WORKSHOP DESCRIPTION

Overview

The first leg of the Excel Workshop series was designed primarily for those who had little experience with Excel. Participants were able to flavor the basics of a spreadsheet, create and modify worksheets, format and enhance worksheets, manage workbooks and setup their work in a presentable fashion.

The second leg of the series was designed to cover topics such as conditional formatting, creating templates, formulas and functions, charting techniques, mail merge and creating lists.

This the conclusion of this series designed primarily for Pivot Tables and Macros. To start off we will cover a general know how on Pivot Tables, followed by creation of a Pivot Table and a Pivot Chart. Later on the workshop concludes with recording macros, macro menus and global macros.

Prerequisites

- Excel 2007 Basics (optional)
- Excel 2007 Formatting (optional)

Objectives

- Learn about Pivot Tables and create them.
- Learn about Pivot Charts and create them.
- Record a Macro
- Go behind the scenes of a Macro
- Write a Macro.
WHAT IS A PIVOT TABLE

A PivotTable summarizes the information from selected fields of a data source. The source can be an Excel list or a relational database file containing similar data.

When you create a PivotTable, you specify which fields you’re interested in, how you want the table organized, and what kinds of calculations you want the table to perform. After you have built the table, you can rearrange it to view your data from alternative perspectives. This ability to “pivot” the dimensions of your table—for example, to transpose column headings to row positions—gives the PivotTable its name and its unusual analytical power.

One minor drawback to using PivotTables is that, unlike a formula based summary report, the data in the PivotTable does not update automatically. PivotTables are linked to the data from where they were derived. If the source is external, the PivotTables can be set to refresh at regular time intervals or when it’s being accessed.

Sample Example

The best way to understand the concept of a PivotTable is to see an example.

The example here shows a list of sales figures spread across two countries.

The list is organized by:

- Country,
- Salesperson,
- Order Date,
- OrderID and
- Order Amount.

The data spans 400 rows however with a few keystrokes you can turn this long list into a table that provides meaningful information to be viewed at a glance.
The PivotTable for the same data looks something like this:

<table>
<thead>
<tr>
<th>Sum of Order Amount</th>
<th>Order Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Salesperson</td>
<td>Qtr1</td>
</tr>
<tr>
<td>Buchanan</td>
<td>3237.12</td>
</tr>
<tr>
<td>Callahan</td>
<td>19992.23</td>
</tr>
<tr>
<td>Davolio</td>
<td>17885.82</td>
</tr>
<tr>
<td>Dodsworth</td>
<td>1310.8</td>
</tr>
<tr>
<td>Fuller</td>
<td>11434.38</td>
</tr>
<tr>
<td>King</td>
<td>15108.34</td>
</tr>
<tr>
<td>Leverling</td>
<td>28406.85</td>
</tr>
<tr>
<td>Peacock</td>
<td>41088.53</td>
</tr>
<tr>
<td>Suyama</td>
<td>5583.16</td>
</tr>
<tr>
<td>Grand Total</td>
<td>14407.23</td>
</tr>
</tbody>
</table>

The Salesperson column is positioned along the row axis, the Order Amount makes the body of the table while the Country is set as the selector on the top. The PivotTable shown here makes it easy to find almost all the information from the sample data shown before. The Order Date and OrderID are not visible in the PivotTable however can be added in if required.

The dropdown option for the Country category allows you to view the data for a specific country. Likewise you can narrow down the option for a particular Salesperson as well.

---

**PivotTable Terminology**

Understanding the terminology associated with pivot tables is the first step in mastering this feature.

- **Column Field**: Fields of data
- **Data Area**: Cells that contain summary data.
- **Grand Totals**: Totals of cells in a row or column
- **Item**: An element in a field that appears as a row or column
- **Group**: Collections of Items
- **Page Field**: Fields of data
- **Refresh**: To recalculate the PivotTable after any changes have been made to the source data.
- **Row Field**: Fields of data
• Source Data: Data that was used to create this PivotTable.

• Sub Totals: Sub total of cells in a row or a column.

---

**Creating a PivotTable**

Here we’ll create a PivotTable from the sample list (Salespeople_Example.xls) from the Workshop Series 3 folder.

To create a PivotTable from an Excel source,

1. To begin, click on a cell within your Data Source.
2. Then choose from the ribbon, Insert > Pivot Tables.
3. This displays the Create Pivot Table dialog box, prompting you to options these four steps…
   a. To specify the type of data source on which the table will be based and whether you want to create a PivotTable or a PivotChart Report.
   b. To specify the location of your source data.
   c. To specify where you want the PivotTable to appear.
   d. Click OK to continue.

