Introduction to Excel and Visual Basic for Excel

Gilbert Ritschard

Department of economics, University of Geneva
http://mephisto.unige.ch

Master in International Trading,
Commodity Finance and Shipping
Outline

1 Excel, what is it?
2 Excel: Basics
3 Macro and Visual Basic (VBA) programming
Outline

1. Excel, what is it?
2. Excel: Basics
3. Macro and Visual Basic (VBA) programming
Excel, what is it?

Excel is a **spreadsheet**

Interactive table (rows and columns) for managing and exploring data.

**Allows**

- **Computation** (arithmetic and other mathematical and statistical functions)
- **Simple data management**
  (sort and filter according to one or several keys alphabetic order, dates, values, ...)
- **Organizing and presenting tables** (frame, totals, rows, columns)
- **Graphical rendering of numerical data**
  (but most often requires data preprocessing)
Excel, what is it?

Excel is a **spreadsheet**

Interactive table (rows and columns) for managing and exploring data.

Allows

- **Computation** (arithmetic and other mathematical and statistical functions)
- **Simple data management**
  (sort and filter according to one or several keys alphabetic order, dates, values, ...)
- **Organizing and presenting tables** (frame, totals, rows, columns)
- **Graphical rendering** of numerical data
  (but most often requires data preprocessing)
Outline

1 Excel, what is it?

2 Excel: Basics

3 Macro and Visual Basic (VBA) programming
Rows, indexed by numbers (1,2,3,...)
- Columns, indexed by letters (A,B,C,...)
- Cell, intersection of a column (ex B) and a row (ex 3)
- Reference Address
  - of a cell: Column letter and row number (ex B3),
  - of a table (range): B2:D5
Rows, indexed by numbers (1,2,3,...)

Columns, indexed by letters (A,B,C,...)

Cell, intersection of a column (ex B) and a row (ex 3)

Reference Address

- of a cell: Column letter and row number (ex B3),
- of a table (range): B2:D5
Spreadsheet

- **Rows**, indexed by numbers (1, 2, 3, ...)
- **Columns**, indexed by letters (A, B, C, ...)
- **Cell**, intersection of a column (ex B) and a row (ex 3)
- **Reference Address**
  - of a cell: Column letter and row number (ex B3),
  - of a table (range): B2:D5
Activate a cell by clicking on it

All what you type in (text, number or formula) goes in the active cell.

- Validate an entry either with [Enter] or by pressing a displacement key:
  - ←, ↑, ↓, →
  - [Enter] (move to next row, same column)
  - [Tab] (move to right cell)
  - [PgUp], [PgDn]
Formula

First inserted symbol is = ⇔ Formula

- Cell displays the result of the formula. Example:
  - B1 contains 3
  - B2 contains 4
  - B3 contains =B1+B2
- La cellule B3 affiche 7, i.e., the sum of the content of celles B1 et B2.
- We can use:
  - Arithmetic operators: +,−,*,/,^,
  - Functions (menu tab: Formulas): sum, average, count, max, min, ...
First inserted symbol is $= \iff$ Formula

- Cell displays the result of the formula. Example:
  - B1 contains 3
  - B2 contains 4
  - B3 contains $=B1+B2$

- La cellule B3 affiche 7, i.e., the sum of the content of celles B1 et B2.

- We can use:
  - Arithmetic operators: $\ +, -, *, /, ^$
  - Functions (menu tab: Formulas): sum, average, count, max, min, ...
Excel and Visual Basic for Excel
Excel: Basics

Formula

First inserted symbol is = ⇔ Formula

Cell displays the result of the formula. Example:

- B1 contains 3
- B2 contains 4
- B3 contains =B1+B2

La cellule B3 affiche 7, i.e., the sum of the content of celles B1 et B2.

We can use:

- Arithmetic operators: +, −, *, /, ^
- Functions (menu tab: Formulas): sum, average, count, max, min, ...
Moving and copying cells

- **Selecting cells**
  - **menu**: Cut or Copy, on Home tab
  - **keyboard**: Ctrl-X or Ctrl-C
  - Select a destination area either of same size or top left cell of the area.

- **Paste**
  - **menu**: Paste on left of Home tab
  - **keyboard**: Ctrl-V

- **Mouse**: To move: place mouse cursor on the border of the selected area (cursor cursor becomes an arrow), click, drag and drop by releasing the mouse button.
  
  To paste: press [Ctrl] while dragging.
Moving and copying cells

- **Selecting cells**
  - **menu**: Cut or Copy, on Home tab
  - **keyboard**: Ctrl-X or Ctrl-C
  - Select a destination area either of same size or top left cell of the area.

