DSS: Lecture 2

(EBC 2088)

János Flesch

Department of Quantitative Economics
Maastricht University, The Netherlands

April 23, 2019

General Remarks:

- Size of a typical database:
  - Entries: between 100-200
  - Questions: between 15-20
- The schedule of the group meetings

VBA Security: Excel 2010 or 2007

File $\rightarrow$ Options $\rightarrow$ Trust Center $\rightarrow$ Trust Center Settings

$\rightarrow$ Macro Settings, and then:

- normally: "Disable all macros with notification" and "Trust access to the VBA project object model"
- when complete and want to run: "Enable all macros" and "Trust access to the VBA project object model"

The VBA Editor:

- Menu “Developer” in Excel:

  File → Options → Customize Ribbon → add “Developer”

- Shortcut to editor: ALT + F11

Saving the VB file:

In a macro enabled file type, normally with extension “xlsm”.
For example, “myDSS.xlsm”.

Remark: older versions of Excel use other extensions.

The "Project explorer":

- Excel objects
- Userforms – main part of programming
- Modules

Userforms:

USERFORMS:

- inserting a new userform

USERFORMS:

- inserting a new userform
- properties specified in the “properties window”

Userforms:

- inserting a new userform
- properties specified in the “properties window”
- for instance, changing caption (or even name)

USERFORMS:

- inserting a new userform
- properties specified in the “properties window”
- for instance, changing caption (or even name)
- the toolbox window

Label: to insert text on a userform

Available through the toolbox window.
You need to use SHIFT + ENTER for new line.

CommandButton: to execute commands after a click

CommandButton: to execute commands after a click

- inserting a CommandButton

CommandButton: to execute commands after a click

- inserting a CommandButton
- changing caption to “Next”

CommandButton: to execute commands after a click

- inserting a CommandButton
- changing caption to “Next”
- code of the CommandButton after double click:

Code: (now empty)

```
Private Sub CommandButton1_Click()

End Sub
```

Switching between two userforms:

Switching between two userforms:

Code for CommandButton in UserForm1:

Code:

```
Private Sub CommandButton1_Click()
    UserForm1.Hide
    UserForm2.Show
End Sub
```

Code for CommandButton in UserForm2:

Code:

```
Private Sub CommandButton1_Click()
    UserForm2.Hide
    UserForm1.Show
End Sub
```

Try to run it from UserForm1 (F5).

Displaying messages:

Displaying messages:

Creating a “Hello” button on UserForm1:

Code:

```
Private Sub CommandButton2_Click()
    MsgBox(“Hello world !”)
End Sub
```

InputBox:

InputBox:

- Asking for a name:

Code:

```
Private Sub CommandButton2_Click()
   InputBox("Please type your name")
End Sub
```

InputBox:

- Asking for a name:

Code:

```
Private Sub CommandButton2_Click()
    InputBox("Please type your name")
End Sub
```

- Using a name:

Code:

```
Private Sub CommandButton2_Click()
    inputname = InputBox("Please type your name")
    MsgBox("Hello " & inputname)
End Sub
```

InputBox:

- Storing a name:

Code:

```
Private Sub CommandButton2_Click()
    inputname = InputBox("Please type your name")
    MsgBox("Hello " & inputname)
    Sheets("UserData").Select
    Range("B2").Select
    ActiveCell.Value = inputname
End Sub
```

TextBox: to retreave info from the user

Storing text:

Code:

```
Private Sub TextBox1_Change()
    inputtext = TextBox1.Value
    Sheets("UserData").Select
    Range("B3").Select
    ActiveCell.Value = inputtext
End Sub
```

Alternative code:

```
Private Sub TextBox1_Change()
    inputtext = TextBox1.Value
    Range("Userdata!B3").Value = inputtext
End Sub
```

If-statements:

Structure:

```
If condition1 Then
    actions1
ElseIf condition2 Then
    actions2
...
Else
    actionsElse
End If
```

→ actions1 is executed if condition1 holds

→ actions2 is executed if condition1 does not hold but condition2 does

→ ...

→ actionsElse is executed if none of the conditions holds

If-statement example:

Code:

```
Private Sub CommandButton2_Click()
    UserName = Range("UserData!B2").Value
    If UserName = "John Doe" Then
        MsgBox ("You entered 'John Doe'")
    ElseIf UserName <> "" Then
        MsgBox ("Welcome, " & UserName)
    Else
        MsgBox ("You didn't enter your name")
    End If
End Sub
```

While-loops:

Structure:

```
While condition
    actions
Wend
```

It executes actions as long as condition holds.

Note: if condition does not hold in the beginning, then actions are not executed at all.

