Excel

Content

- Software overview
- General pros & cons
- Model complexity
- Plotting
- Data
- Transparency
- Learning curve
- Conclusions

Software overview

- Excel is a spreadsheet developed by Microsoft for Windows, macOS, Android and iOS
- Features calculation, graphing tools, pivot tables
- Macro programming language called Visual Basic for Applications (VBA)
  - Visual Basic Editor (VBE)
- Industry standard for spreadsheets
- Excel forms part of Microsoft Office

General Pros

- Learning curve
  - Easy and efficient for constructing simple models
  - Point-and-click GUI
  - Often used in introductory courses and text books
- Excel models are implicitly transparent
- Extensible
  - Through the use of macros
  - Numerous third-party packages to enhance functionality
- Available cross-platform
  - E.g. MS Windows and Mac OS X operating system
- Links between Excel and other components of the MS Office Suite e.g. Word and Powerpoint, support efficient production of reports and presentations for dissemination

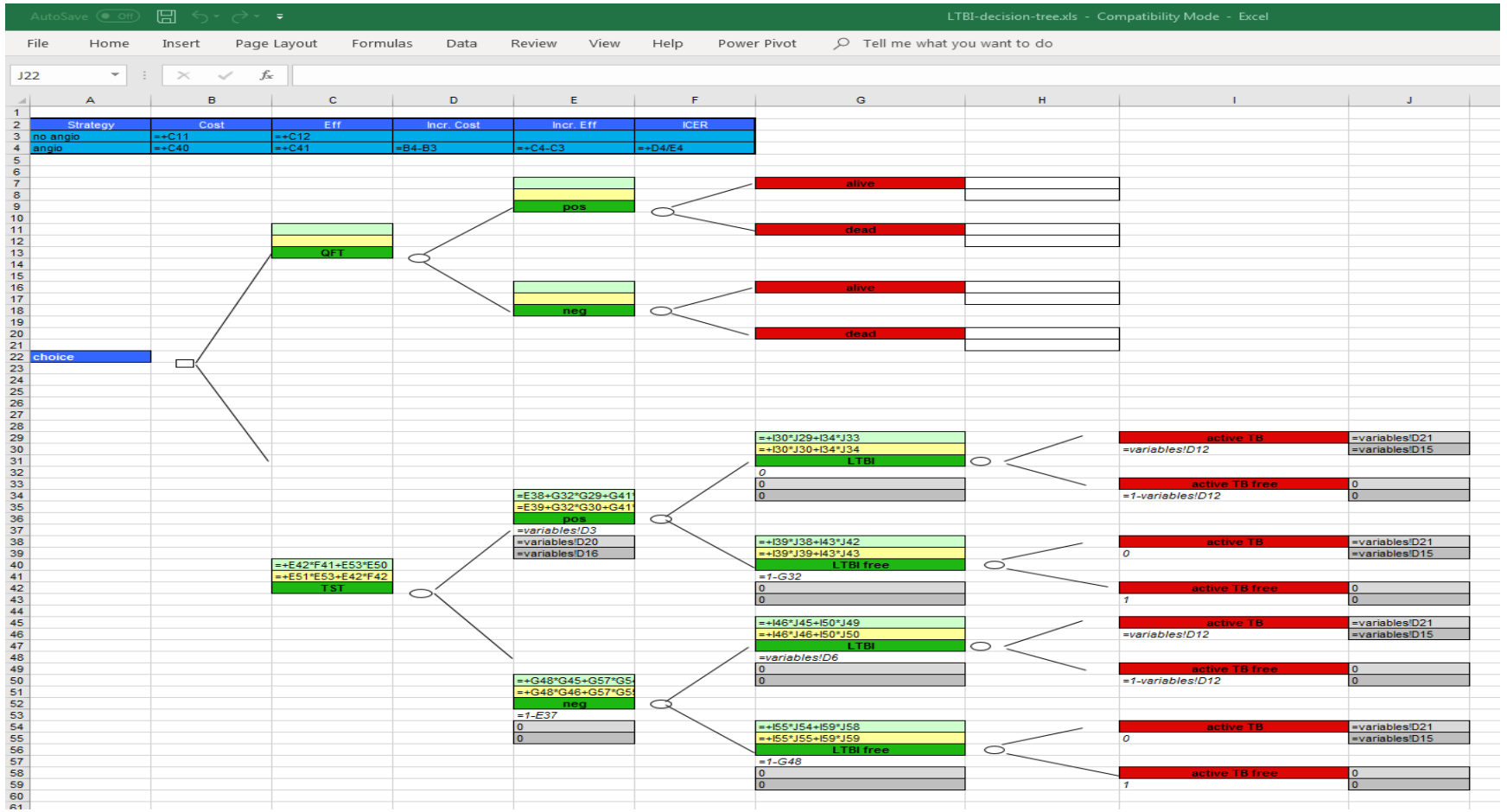
General Cons

- Constructing more complex models were better suited to other packages
- Repeated (probabilistic) simulation and reproducibility is not straightforward
- Steps of an analysis are not explicitly recorded
- Plotting limited
- Updated infrequently
- (Large) Data set reading, handling and saving in different formats can be intractable