- **Paste**
  - **menu**: Paste on left of Home tab
  - **keyboard**: Ctrl-V
  - **Mouse**: To move: place mouse cursor on the border of the selected area (cursor cursor becomes an arrow), click, drag and drop by releasing the mouse button.
  - To paste: press [Ctrl] while dragging.
Moving and copying cells

- **Selecting cells**
  - **menu**: Cut or Copy, on Home tab
  - **keyboard**: Ctrl-X or Ctrl-C
  - Select a destination area either of same size or top left cell of the area.

- **Paste**
  - **menu**: Paste on left of Home tab
  - **keyboard**: Ctrl-V

- **Mouse**: To move: place mouse cursor on the border of the selected area (cursor cursor becomes an arrow), click, drag and drop by releasing the mouse button. To paste: press [Ctrl] while dragging.
Example: Conversion table between Celsius and Fahrenheit.
Enter:
- B1 = C, C1 = F (titres des colonnes)
- B2 = 0, C2 = ‘=32+(9/5)*B2’
- B3 = 5

Select C2, put cursor on small square bottom-right of selected area (cursor changes to +)
drag one case below.
Check that C3 contains =32+(9/5)*B3

Remark: formula was copied with relative reference.
Fill an area

- Example: Conversion table between Celsius and Fahrenheit.
- Enter:
  - $B1 = \text{C}, C1 = \text{F}$ (titres des colonnes)
  - $B2 = 0, C2 = '=32+(9/5)*B2$
  - $B3 = 5$
- Select C2, put cursor on small square bottom-right of selected area (cursor changes to +)
- drag one case below.
- Check that C3 contains $=32+(9/5)*B3$
- Remark: formula was copied with relative reference.
Fill an area (2)

- Select now area B2:C3,
- put cursor on the small square at bottom right of selection and drag until row 12.

- 1st column: sequence of numbers with increment of 5
- 2nd column: copy of formula with relative reference.
Relative versus absolute references (1)

Relative reference

- Formula in C2 \((=32+(9/5) \times B2)\) contains a relative reference to B2 (cells on left of C2).
- In any copy of C2, B2 will be replaced by the address of the cell on its left.

Absolute reference

- Reference to a fixed column:
  Specified with a $ in front of the letter ($B2)
- Reference to a fixed row:
  Specified with a $ in front of the number (B$2)
- Reference to a :
  Specified with a $ in front of each the letter and the number ($B$2)
Relative versus absolute references (1)

**Relative reference**
- Formula in C2 ($=32+(9/5)\times B2$) contains a relative reference to B2 (cells on left of C2).
- In any copy of C2, B2 will be replaced by the address of the cell on its left.

**Absolute reference**
- Reference to a fixed column: Specified with a $ in front of the letter ($B2$)
- Reference to a fixed row: Specified with a $ in front of the number (B$2$)
- Reference to a : Specified with a $ in front of each the letter and the number ($B$2$)
Relative versus absolute references (2)

Absolute references do not change when moved or copied

Example:

- A1 = quantity, B1 = price
- A2 = 1, B2 = 20, C2 = (unit price)
- A3 = 2, B3 = =A3*$B$2
- A4 = 3
- A5 = 5
- A6 = 10

Fill the second column with content of cell B3.

Check that B6, for example, contains =A6*$B$2
Some useful functions

SUM(x) sum
AVERAGE(x) mean value
VAR.P(x) variance
VAR.S(x) estimated variance
STDEV.P(x) standard deviation
STDEV.S(x) estimated standard deviation
MEDIAN(x) median
QUARTILE(x, k) kth quartile
COVARIANCE.P(x, y) covariance
CORREL(x, y) Pearson linear correlation
TRANSPOSE(A) Transpose of matrix A
MMULT(A, B) Product of matrices AB
MDETERM(A) Determinant of A
MINVERSE(A) Inverse of matrix A
LN(x) natural logarithm
LOG(x, b) logarithm to base b
EXP(x) $e$ raised to the power $x$: $\exp(x) = e^x$
Probability distributions

(cumulated when $c = \text{TRUE}$)

- NORM.DIST($x, \mu, \sigma, c$): normal distribution $N(\mu, \sigma^2)$
- NORM.S.DIST($x, c$): standardized normal distribution $N(0, 1)$
- CHISQ.DIST($x, d$): Chi-square distribution for $d$ degrees of freedom
- NORM.INV($p, \mu, \sigma$): inverse of normal distribution
- NORM.S.INV($p$): inverse of $N(0, 1)$ distribution
- CHISQ.INV($p, d$): inverse of Chi-square distribution
Outline

