CITS2401
Computer Analysis & Visualisation

Lectures 3 and 4
Analysis in Excel
Agenda

- Charts in Excel
- Graphical Representation of Data.
- Importing data into Excel
- Data Types
- Arrays and Array Formula
- Spreadsheet Macros
Charts

- A chart is a graphical representation of the data in a worksheet.
- A chart summarizes the numerical data (often large, sometimes vast) in a visual form.
- Spreadsheets provide tools to create a chart from worksheet data. The Chart Ribbon provides a number of options and menus for producing charts from spreadsheet data.
- A chart can be embedded in the same sheet or on a separate chart sheet.
- You can use many different types of charts, and apply many effects to a chart, to present your data in the most appropriate way.
- All tools we examine in this unit have the facility to produce charts and visualisations, and they provide a useful yardstick for comparing different tools.
Chart Ribbon

• A series of options and menus that guide you through the chart-building process
• Decide whether or not the chart should be on the current sheet or on a separate chart sheet

• Open a workbook
• Define the cells to be used to include in the chart
• Select the Chart tab and choose the desired type of plot
• Right click on the chart to define and modify options.
Changing charts

- A chart is an object composed of a number of different objects.
- Changes can be made to a chart only after it has been selected (grab handles visible). Single-click to select chart.
- Right button click and select Edit. The chart will now have a grey border.
- Right button click activates a drop down menu, with a number of options to change the chart.
- Equivalently change the chart options using the Format menu.
Changing data ranges

◊ The existing worksheet range used to generate the graph can be modified by selecting the chart with a left button click. Then using a right button click (again) select “Modify Data Range”

◊ Redefine the range of cells to be included in the chart.

◊ The chart automatically changes to reflect the change.

◊ If named ranges are used, changing the range in the name definition automatically changes charts, and anything else dependent on that name!
Common chart types: Histograms

- Uses Bars to relate quantities (often cardinal values) to elements of a set (the domain set).
- Options to plots several series at once, stack columns, apply lines of fit.
- The domain set should be discrete (non-continuous). Otherwise a line graph may be more appropriate.
Common chart types: Line graphs

- Line graphs also represent a function from a domain set, except they imply the domain and the range are continuous.
- When modelling data we often seek to find a continuous function to approximate the data.
Pie Charts are useful for representing relative proportions. Elements are represented as fractions of the whole, and are only suitable for cardinal data.

A variety of different looks and effects are possible.
Common chart types: Scatter Plots

◊ Scatter plots show the correlation between two dimensions of the data. They are often used to demonstrate or suggest good models for data.

◊ They also may be used when you have incomplete domains.
Common chart types: Surface Plots

- Surface plots have a 2-dimensional domain and represent continuous data. They’re often used for characterizing the behaviour of complex systems, or visualizing optima or points of stability.
Other chart types….

There are a variety of other chart types supported by excel:

◊ Radar charts give a good visualisation of relative compositions
◊ Stock charts and error bars can be used to visualize maxima and minima.

Experiment with these, and the different options accessible through the ribbon.

Remember to think critically about what data you are trying to visualise and what chart is best suited to the purpose.

Don’t be tempted by the exotic Excel chart options
What makes a good chart?

• Label axis
• Include units
• Descriptive title
• Axis scale
• Include source of data

Measured distance a projectile travels in time
Common mistakes

Measured distance a projectile travels in time

Not clearly labelling multiple data sets or providing a legend
Not clearly labelling multiple data sets or providing a legend
Common mistakes

Measured distance a projectile travels in time

Connecting the dots in experimental data

In Excel use ‘Scatter’ plot instead of ‘Line’ plots
Common Mistakes

Two much significance given to variation by choice of scale

Excel defaults to this… and it shouldn’t!!!!
Common Mistakes

Two much significance given to variation by choice of scale

Usually start vertical axis at zero!

RP Data-Rismark Daily Home Value Index values over last 365 days 27/02/2013 - 29/02/2012 Combined Dwellings Perth
Common Mistakes

Finding a correlation when there is none

Random number generator

y = 0.0425x + 0.1281
Common Mistakes

**FIGURE A.** Trend in NAEP reading average scores for 9-, 13-, and 17-year-old students

Non Linear scale

http://www.bipps.org/tag/testing/page/2/
Common mistakes

% distribution of students in class by gender

Using a chart when data is simple enough to understand without one
Chart too complex

Consider redesign or separate in multiple charts

Clarify message

http://sandeen.net/wordpress/energy/a-day-in-the-life-of-a-net-meter/
Common mistakes