While-loop example:

Code:

```
Private Sub CommandButton3_Click()
    Sheets("UserData").Select
    i = 1
    While i <= 10
        Range("D" & i).Value = i * 2
        i = i + 1
    Wend
End Sub
```

For-loops:

Structure:

```
For counter = start To end Step step
    actions
Next
```

Step 1. Sets counter equal to start, and executes actions.

Step 2. Increases counter with step.

- If counter is at most end: executes actions, and repeats step 2.
- Otherwise, stops.

Note: actions executed at least ones. Also, if step equals 1, then it can be omitted.

For-loop example:

Code:

```
Private Sub CommandButton3_Click()
    Sheets("UserData").Select
    For i = 1 To 10
        Range("E" & i).Value = i * 2
    Next
End Sub
```

Equivalent code:

```
Private Sub CommandButton3_Click()
    Sheets("UserData").Select
    For i = 2 To 20 Step 2
        Range("E" & i / 2).Value = i
    Next
End Sub
```

CheckBox: can be clicked on or off

- Value of checkbox: on → TRUE, off → FALSE
- Adding three checkboxes to UserForm1
- Making sure that at least one is selected:

Code:

```
Private Sub CommandButton1_Click()
  If CheckBox1.Value + CheckBox2.Value + CheckBox3.Value = 0 Then
    MsgBox (“Please select at least one option”)
  Else
    UserForm1.Hide
    UserForm2.Show
  End If
End Sub
```

Note: FALSE = 0, TRUE = -1.

Equivalent code:

```
Private Sub CommandButton1_Click()
  If (CheckBox1.Value = False And CheckBox2.Value = False
   no line break And CheckBox3.Value = False) Then
    MsgBox ("Please select at least one option")
  Else
    UserForm1.Hide
    UserForm2.Show
  End If
End Sub
```

CheckBox:

Making sure that at most two are selected:

Code:

```
Private Sub CommandButton1_Click()
 If CheckBox1.Value + CheckBox2.Value + CheckBox3.Value = -3 Then
   MsgBox (“Please select at most two options”)
 Else
   UserForm1.Hide
   UserForm2.Show
 End If
End Sub
```

CheckBox:

Storing the value of the CheckBox (TRUE or FALSE):

Code:

```
Private Sub CheckBox1_Click()
    Sheets("UserData").Select
    Range("C2").Select
    ActiveCell.Value = CheckBox1.Value
End Sub
```

CheckBox:

Storing special values:

Code:

```
Private Sub CheckBox1_Click()
    Sheets("UserData").Select
    Range("C2").Select
    If CheckBox1.Value = True Then
        ActiveCell.Value = "Yes"
    Else
        ActiveCell.Value = "No"
    End If
End Sub
```

ToggleButton: same as CheckBox, looks different

- can be clicked on or off
- its value is TRUE or FALSE

OptionButtons: related to CheckBoxes

Difference: at most one optionbutton can be clicked "TRUE" per frame (not per userform).

Event Procedures:

These are code procedures associated with an event.

In the code window:

- upper-left corner: select the name of control
- upper-right corner: select the action of control

To perform a specific task, you can use a procedure. Two types:

FUNCTIONS: procedures that returns a value

SUBS: procedures that do not return a value

Functions: procedures that return a value

- They are stored in Module.
- They can be called from any sub procedure or another function.

Example of calling the square root:

Function code:

```
Public Function CalculateSquareRoot(number)
    CalculateSquareRoot = Sqr(number)
End Function
```

If the user is asked to enter a number, you have to verify the input before using this function.

The following code for a CommandButton yields a run-time error if the user enters -2:

Code:

```
Private Sub CommandButton1_Click()
    myvar = InputBox("Please enter a positive number")
    MsgBox ("The square root is " & CalculateSquareRoot(myvar))
End Sub
```

The following code verifies the input of the user:

Code:

```
Private Sub CommandButton1_Click()
    myvar = InputBox("Please enter a positive number")
    If (IsNumeric(myvar) And myvar > 0) Then
        MsgBox ("The square root is " & CalculateSquareRoot(myvar))
    Else
        MsgBox ("Wrong input!")
    End If
End Sub
```

For subs: Have a look at “subs.xlsm”.

References in the reader:

- Introduction: section 7, page 6
- Userforms: section 7.1, including 7.1.1
- Functions: section 7.2
- IF-statement and Loops: section 7.3
- Toolbox: within section 7.6
  - Label: subsection 1
  - CheckBox: subsection 2
  - ToggleButton: subsection 3
  - OptionButton: subsection 4
  - TextBox: subsection 5
- Event procedure: section 7.7