Model complexity

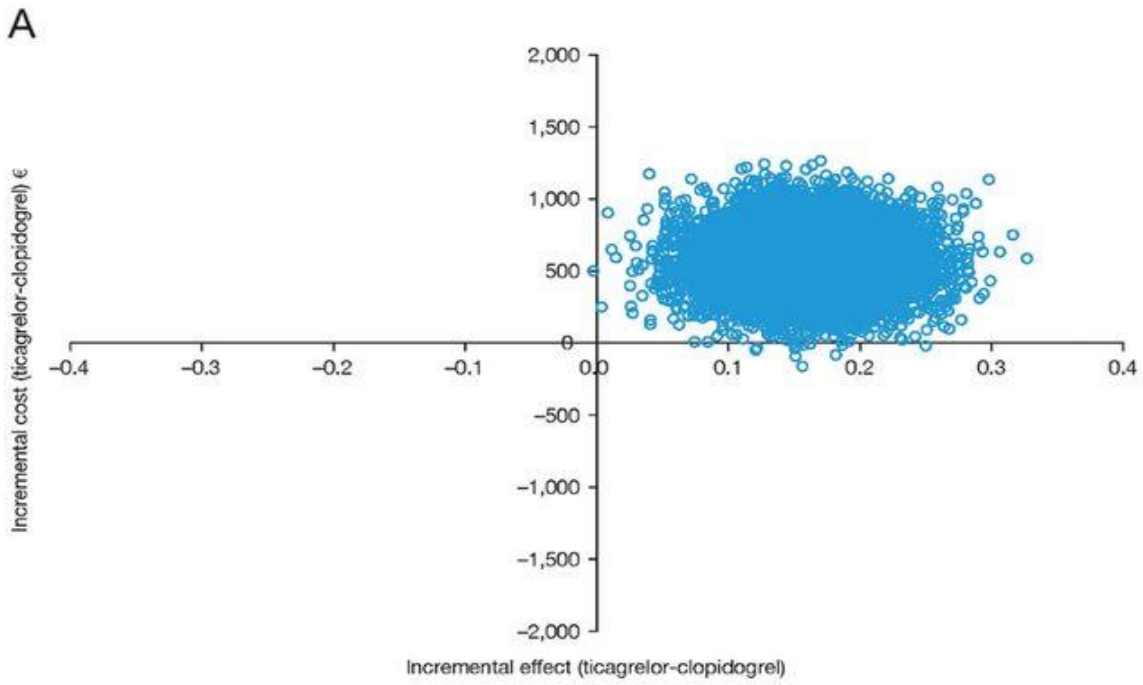
- Point-and-click GUIs are slow to update and cumbersome to use for larger models.
- Users may have to repeat similar steps many times
  - e.g. specifying a decision tree with hundreds of nodes may require multiple cutting and pasting, which is time-consuming and error-prone.