Connecting unrelated categories

Number of students with each eye colour in kindergarten class

- Blue
- Brown
- Green
- Black

Boys
Girls
Common mistakes

Distortion introduced by using 2D and 3D objects incorrectly
Common mistakes

◊ Distortion introduced by 3D perspective
3D plots are best avoided altogether

◊ Consider a contour plot instead.
Common mistakes

Graph junk
Misleading by hiding important information needed to properly interpret the graph.
Graphs summary

◊ Be clear
◊ By unambiguous
◊ Don’t ‘pimp my chart’
◊ Ask yourself: How long would it take someone to understand what this chart is about?
Types of Data

◊ We have already seen some of the types of Data that Excel supports: Dates, Integers, Currency, Logical values.

◊ When analysing data it is important that you understand the type of data, and also the meaning of that data.

◊ In fact, by analysing only the dimensions and type of data we can go a long way towards building a complete model of that data (this is called *dimensional analysis*).

◊ For now we will focus on just the types and meaning of data.
Excel Data Types

- Excel Formats Data according to a set of types.
- More generally, we can classify data as:
  1. Set elements (Names, words)
  2. Ordinal Numbers (1st, 2nd etc)
  3. Cardinal Data (sizes, always non-negative integers)
  4. Scalar Data (time, distance, etc)
  5. Probabilities (always between 0 and 1)
  6. Boolean/Logical (true or false)
  7. Functions
  8. Several others....
Semantics of Data

Semantics are the study of meaning. When Analysing and Visualising data, we have to be aware of the meaning of data, its implication for the analysis, and it is represented in the visualisation.

◊ The type of each data point determines what operations are suitable to apply.
◊ Set elements are typically unordered, with no arithmetic operators.
◊ Ordinal numbers have an ordering, but arithmetic is meaningless.
◊ Addition and subtract can make sense for cardinal numbers, but multiplication less so*.
◊ Arithmetic is meaningful for scalar data, but not for logical data.
◊ Addition is rarely meaningful for probabilities.
What types are the following data?

- 60 miles per hour
- 13 to 1 against
- CITS2401
- The number of students in CITS2401
- May 4\textsuperscript{th} 1977
- 12 kilometers
- 31.9554° S, 115.8585° E
- 7<6
- 2 hours 10 minutes
- Your final exam mark
Composite Data Types

◊ Data types may be combined to produce more complex data.

◊ For example, a function is a map from a Domain set to a Codomain set. Your timetable is a function from a set of times, to a set of classes.

◊ Relations link elements from two sets: e.g. the Friend relationship in Facebook links two Facebook users.

◊ Multi-dimensional data combines values from several sets into a single data-point: e.g. the status of a car might include its location (x and y coordinates), its velocity (x’ and y’ coordinates), its engine temperature (degrees Celsius), amount of petrol remaining (litres) etc.
How do we get data into excel?

◊ Copy and paste
◊ ‘Import’ option under the file menu
◊ Data file typically has data arranged in columns separated by space, tab or comma.
◊ Follow import wizard.
How do we get data into excel?
Array Formulae in Spreadsheets
Arrays

◊ A powerful feature of modern spreadsheets is working with arrays in formulae.

◊ The concept of an array is fundamental in most computer programming. *An array is a container or collection of items, and these items can be operated on individually or as a collection.*

◊ Arrays can be one- or two-dimensional, these dimensions corresponding to the rows and columns.

◊ A 1D array can be a range consisting of one row or column of cells. A 2D array is a rectangular range of cells.
• We restrict ourselves to 1D arrays, for simplicity only.

• **Arrays need not be stored in cells, and they can exist in the applications memory.**

• In Excel the memory is used for storage unless complex nested calculations are performed. In those cases you need to store the results of interim calculations somewhere in the workbook. The current discussions amongst developers focusses on how extensive the array formulae support needs to be in future releases.

• The results of an array formula may occupy a *single* cell or *multiple* cells. That is, the result of the array formula can be a scalar (single) value or an array.
A multi-cell array formula

• The product of two cells is given by =A1*B1, and input in cell C1 (say). For the products A1 to A7, with the corresponding B1 to B7, we make cell C1 active and fill down. Automatic updating results in =A2*B2 in the adjacent cell and so on. *Note this method results in 7 different formulae in the 7 cells C1 to C7.*

• An equivalent way uses a *single* array formula. We first select C1 and type =A1:A7*B1:B7 (which will initially appear in cell C1). As this is an array formula we press *Control-Shift-Enter* rather than *Enter*. The exact formula above, held between *{braces}*, appears in cells C1 to C7 and gives the same results as previously.
How did it work?