You will notice two new tabs appear in the Ribbon under PivotTable Tools, as shown below.
Layout of the PivotTable

Now all that’s left for you to do is to check the fields that you would like to add to the report. Another way to do this would be to drag the fields and drop them into the four quadrants below the PivotTable field-list, or drag them directly into one of the PivotTable zones.
Pivoting a PivotTable

To pivot, or rearrange, a PivotTable, drag one or more field headings. To move a field from the column axis to the row axis, for example, all you have to do is drag its heading from the column area to the row area.

In addition to transposing columns and rows, you can change the order in which fields are displayed on the column or row axis. For example, you can drag a heading to the left.
Refresh a PivotTable

PivotTables are not updated each time a change occurs in their source data.

- To manually update a table, select any cell in the table and choose Options, Refresh on the Ribbon.
- If you want Excel to refresh your PivotTable every time you open the workbook in which it resides, choose from the Ribbon, Options, Pivot Table Options. Then select the Refresh Data When Opening the File check box in the PivotTable Options dialog box under Data tab. If you want to prevent Excel from updating the table each time you open the workbook, be sure this check box is cleared.

Grouping and Ungrouping Data

PivotTables group inner field items under each outer field heading and, if requested, create subtotals for each group of inner field items. You might find it convenient to group items in additional ways. Excel provides several options for grouping items.
Grouping Numeric Items

To group numeric items in a field:

**step 1.** Select a numeric item in the field & choose **Options > Group Field**.

**step 2.** You’ll see a dialog box similar to the one shown, but tailored for the numeric range of your own data. Fill in the Starting at, Ending at, and By values as appropriate.

Grouping Items in Date or Time Ranges

To make this kind of table more meaningful, you can group the date field:

**step 1.** Select a date item in the field & right-click to Group and Show Detail, & choose Group.

**step 2.** Then fill out the Grouping dialog box as shown.

**step 3.** Excel gives you a great deal of flexibility in the way your data and time fields are grouped. In the By list, you can choose any common time interval from seconds to years, and if the standard time intervals don’t meet your needs, you can select an ad hoc number of days.
Removing Groups
To remove any group and restore a field to its ungrouped state:
select a grouped item, choose Options > Ungroup.

Creating a PivotChart
You can create a PivotChart by choosing PivotChart under PivotTable Tools, Options and Tools.
Notice that you can rearrange a PivotChart exactly as you would a PivotTable—by dragging field headings from one axis to another. To add fields, drag them from the PivotTable Field List. To remove fields, drag them off the chart. To limit the display to particular items in a field, select those items in the field’s drop-down list.
EXERCISE 1
PIVOTTABLE & PIVOTCHART

In this exercise, you will practice creating a PivotTable and a PivotChart by using the techniques learned till now.

1. From the Workshop Series 3 folder on your desktop, open the file named exercise1.xls.

2. Create a PivotTable that can give you an up-to-date information on the total expenses for the various Funds.

3. Summarize the fund distribution for each vendor.

4. Save the file.

5. Next, create a PivotChart on the data from the same file exercise1.xls.

6. Save the PivotChart as a new Worksheet

7. Save your file (using the same filename).
WHAT IS A MACRO

A macro is a set of instructions that tells Microsoft Excel to perform one or more actions for you. Macros are like computer programs, but they run completely within Excel. You can use them to automate tedious or frequently repeated tasks.

Macros can carry out sequences of actions much more quickly than you could yourself. For example, you can create a macro that enters a series of dates across one row of a worksheet, centers the date in each cell, and then applies a border format to the row. Or you can create a macro that defines special print settings in the Page Setup dialog box and then prints the document.

Macros can be simple or extremely complex. They can also be interactive; that is, you can write macros that request information from the user and then act on that information.

There are two ways to create a macro: You can record it, or you can build it by entering instructions in a module. Either way, your instructions are encoded in the programming language Microsoft Visual Basic for Applications (VBA).

Recording a Macro

Rather than type macros character by character, you can have Excel create a macro by recording the menu commands, keystrokes, and other actions needed to accomplish a task. After you’ve recorded a series of actions, you can run the macro to perform the task again. As you might expect, this playback capability is most useful with macros that automate long or repetitive processes, such as entering and formatting tables or printing a certain section of a worksheet.

When you record a macro, all steps that are needed to complete the actions that you want to record are recorded by the macro recorder. Navigation on the Ribbon is not included in the recorded steps.

NOTE: The Ribbon is a component of the Microsoft Office Fluent user interface.

step 1. If the Developer tab is not available, do the following to display it:
   a. Click the Microsoft Office Button, and then click Excel Options.
b. In the *Popular* category, under *Top options for working with Excel*, select the *Show Developer* tab in the Ribbon check box, and then click *OK*.

**step 2.** To set the security level temporarily to enable all macros, do the following:

*a.* On the *Developer* tab, in the *Code* group, click *Macro Security.*
b. Under Macro Settings, click Enable all macros (not recommended, potentially dangerous code can run), and then click OK.

