parts Two and Four) to create models. Just as you do not need to know everything about the grammar of a language before you can start using it, you do not need to know everything about Excel or VBA before you can start creating models using them. I suggest that you start with Part One to become familiar with Excel and quickly move on to working on the models in Part Two. Do not spend a lot of time trying to master the grammar. Come back and learn it as you need it.

In each chapter in Part Two, I have included a section called “Review of Theory and Concepts.” These are the theories and concepts of finance that you will need to work on the models in that chapter. If you are familiar with them, you can skip them and go on to the models. If you want to refresh your memory and understanding, the material is there.

You will notice that many of the modeling chapters in Parts Two and Four have the same titles. There is a reason for including such parallel chapters. It has been my experience that the easiest way to learn modeling with VBA is to start with problems that you can already model in Excel, because then you already know a lot about the problem and you can focus on the VBA aspects of it. In

many cases, the parallel VBA models also demonstrate that even if you can model a problem using Excel, using VBA can provide additional flexibility, power, and so on. Most of the VBA chapters include additional models that are cumbersome or impossible to create using Excel.

What this means is that you may be better off covering Parts One and Two before you go on to the VBA parts. However, if you are already good at modeling with Excel and are primarily interested in learning VBA and modeling using VBA, you can start with Part Three. Then, as you cover the chapters of Part Four, you can review the corresponding chapters in Part Two as necessary.

Because there are certain differences between Excel-based models and VBA-based models, I have provided in the first chapters of both Parts Two and Four additional information on how to develop good models using them. I have also included suggestions on how best to use the material in both parts to improve your modeling skills rapidly. You will progress faster if you read and follow these suggestions instead of taking a haphazard approach.