Using Excel to Analyze Experimental Data, Part II

Tips and Tricks that Simplify Handling Large Amounts of Data

May 22, 2013

NASA Advanced Supercomputing Division
Preface

• The webinar assumes basic knowledge of Excel
• The advice here is somewhat unconventional and is influenced by:
  - The author’s computer science background
  - The author’s experiences with large workbooks for analyzing
    • Benchmarking data
    • PBS usage data
• There is a lot of material here
  - You may need to invest some time after the talk to experiment with the techniques described
  - To help with that there is an accompanying workbook:
    • Excel_Webinar_Examples.xlsx
• Some of the techniques here are Excel-specific
  - Some will not work with OpenOffice, Numbers (Apple), or Google Tools
Recap of first webinar

1. Useful to think of Excel as a **functional** programming language
2. Large tables of similar records a useful paradigm
   - Columns with computations can be constructed with copy & paste
3. Pivot tables are useful for assigning data records into buckets and then performing a data reduction operation on each bucket
4. A useful technique to improve reliability and facilitate maintenance is to use multiple sheets to separate data from formulas
   - Can “copy” data into formula sheet with `=INDIRECT(ADDRESS(...)`) 
   - For identical analysis of multiple datasets:
     - For each dataset, one sheet for each of:
       - Experimental data, organized as a table of records with a header
       - Parameter sheet for analysis parameters (optional)
       - Formula sheet that pulls data values and per-experiment parameters
     - Sheets are named using a pattern
       - Allows data & parameter sheet names to be calculated on formula sheet
       - (see sheets with names beginning with “Exp” in examples workbook)
Q & A from Part I

1. When separating data from formulas, what’s the best way to modify multiple formula sheets?
   - From Part I: delete all but one; make changes; make copies
   - Alternative (if not making changes to graphs or pivot tables):
     • Select all formula sheet tabs
     • Make editing changes; changes will occur on each selected sheet
       - Assumes all formula sheets are identical

2. ?
Looking up Values in a Table

- Suppose you have a table of data values, e.g.,

<table>
<thead>
<tr>
<th>day_id</th>
<th>job_id</th>
<th>uid</th>
<th>gid</th>
<th>queue</th>
<th>MAUs</th>
<th>SBUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>20110701</td>
<td>1068006</td>
<td>11127</td>
<td>40836</td>
<td>normal</td>
<td>8</td>
<td>11.6</td>
</tr>
<tr>
<td>20110701</td>
<td>1068007</td>
<td>11127</td>
<td>40836</td>
<td>normal</td>
<td>8</td>
<td>11.6</td>
</tr>
<tr>
<td>20110701</td>
<td>1068102</td>
<td>11127</td>
<td>40836</td>
<td>normal</td>
<td>8</td>
<td>0.3</td>
</tr>
</tbody>
</table>

and you want to lookup items there for another table:

<table>
<thead>
<tr>
<th>host</th>
<th>PBS</th>
<th>gid</th>
<th>nodes</th>
<th>jobname</th>
<th>mem(k)</th>
<th>wait hrs</th>
<th>run hrs</th>
<th>job_id</th>
<th>queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>columbia22</td>
<td>1121693.pbs1.0</td>
<td>s0836</td>
<td>128</td>
<td>nbody</td>
<td>970,915,904</td>
<td>0</td>
<td>45.93</td>
<td>1121693</td>
<td>vlong</td>
</tr>
<tr>
<td>columbia24</td>
<td>1121703.pbs1.0</td>
<td>s0836</td>
<td>128</td>
<td>nbody</td>
<td>926,170,816</td>
<td>0</td>
<td>10.35</td>
<td>1121703</td>
<td>vlong</td>
</tr>
<tr>
<td>columbia22</td>
<td>1091589.pbs1.0</td>
<td>s0836</td>
<td>128</td>
<td>nbody</td>
<td>916,032,832</td>
<td>139.65</td>
<td>100.1</td>
<td>1091589</td>
<td>vlong</td>
</tr>
</tbody>
</table>

- Use the **VLOOKUP(x, table, c, approx)** function
  - lookup x in 1st column of table
  - return item in column c
Trick #5: Sorting as a Functional Operation