NOTE: To help prevent potentially dangerous code from running, we recommend that you return to any of the settings that disable all macros after you finish working with macros.

step 3. On the Developer tab, in the Code group, click Record Macro.

step 4. In the Macro name box, enter a name for the macro.

NOTE: The first character of the macro name must be a letter. Following characters can be letters, numbers, or underscore characters. Spaces are not allowed in a macro name; an underscore character works well as a word separator. If you use a macro name that is also a cell reference, you may get an error message that the macro name is not valid.

step 5. To assign a CTRL combination shortcut key to run the macro, in the Shortcut key box, type any lowercase letter or uppercase letter that you want to use.

NOTE: The shortcut key will override any equivalent default Excel shortcut key while the workbook that contains the macro is open. For a list of CTRL combination shortcut keys that are already assigned in Excel, see Excel shortcut and function keys.

step 6. In the Store macro in list, select the workbook in which you want to store the macro.
**TIP:** If you want a macro to be available whenever you use Excel, select **Personal Macro Workbook**. When you select Personal Macro Workbook, Excel creates a hidden personal macro workbook (Personal.xlsm) if it doesn't already exist, and saves the macro in this workbook. In Windows Vista, this workbook is saved in the C:\Users\username\AppData\Local\Microsoft\Excel\XLStart folder. In Microsoft Windows XP, this workbook is saved in the C:\Documents and Settings\username\Application Data\Microsoft\Excel\XLStart folder. Workbooks in the XLStart folder are opened automatically whenever Excel starts. If you want a macro in the personal macro workbook to be run automatically in another workbook, you must also save that workbook in the XLStart folder so that both workbooks are opened when Excel starts.

**step 7.** To include a description of the macro, in the *Description* box, type the text that you want.

**step 8.** Click **OK** to start recording.

**step 9.** Perform the actions that you want to record.

**step 10.** On the Developer tab, in the Code group, click **Stop Recording**.

**TIP:** You can also click **Stop Recording** on the left side of the status bar.

---

**Creating a Macro**

Let’s create a simple macro that inserts your Department name and University Logo in a worksheet.
**step 1.** Begin by opening a new file and saving it to the desktop as **Macro.xls**.

**step 2.** Choose Developer, Code, and Macro which then displays the Record Macro dialog box.

**step 3.** Assign a name to the macro. The default is Macro1 or you can enter your own name. Let’s use **UniversityLogo**. Note that this name cannot have any spaces.

**step 4.** Assign a key combination to the macro by entering a letter—in this case, uppercase L – in the Shortcut key edit box.

**step 5.** Store the macro in the currently active workbook by making sure the **This Workbook** option is selected.

**step 6.** Enter a description for the macro in the **Description** box; in this case, type **Insert University Logo**.

**step 7.** To begin recording, click OK. Excel displays the message Recording in the status bar.

**step 8.** Select Page Layout > Page Setup dialog box.

**step 9.** Select the Margins tab and change the **Top Margin** to 1.75.

**step 10.** Select the **Header/Footer** tab and click the **Custom Header** button.

**step 11.** To the Left section, add the **CSUS Logo**, to the Center section add the **University Name** and your **Department Name**. Click **OK**.

**step 12.** Similarly add the **Page number** to the left section and **File path** to the right section of the **Footer**. Then click **OK**.

**step 13.** Click **OK** to complete Page Setup.

**step 14.** Click the **Stop Recording Macro** button on the Developer tab..

**step 15.** This step is **IMPORTANT**; if you don’t stop the macro recorder, Excel continues to record your actions indefinitely.

**step 16.** To test the new macro, open a new worksheet. Type your name in cell A1 and then press **Ctrl+Shift+L**. Excel runs the macro and performs the sequence of actions in the same way you recorded them. Do a Print Preview!!
Run a Macro Without Using Keyboard Shortcuts

You don’t have to know a macro’s key combination to run the macro. Instead, you can use the Macro dialog box:

*step 1.* Choose Developer, Code, and Macros to display the dialog box.
*step 2.* Select the name of the macro, and click Run.
*step 3.* You also can use the Macro dialog box to view and edit macros, as you’ll see in the next section.

---

**The VBA Environment**

Now that you’ve recorded your macro, let’s find out what Excel did. When you clicked OK in the Record Macro dialog box, Excel created something called a module in the active workbook. Excel recorded your actions and inserted the corresponding VBA code in the module.

- Choose Developer, Code, and Macros to display the dialog box. Click the Edit button.
As you can see, a module looks like a window that you might see in a word processing program. The menu bar above the module includes menus for editing, debugging, and running VBA code. In the module you can review, enter, copy, move, insert, and delete VBA statements and comments using techniques that are similar to those you use in a word processing program. The VBA environment is a big place, full of interesting details, but for now let’s focus only on the code we’ve recorded.