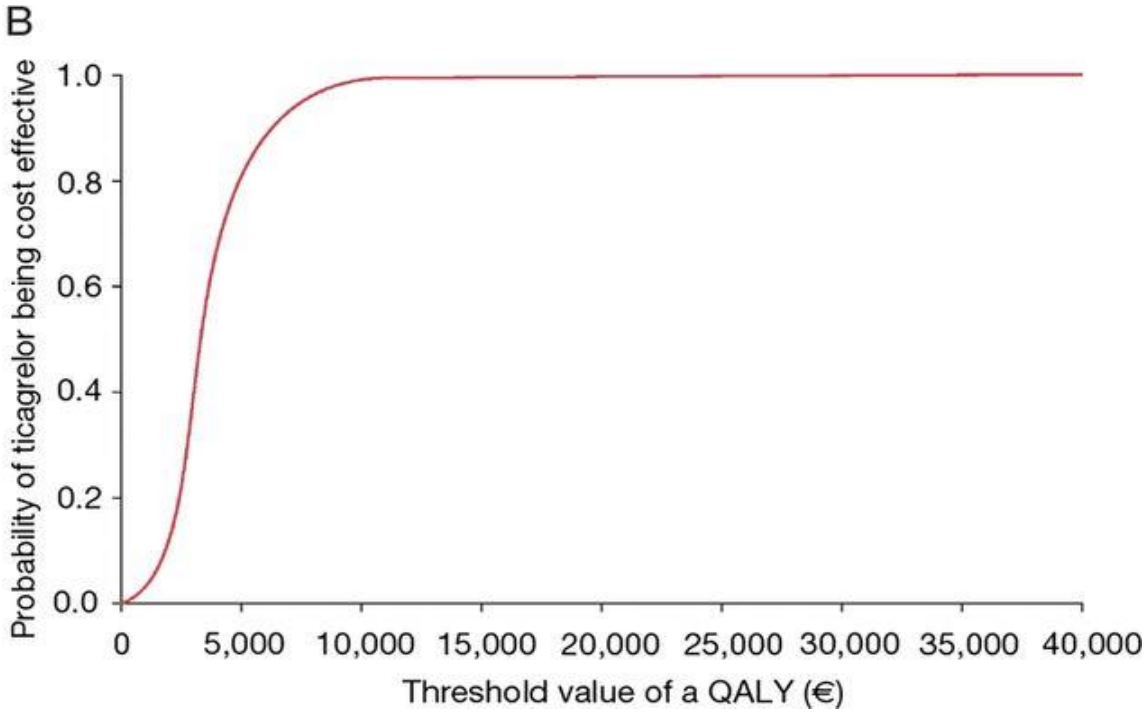
Example decision tree

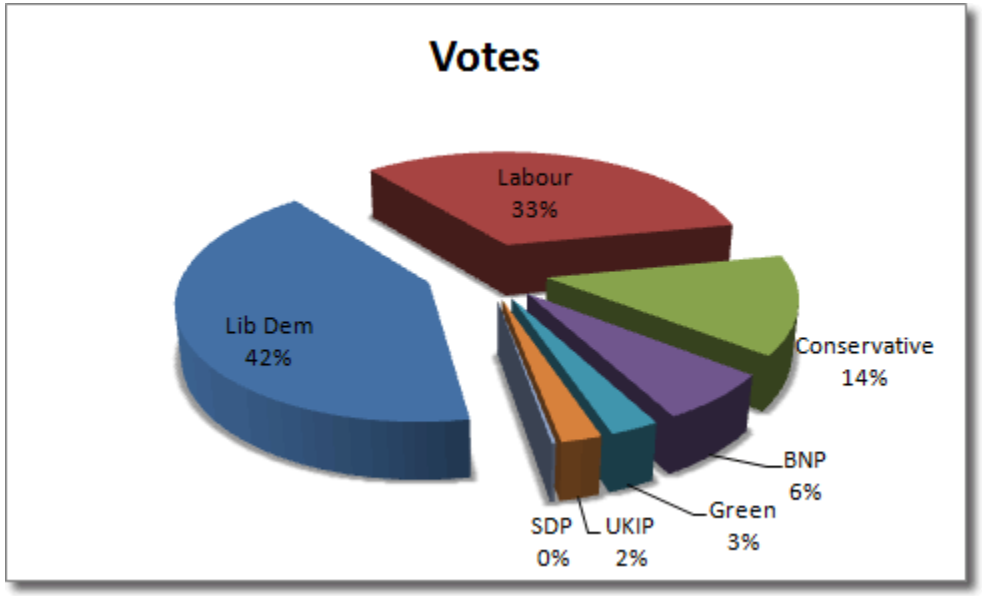


Plotting (Charts)

- Excel supports basic charts, graphs, or histograms
- The generated graphic component can either be embedded within the current sheet, or added as a separate object.
- These displays are dynamically updated if the content of cells change.


A
2,000
1,500
1,000
500
0
−500
−1,000
−1,500
−2,000
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
Incremental cost (ticagrelor-clopidogrel) €
Incremental effect (ticagrelor-clopidogrel)


B
1.0
0.8
0.6
0.4
0.2
0.0
Probability of ticagrelor being cost effective
0
5,000
10,000
15,000
20,000
25,000
30,000
35,000
40,000
Threshold value of a QALY (€)