• Because you actually wish to perform this multiplication element by element you instruct Excel to do this by pressing Control-Shift-Enter and the formula becomes \{=A1:A7*B1:B7\}. Excel places the brackets to indicate an array formula is being evaluated.

• What is actually happening? Excel is taking a 7 element array (A1:A7), and multiplying it by another 7 element array (B1:B7), resulting in another 7 element array, which is placed in as many cells as needed.
The advantages of an array formula

The second technique resulted in a single array formula in the range of cells, whereas the first technique had 7 different formulas in the range of cells. What advantage did this provide?

- It ensures all formulas in the range are identical
- You cannot easily overwrite the cell formulas, because you cannot change one cell in a multi-cell array formula.
Arrays and named ranges

The previous examples could be made easier by giving the cell range references a name and using those as in 
\{=Score*Weight\}.

When working with arrays you don’t need to use Control-Shift-Enter everywhere. It is only when the formula requires element by element operation in the arrays that Control-Shift-Enter need be employed.

There are many functions that expect an array in its argument eg. SUM(), AVERAGE() etc. Hence the formula =AVERAGE(Score) needs only Enter, but =AVERAGE(Score*Weight) needs Control-Shift-Enter because the array multiplication must be executed first.
The normalised vector calculation, ie. the elements of the vector divided by the square root of the sum of squares of the vector elements!


where SQRT returns the square root of a number, and SUM adds up the elements of the array. 
SUM(1;2;3) is 6.
If cells A1:A3 held the values 1, 2 and 3 respectively, then SUM(A1:A3) is 6.
A1:A3^2 wouldn’t do what we hope because its argument is an array, not a single number. We force it to operate over every element of the array by entering the formula with *Control-Shift-Enter*. This will return an array which is eventually summed.

\[
\]
Array formula example

Another example from Excel is

\{=\text{SUM}(	ext{LEN}(A1:A3))\}

where \text{LEN} returns the number of characters in the “text” argument.

\text{LEN(“hello”)} is 5.

If cell C1 = hello, then

\text{LEN(C1)} is 5.
Example detail

LEN(A1:A3) wouldn’t do what we hope because its argument is an array, not a single cell or text data. We would like to force LEN to operate over every element of the array by entering the formula with Control-Shift-Enter. The current version of Calc doesn't support the iteration of a function (LEN) over the elements of an array. This is what is required. \{=\text{SUM}(\text{LEN}(\text{A1:A3}))\}

\begin{array}{c|c|c|c|c}
1 & \text{hello} & \text{LEN} & 5 \\
2 & 2.3 & \text{SUM} & 3 \\
3 & \text{FALSE} & & 5 \\
\end{array}

Intermediate array
If you sum an array generated by a formula that contains elements that are error states, the summation will not work. Hence we screen them out with,

\{=SUM(IF(ISERROR(Scores),"",Scores))\}

This creates a new array containing the original values, minus the errors (which are replaced by NULL).

If rather than summing the values of the array elements, you want to count them the errors, so add 1 every time the value is an error, by \{=SUM(IF(ISERROR(Scores),1,0))\}

This formula creates a new array consisting of 1 and 0s.
A simplification of the previous formula would be
\{=\text{SUM}((\text{ISERROR}((\text{Scores})\times 1))}\}

which works because TRUE is represented by 1 and FALSE by 0, and hence the only two values that could be summed would be TRUE\times 1=1, or FALSE\times 1=0.

This could be considered a bit of a hack since you are taking advantage of the internal representation of TRUE and FALSE to do arithmetic!
Just a few more array formulas

You can use functions provided, but you can also use several functions together to extend the power of your formulas.

=SUMIF(Score, ">=50")

returns the sum of the scores greater than or equal to 50 in the array named “Score”. You can build this from other functions as with, 

{=SUM(IF(Score>=50,Score,0))}

SUMIF can't handle more than one condition, but the above formula can through nested IFs. But you can also use the previous bit of a hack, as with

{=SUM((Score>=50)*(Score<60)*Score)}

to give the sum of scores between 50 and 60.
Manipulating an array

\{=\text{AVERAGE}(\text{IF}(\text{score} \geq 50, \text{score}, ""))\}

- Is a simple, yet powerful array formula construct.
- Uses a named array (score).
- The IF function is iterated over the elements of score.
- It returns an array whose elements are either the score values if they were $\geq 50$, or null if they were $< 50$.
- The AVERAGE of this new array is taken.