- Not talking about **Data:Sort** (that is *imperative*); instead:
  - Have unsorted data, say on experimental data sheet
  - Want formulas to produced sorted copy of that data:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>rank of col. E</td>
<td>header1</td>
<td>header2</td>
<td>header3</td>
<td>header4</td>
<td>0</td>
<td></td>
<td>header1</td>
<td>header2</td>
<td>header3</td>
<td>header4</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>data11</td>
<td>data12</td>
<td>6</td>
<td>data14</td>
<td>1</td>
<td></td>
<td>data21</td>
<td>data22</td>
<td>3</td>
<td>data24</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>data21</td>
<td>data22</td>
<td>3</td>
<td>data24</td>
<td>2</td>
<td></td>
<td>data31</td>
<td>data32</td>
<td>4</td>
<td>data34</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>data31</td>
<td>data32</td>
<td>4</td>
<td>data34</td>
<td>3</td>
<td></td>
<td>data11</td>
<td>data12</td>
<td>6</td>
<td>data14</td>
</tr>
</tbody>
</table>

**A2** is \(=\text{RANK.EQ}(E2,E$2:E$4,1)\)

**I2** is \(=\text{VLOOKUP}($H2,$A$2:$F$4,COLUMN()-COLUMN($F2),\text{FALSE})\)

- Where:
  - **RANK.EQ(x,range,order)** is the rank of \(x\) in *range* (*order*=1 for ascending)
  - Other cells in same color region can be pasted from **A2**, **H2**, and **I2**
  - (See “Sort” sheet of examples workbook)
Reductions across Multiple Sheets

- Excel sheets are (obviously) 2-dimensional
- You can get 3-D “arrays” by using multiple sheets
- (Some) functions allow 3-D ranges to be supplied
  - `SUM('Sheet1:Sheet3'!A1:D4)`
  - Similarly: `PRODUCT`, `MAX`, `MIN`, `AVERAGE`

- Suppose you have a collection of experiments on sheets named: `Exp1 Exp2 … ExpN`
- Tip: put in empty “sentinel” sheets on either side:
  - `AllExp< Exp1 Exp2 … ExpN >`
- Can average location D3 across all experiments with:
  - `=AVERAGE('AllExp<:>'!D3)`
    (the empty cells don’t cause a problem)
- Adding new experiment sheets between the sentinels will cause update
- (see “Sum over Sheets” in example workbook)
Array Formulas

• Suppose you want to compute a weighed average:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>weight</td>
<td>case I</td>
<td>case II</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
<td>1.0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>2.0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>C</td>
<td>1.0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>weighted average</td>
<td>1.50</td>
</tr>
</tbody>
</table>

• Could compute C5 with:

\[ \text{=SUMPRODUCT(B2:B4,C2:C4)} / \text{SUM(B2:B4)} \]

• But if we want to ignore blank values (e.g. D3):

\[ \{=\text{SUMPRODUCT(B2:B4,D2:D4)} / \text{SUM(IF(ISNUMBER(D2:D4),B2:B4,0))}\} \]

  - An “Array Formula”
  - Type control-shift-return instead of return when entering
  - (see “Array Formulas” in examples workbook)
  - Note: IF(condition, then-value, else-value) is a conditional expression; e.g. IF(A1=A2, 1, 0)
Approaching the Dark Side: Visual Basic

- Visual Basic for Applications (VBA)
  - Imperative-style programming in a BASIC-like language
  - Can simplify very complicated formulas
  - Can provide functionality not available otherwise
    - E.g. “name of worksheet #n”

- Recommendations:
  - In general, stick to functional programming style
    - I.e., no side effects in the VBA code!
  - Exception might be to provide macro-like support for administering workbook
    - E.g., updating program sheets from a template when changes have been made

- Note: workbook will need to be .xlsm file (it has “macros”)
Adding User-Defined Functions in VBA

• Enable **Developer** ribbon items
  – (Mac) **Preferences**:Ribbon check **Developer** box in list
• Click **Developer** ribbon item
• Click **Editor** item
• Click
  – Then click **Module** and editor window pops up:
• Type VB code in window, e.g.
  
  ```vbnet
  Public Function WorksheetName(num As Long) As String
    If Worksheets.Count < num Then
      WorksheetName = ""
    Else
      WorksheetName = Worksheets(num).Name
    End If
  End Function
  ```
  – Use in formula as: `=WorksheetName(7)` which returns name of 7th sheet

Question? Use the Webex chat facility to ask the Host
Issue: Graphs & Pivot Tables on Formula Sheets

• Issue with copying a formula sheet with a graph or a pivot table
  - Graph/table will copy, but will refer to data on original sheet not the new sheet 😞

• Workaround:
  - Copy the template worksheet to a new workbook
  - Copy the new worksheet back to original book
    • Repeat as needed to get all the copies you need
  - Rename the new sheets as appropriate
  - With a pivot table: must also “Change Source” to refer to new sheet name