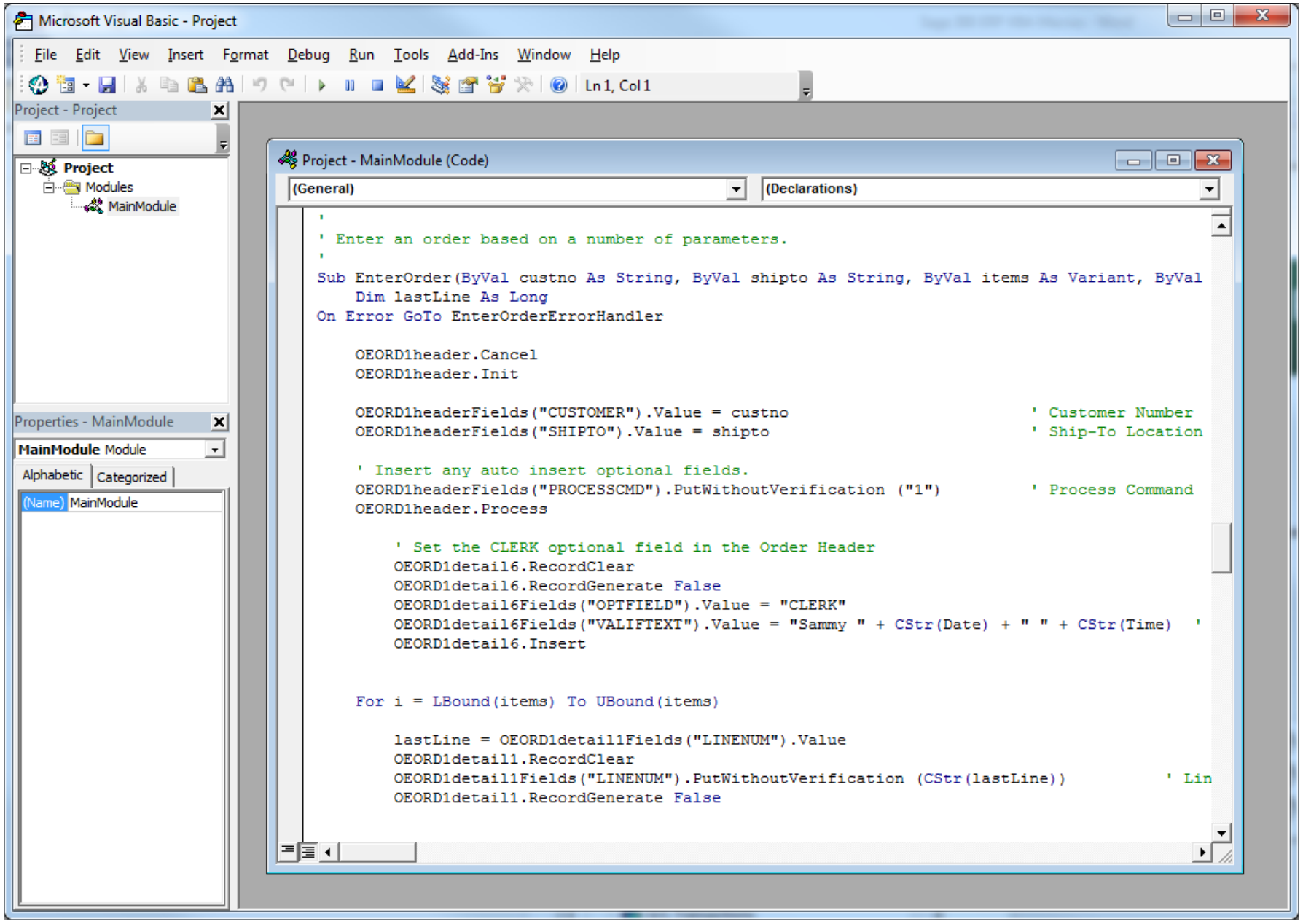

Votes
Labour
33%
Lib Dem
42%
Conservative
14%
BNP
6%
Green
3%
UKIP
2%
SDP
0%

Data

- Version 12.0 can handle 1M rows and 16384 columns
- File formats
  - Proprietary binary file format called Excel Binary File Format (.XLS)
- Data cleaning/munging/manipulation can be cumbersome
- Most versions of Microsoft Excel can read CSV, DBF, SYLK, DIF, and other legacy formats
- Support for some older file formats was removed in Excel 2007
- TreeAge offers an Excel module
  - Enables exporting variables and tables from TreeAge to Excel and vice versa