On the right side of the VBE is a window displaying the module containing your code. The first and last lines of the code act as the beginning and endpoints for the macro you’ve recorded; a Sub statement starts the macro and an End Sub statement ends the macro. You’ll notice that special VBA terms, called keywords, are displayed in dark blue.

Here is an explanation…

- **SUB**: This is a keyword that begins a Macro and is followed by the name of the Macro.
- **COMMENTS**: The comments are entered following an Apostrophe. VB ignores the apostrophe.
- **STATEMENTS**: Each action carried out by the Macro is written in statements. The syntax of a statement is an Object followed by the Action. Here are some examples:
  - **RANGE(“A6”).Select**: The Object is the range reflecting cell A6 and the Action taken is Select.
  - **SHEETS(3).ACTIVATE**: An action is something that the object knows how to perform and so is known as the Method of the object.
- \texttt{SELECTION.FONT.NAME = "ARIAL"}: An action can contain an equal sign which usually is followed by the characteristic or an attribute of the object. This is termed as Property of the Object.
- \texttt{END SUB}: This is a keyword that ends the Macro.

---

**Editing a Macro**

Suppose you’ve recorded a macro that enters your name and address. Then you discover that you forgot a step or that you recorded a step incorrectly—such as you chose the wrong border format, for example.

What do you do?

To add code to an existing macro, you can record actions in a temporary macro and then transfer the code into the macro you want to change. For example, to add to the UniversityAddress macro a step that sets font options for the address, follow these steps:

\begin{itemize}
\item \textbf{step 1.} Open the \texttt{macro_example.xls} workbook that contains an address.
\item \textbf{step 2.} Choose Developer, Code, and Record Macro. Excel presents the Record Macro dialog box. In the \textit{Macro Name} box, enter \texttt{MacroTemp} and click \texttt{OK}.
\end{itemize}
step 3. Choose Home, Cells, Format, and Format Cells and click the Font tab. Select Arial, 14-point, and Bold Italic. Then click OK to apply the formats.

step 4. Click the Stop Recording button in the Developer tab.

step 5. Choose Developer, Code, and Macros. In the Macro dialog box, select MacroTemp and click Edit.

step 6. A window appears that contains the original macro recorded plus the MacroTemp macro.

step 7. Select all the code inside the MacroTemp—from the line beginning With through the line beginning End With—and then choose Edit, Copy.

step 8. Scroll up to display the UniversityAddress macro.


step 10. Press Enter to create a blank line. Then position the insertion point at the beginning of the blank line.


step 12. Scroll back down and delete the entire MacroTemp macro, from the Sub statement to the End Sub statement.

step 13. Run the original macro and see the difference.

---

Writing your own Macro using the VB Editor

step 1. Open the workbook Layout.xls

step 2. Choose Developer, Coder, Visual Basic Editor

step 3. Choose Insert Module

step 4. Type the following in the module area

```
Sub LayoutPortrait()
   'Macro to change the layout to Portrait
   'Keyboard Shortcut: Ctrl+Shift+P
   With ActiveSheet.PageSetup
       .Orientation = xlPortrait
   End With
End Sub
```

step 5. Save the Macro

step 6. Open Excel, and go to Tools – Macro – Macros

step 7. Select the newly entered Macro and click the Options button.

step 8. At the prompt for Shortcut Key type P

step 9. Click OK and close the Macro window.

a. Test your Macro by choosing the sheet Expense Statement.

b. View the sheet via Print Preview and notice that the document is in Landscape

c. Now run your macro.

d. View the sheet via Print Preview
EXERCISE 2
MACRO CREATION

In this exercise, you will add a new Macro.

1. You can create the Macro either by using the standard method or by using the VB editor.

2. The file to work on is exercise2.xls

3. Change the appearance of the document by updating the font to Comic Sans MS, size 11, italic and column width of 20.

4. Save the Macro and execute it.
RESOURCES

**Faculty / Staff Resource Center**
Located in ARC 3012. Assistance available on walk-in basis.
Open Lab on Fridays, 1-4 pm (Fall, Winter, Spring)
Open Lab on Thursdays 1-4 pm (Summer only)
FSRC Website
http://www.csus.edu/uccs/fscenter/

**Getting Help**

University Help Desk
(916) 278-7337 or helpdesk@csus.edu

Academic Technology Consultants
http://www.csus.edu/atcs/contact.htm

Help Desk - Problem Reports & Contact Information
http://www.csus.edu/uccs/helpdesk/contact.stm

Training Requests
training@csus.edu

**Campus Resources**

Training
http://www.csus.edu/training

Training Handouts
http://www.csus.edu/training/handouts

Online Tutorials
http://www.csus.edu/atcs/tools/tutorials.htm

Educational Tools
http://www.csus.edu/atcs/tools

Accessibility at Sacramento State
http://www.csus.edu/accessibility