1. Excel, what is it?
2. Excel: Basics
3. Macro and Visual Basic (VBA) programming
Recording a macro

Best way to start with VBA programming, is

1. record a macro
2. look at the macro
3. edit the macro
Example

- Fill a range with values from 1 to 20 and put a heading ‘x’.
- Next to column x, create a column y with 20 random numbers (by inserting =RAND() in the cells)
- Rename the Sheet as ‘My Data’.
- Now we want to create a macro that
  - makes a copy of the values (not the formula) in the range of the x and y variables in a new sheet
  - renames the new sheet as ‘Outcome’
- Record a first macro
  - Go to the ‘Developer’ menu tab, and click on ‘Record Macro’ (you will be asked to give a name for the macro: name it ‘MyFirstMacro’)
  - Make the wanted operations
  - Click on ‘Stop Recording’.
Example

- Fill a range with values from 1 to 20 and put a heading ‘x’.
- Next to column x, create a column y with 20 random numbers
  (by inserting =RAND() in the cells)
- Rename the Sheet as ‘My Data’.
- Now we want to create a macro that
  - makes a copy of the values (not the formula) in the range of the x and y variables in a new sheet
  - renames the new sheet as ‘Outcome’
- Record a first macro
  - Go to the ‘Developer’ menu tab, and click on ‘Record Macro’
    (you will be asked to give a name for the macro; name it ‘MyFirstMacro’)
  - Make the wanted operations
  - Click on ‘Stop Recording’
Example

• Fill a range with values from 1 to 20 and put a heading ‘x’.
• Next to column x, create a column y with 20 random numbers (by inserting =RAND() in the cells)
• Rename the Sheet as ‘My Data’.
• Now we want to create a macro that
  • makes a copy of the values (not the formula) in the range of the x and y variables in a new sheet
  • renames the new sheet as 'Outcome'
• Record a first macro
  • Go to the ‘Developer’ menu tab, and click on ‘Record Macro’ (you will be asked to give a name for the macro; name it ‘MyFirstMacro’)
  • Make the wanted operations
  • Click on ‘Stop Recording’
Example

- Fill a range with values from 1 to 20 and put a heading ‘x’.
- Next to column x, create a column y with 20 random numbers (by inserting =RAND() in the cells)
- Rename the Sheet as ‘My Data’.
- Now we want to create a macro that
  1. makes a copy of the values (not the formula) in the range of the x and y variables in a new sheet
  2. renames the new sheet as ‘Outcome’

Record a first macro

1. Go to the ‘Developer’ menu tab, and click on ‘Record Macro’ (you will be asked to give a name for the macro; name it ‘MyFirstMacro’)
2. Make the wanted operations
3. Click on ‘Stop Recording’
Example

- Fill a range with values from 1 to 20 and put a heading ‘x’.
- Next to column x, create a column y with 20 random numbers (by inserting =RAND() in the cells)
- Rename the Sheet as ‘My Data’.
- Now we want to create a macro that
  1. makes a copy of the values (not the formula) in the range of the x and y variables in a new sheet
  2. renames the new sheet as ‘Outcome’

Record a first macro

1. Go to the ‘Developer’ menu tab, and click on ‘Record Macro’ (you will be asked to give a name for the macro; name it ‘MyFirstMacro’)
2. Make the wanted operations
3. Click on ‘Stop Recording’
Example

- Fill a range with values from 1 to 20 and put a heading ‘x’.
- Next to column x, create a column y with 20 random numbers (by inserting =RAND() in the cells)
- Rename the Sheet as ‘My Data’.
- Now we want to create a macro that
  1. makes a copy of the values (not the formula) in the range of the x and y variables in a new sheet
  2. renames the new sheet as ‘Outcome’
- Record a first macro
  1. Go to the ‘Developer’ menu tab, and click on ‘Record Macro’ (you will be asked to give a name for the macro: name it ‘MyFirstMacro’)
  2. Make the wanted operations
  3. Click on ‘Stop Recording’
Example

- Fill a range with values from 1 to 20 and put a heading ‘x’.
- Next to column x, create a column y with 20 random numbers (by inserting =RAND() in the cells)
- Rename the Sheet as ‘My Data’.
- Now we want to create a macro that
  1. makes a copy of the values (not the formula) in the range of the x and y variables in a new sheet
  2. renames the new sheet as ‘Outcome’
- Record a first macro
  1. Go to the ‘Developer’ menu tab, and click on ‘Record Macro’ (you will be asked to give a name for the macro: name it ‘MyFirstMacro’)
  2. Make the wanted operations
  3. Click on ‘Stop Recording’.
Example