Exstensible via VBA

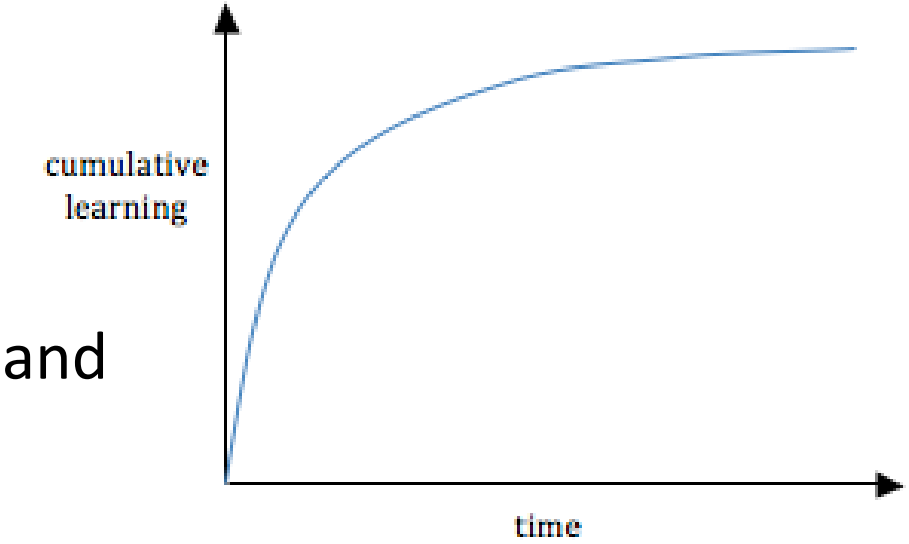
- Allows spreadsheet manipulation that is awkward or impossible with standard spreadsheet techniques
- Programmers may write code directly using the Visual Basic Editor (VBE)
  - Includes a window for writing code
  - Debugging code
  - Code module organization environment
- User can implement numerical methods as well as automating tasks such as formatting or data organization
- Any desired intermediate results can be reported back to the spreadsheet
- A common and easy way to generate VBA code is by using the Macro Recorder


Microsoft Visual Basic - Project
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 1, Col 1
Project - Project
Project
Modules
MainModule
Properties - MainModule
MainModule Module
Alphabetic Categorized
(Name) MainModule
Project - MainModule (Code)
(General)
(Declarations)
'
' Enter an order based on a number of parameters.
'
Sub EnterOrder(ByVal custno As String, ByVal shipto As String, ByVal items As Variant, ByVal
Dim lastLine As Long
On Error GoTo EnterOrderErrorHandler
OEORD1header.Cancel
OEORD1header.Init
OEORD1headerFields("CUSTOMER").Value = custno ' Customer Number
OEORD1headerFields("SHIPTO").Value = shipto ' Ship-To Location
' Insert any auto insert optional fields.
OEORD1headerFields("PROCESSCMD").PutWithoutVerification ("1") ' Process Command
OEORD1header.Process
' Set the CLERK optional field in the Order Header
OEORD1detail6.RecordClear
OEORD1detail6.RecordGenerate False
OEORD1detail6Fields("OPTFIELD").Value = "CLERK"
OEORD1detail6Fields("VALIFTEXT").Value = "Sammy " + CStr(Date) + " " + CStr(Time) '
OEORD1detail6.Insert
For i = LBound(items) To UBound(items)
lastLine = OEORD1detail1Fields("LINENUM").Value
OEORD1detail1.RecordClear
OEORD1detail1Fields("LINENUM").PutWithoutVerification (CStr(lastLine)) ' Lin
OEORD1detail1.RecordGenerate False

Transparency

- Parameters should be obvious as should be model structure
- Use of cell names is required for efficient updating of parameter values in any but the most simple models.
- Models can become opaque through “cell chasing” where references and names lead to a tangled web of variables
- The ability to highlight these relationships, using the trace precedent and dependent facility, offers some reprieve
- Validation of macros requires additional technical skill.
- The seed cannot be set so any model should be reproducible upon demand
  - E.g. without storing the entire draw for the entire simulation
  - If one changes one parameter of one distribution, all the other draws should be identical for the sake of consistency.
- When non standard software is used could be a validation model
  - This would allow the fundamental structure and parameter values to be interrogated and base case and univariate sensitivity analyses to be replicated or conducted

Learning curve



- Many user have at least basic knowledge of Excel and spreadsheets
- With an understanding of spreadsheets and the linkages between cells, it is relatively straightforward to implement a CEM in Excel
- However, stochastic models require some command of VBA, and more complex simulations require an increasingly sophisticated command of VBA
- Provides a rapid development environment where students are forced to engage not only with the concepts but the maths as well
- Learning new software may be timely and costly

Conclusions

- Excel is great for what it is designed to do
- Its intuitive, simple to use and transparent for smaller models
- However, when models are more complex its strengths can become its weaknesses
  - E.g. navigating a model (cell chasing)
- Excel works well *complementary to* other tools
  - E.g. model validation, prototyping, etc