It returns the average mark of the passing cohort.
This array will exist in memory only

\{=\text{AVERAGE}(\text{IF}(\text{score} \geq 50, \text{score}, ""))\}

This array is passed to the AVERAGE function

<table>
<thead>
<tr>
<th>Std Number</th>
<th>Family Name</th>
<th>Enrol</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>95000000</td>
<td>NAME-A</td>
<td></td>
<td>56</td>
</tr>
<tr>
<td>96000000</td>
<td>NAME-B</td>
<td></td>
<td>29</td>
</tr>
<tr>
<td>97000000</td>
<td>NAME-C</td>
<td></td>
<td>79</td>
</tr>
<tr>
<td>98000000</td>
<td>NAME-D</td>
<td></td>
<td>89</td>
</tr>
<tr>
<td>99000000</td>
<td>NAME-E</td>
<td></td>
<td>78</td>
</tr>
<tr>
<td>00000000</td>
<td>NAME-F</td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>01000000</td>
<td>NAME-G</td>
<td></td>
<td>66</td>
</tr>
<tr>
<td>02000000</td>
<td>NAME-H</td>
<td></td>
<td>83</td>
</tr>
<tr>
<td>03000000</td>
<td>NAME-I</td>
<td></td>
<td>47</td>
</tr>
<tr>
<td>04000000</td>
<td>NAME-J</td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>05000000</td>
<td>NAME-K</td>
<td></td>
<td>76</td>
</tr>
<tr>
<td>06000000</td>
<td>NAME-M</td>
<td></td>
<td>78</td>
</tr>
</tbody>
</table>
\{=\text{AVERAGE}(\text{IF}(\text{enrol}=5013,\text{score},""))}\}

- There are two named arrays (\textit{enrol} & \textit{score}).
- The IF function is \textit{iterated} over the elements of \textit{enrol}.
- Returns an array whose elements are either the \textit{score values} if the \textit{enrol values} are 5013, or \textit{null} if the \textit{enrol values} are not 5013.

The \textit{score value} returned is the one that matches the \textit{enrol value}.

- The AVERAGE of this \textit{new} array is taken.
- It returns the average mark of BSc/BE (5013) students.
This array is passed to the `AVERAGE` function

`=AVERAGE(IF(enrol=K11,score,""))`

..... and fill down to do all the degree programmes
Spreadsheet macros

Macros are repositories of instructions (ie. a program) stored in modules in the spreadsheet

Can have an impact on the entire worksheet, unlike functions which can only return a value to a cell location.

Macros can affect any cell value and can automatically change the format of cells, cell ranges or the entire workbook.

Novices are encouraged to stick to simple macros. Macro languages can be quite complex and Microsoft's VisualBasic can be difficult to master.
Recording a macro

A user can write a complete macro, and Excel provides the ability to automatically generate a macro by recording user behaviour.

As you use the spreadsheet, what you are doing is recorded as program commands.

Use the menu sequence Tools>Macros>Record Macro

A toolbar that allows you to pause and restart the recording is displayed.

The user performs the operations and instructions to be recorded.

When the user has completed the required task, the recording can be stopped and you are requested to name your program.
Diamond Use the menu sequence Tools-> Macro-> Record New Macro
Recording a macro

◊ Use the menu sequence Tools-> Macro-> Record New Macro

![Record Macro dialogue box](image-url)
Viewing a macro

With the menu selection **Tools>Macros>Macro** the user can choose to edit and view the recorded macro.

The contents of the recorded macro are the programming statements that will repeat the task the user has just completed by hand.

The user can improve the macro by changing the program statements (if they know what they are doing).
An example Visual Basic macro

Sub InsertData()
    Workbooks.OpenText Filename:="C:\Nick\MyFile.xls"
    Range("A1:D12").Select ' Select the data from the above opened file
    Selection.Copy ' Copy the selected data into the clipboard
    ActiveWorkbook.Close ' Close that file
    Range("A4").Select ' Back to original sheet, select the cell to begin pasting
    ActiveSheet.Paste ' Paste data from the clipboard
End Sub

In this example the user opened another spreadsheet, selected the cells A1:D12, copied them, closed down that workbook, and pasted the values into the original workbook from cell A4.

If this task was required repeatedly, one could just run this macro.
Programming....

◊ We will build on this idea of making repeatable scripts that can automate complex tasks.

◊ However, we will not use Visual Basic.

◊ We cover the fundamentals of programming using the Matlab environment.

◊ These fundamentals can be applied in many different environments including Visual Basic and Excel.