• Fill a range with values from 1 to 20 and put a heading ‘x’.
• Next to column x, create a column y with 20 random numbers (by inserting \( \text{=RAND()} \) in the cells)
• Rename the Sheet as ‘My Data’.
• Now we want to create a macro that
  1. makes a copy of the values (not the formula) in the range of the x and y variables in a new sheet
  2. renames the new sheet as ‘Outcome’
• Record a first macro
  1. Go to the ‘Developer’ menu tab, and click on ‘Record Macro’ (you will be asked to give a name for the macro: name it ‘MyFirstMacro’)
  2. Make the wanted operations
  3. Click on ‘Stop Recording’.
Example

- Fill a range with values from 1 to 20 and put a heading ‘x’.
- Next to column x, create a column y with 20 random numbers (by inserting =RAND() in the cells)
- Rename the Sheet as ‘My Data’.
- Now we want to create a macro that
  1. makes a copy of the values (not the formula) in the range of the x and y variables in a new sheet
  2. renames the new sheet as ‘Outcome’
- Record a first macro
  1. Go to the ‘Developer’ menu tab, and click on ‘Record Macro’ (you will be asked to give a name for the macro: name it ‘MyFirstMacro’)
  2. Make the wanted operations
  3. Click on ‘Stop Recording’. 
Example

- Fill a range with values from 1 to 20 and put a heading ‘x’.
- Next to column x, create a column y with 20 random numbers (by inserting `=RAND()` in the cells).
- Rename the Sheet as ‘My Data’.
- Now we want to create a macro that
  1. makes a copy of the values (not the formula) in the range of the x and y variables in a new sheet
  2. renames the new sheet as ‘Outcome’
- Record a first macro
  1. Go to the ‘Developer’ menu tab, and click on ‘Record Macro’ (you will be asked to give a name for the macro: name it ‘MyFirstMacro’)
  2. Make the wanted operations
  3. Click on ‘Stop Recording’.
Running and editing the macro

- To run the macro, click on ‘Macros’ and select the macro you want to launch.
- As recorded, MyFirstMacro will end on an error. Clicking ‘debug’, will open the recorded VBA script and highlight the faulty line.
- Problem is that the newly created sheet has a different name than the one created when recording the macro (e.g., Sheet5 instead of Sheet4)
- Edit the macro to replace the faulty line with
  ```vbnet
  Sheets.Add
  ActiveSheet.Name = "Outcome"
  ```
  and simply delete any other line referring to ‘Sheet4’.
Deleting an existing sheet

- The macro works, but we need to first delete the existing ‘Outcome’ sheet when it exists.
- We automatize that with the following code

```vba
CurrentSheet = Format(ActiveSheet.Name)
For Each nm In ActiveWorkbook.Sheets
    If nm.Name Like "Outcome" Then
        If nm.Delete Then Exit For
    End If
Next nm
Sheets(CurrentSheet).Select
```
Creating a button for launching the macro

- Go on ‘My Data’ sheet.
- On the Developer tab, Open the ‘Insert’ list and select the top left ‘Button (Form Control)’
- A dialog will ask you to select a macro: select ‘MyFirstMacro’ and click OK
- Click on ‘Design Mode’, then with the right mouse button on the created button and select ‘Edit text’ on the menu which pops up. Replace the text with the macro name or whatever you want. Click again on ‘Design Mode’ when you are done.
- Now, just try it!
Looping through cells using Cells(i,j)

- To loop through cells use `Cells(i,j)`
- To illustrate, write a macro to collect in z and w columns respectively the x and y values of cases for which $y < 0.5$
- For the loop, use `for i = 3 to 22`
Sub MySelectMacro()
    ' MySelectMacro Macro
    Dim i As Integer
    Dim ii As Integer

    Cells(2, 5) = "z"
    Cells(2, 6) = "w"
    ii = 3
    For i = 3 To 22
        If Cells(i, 3) < 0.5 Then
            Cells(ii, 5) = Cells(i, 2)
            Cells(ii, 6) = Cells(i, 3)
            ii = ii + 1
        End If
    Next i
End Sub
Cells properties

- VBA is an object oriented language, with objects, properties and methods
  - This means that when applied to an object, the behavior of the method is determined according to the object properties.
- This is for instance of importance when assigning values to Cells or Ranges, where it may be of importance to distinguish between properties
  - `Cells.Value` Value in the cell
  - `Cells.Formula` Formula in the cell with standard referencing
  - `Cells.FormulaR1C1` Formula in the cell with R1C1 referencing
- R1C1 references can be turned in ‘Options’ (File tab), under ‘Formulas’.
Useful links

- Getting Started with VBA in Excel 2010

- Creating VBA Macros to Manipulate Worksheets in Excel 2007

- Excel VBA Easy (100 examples)
  http://www.excel-vba-easy.com/
