VBA Extras Part 2: Excel Programming & Data Visualization

- Excel VBA programming: Inserting charts into a spreadsheet & sorting data
- General programming concept, nesting: loops within loop
- Visualizing information in Excel: Pivot tables

Excel VBA Programming

Inserting Charts Into A Spreadsheet[1]

- **Step 1**: A range of cells needs to be selected via the Range object, examples:
  - Adjacent columns:
    ```
    Range("C1:D13").Select
    ```
  - Non-adjacent columns
    ```
    Range("C1:C13,E1:D13").Select
    ```
- **Step 2**: the chart[2] is added inside of container shape
  - ```
    ActiveSheet.Shapes.AddChart2(201, xlLineMarkers).Select
    ```

1 Information links on adding a chart
- https://docs.microsoft.com/en-us/office/vba/api/project.shapes.addchart
- https://docs.microsoft.com/en-us/office/vba/api/Excel.shapes.addchart2

2 Information specifying named constants for different chart types
- https://docs.microsoft.com/en-us/office/vba/api/Excel.XlChartType

VBA Example: Inserting Chart Clustered Line

- **Name of the spreadsheet that contains the VBA example**: `Excel5_insert_chart_clustered_line`
  - **Learning objective**: inserting this chart type with a hard coded (fixed) range.

```
Sub insertChartClusteredLine()
    Range("C1:C13,D1:D13").Select
    ActiveSheet.Shapes.AddChart2(201, xlLineMarkers).Select
End Sub
```
```

VBA Example: User Specified Values For Charts

- Specifying variable range of data (entered by a user) to chart
  - (Assumes the columns are side by side, modifications needed to chart non-continuous data).

```
startRange = InputBox(...)
startRange = InputBox(...)
range(startRange & ":" & endRange).Select
```

- Specifying chart title from a variable (entered by a user)
  - (Assumes that a chart has just been added)

```
chartTitle = InputBox(...)
ActiveChart.chartTitle.Select
ActiveChart.chartTitle.Text = chartTitle
```

VBA Example: Inserting Variable Chart Data

- **Name of the spreadsheet that contains the VBA example**: `Excel6_insert_chart_clustered_line_variable_range_and_title`
  - **Learning objective**: inserting this chart type with a user specified range and title.

```
Dim startRange As String
Dim endRange As String
Dim chartTitle As String

'Specifying the user selected range
startRange = InputBox("Start cell of data to chart: ")
endRange = InputBox("End cell of data to chart: ")
Range(startRange & ":" & endRange).Select
```

VBA Example: Inserting Variable Chart Data (2)

```
ActiveSheet.Shapes.AddChart2(201, xlLineMarkers).Select

'Setting the user specified title
chartTitle = InputBox("Title for the chart: ")
ActiveChart.chartTitle.Select
ActiveChart.chartTitle.Text = chartTitle
```

Counting Number Of Rows Of Data

	A	B	C	D	
1	**Min. gpa**	**Max. gpa**	**Letter**	**# Occurences**	**Start count**
2	0	Less than 0.7	F	0	
3	0.7	Less than & 1.15	D	0	
4	1.15	Less than 1.5	D+	1	
5	1.5	Less than 1.85	C-	2	
6	1.85	Less than 2.15	C	0	
7	2.15	Less than 2.5	C+	0	← Count + 1
8	2.5	Less than 2.85	B-	5	
9	2.85	Less than 3.15	B	20	
10	3.15	Less than 3.5	B+	7	
11	3.5	Less than 3.85	A-	9	
12	3.85	Less than 4.15	A	3	
13	4.15	Less than 4.3	A+	1	
14					

VBA Counting Rows Of Data

- **Name of the spreadsheet that contains the VBA example**: `Excel7_counting_rows`
  - **Learning objective**: determining the number rows of data (data = non-empty) in a spreadsheet.

```
Const LETTER_GRADE_COLUMN As Long = 3
Const START_ROW As Long = 1
Const EMPTY_ROW As String = ""
Dim rowData As String
Dim currentRow As Long
Dim count As Long
```

LETTER_GRADE_COLUMN

	A	B	C	D
1	**Min. gpa**	**Max. gpa**	**Letter**	**# Occurences**
2	0	Less than 0.7	F	0
3	0.7	Less than & 1.15	D	0
4	1.15	Less than 1.5	D+	1
5	1.5	Less than 1.85	C-	2
6	1.85	Less than 2.15	C	0
7	2.15	Less than 2.5	C+	0
8	2.5	Less than 2.85	B-	5
9	2.85	Less than 3.15	B	20
10	3.15	Less than 3.5	B+	7
11	3.5	Less than 3.85	A-	9
12	3.85	Less than 4.15	A	3
13	4.15	Less than 4.3	A+	1

EMPTY_ROW

VBA Counting Rows Of Data (2)

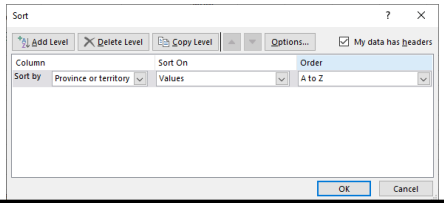
```
currentRow = START_ROW
count = 0
rowData = Cells(currentRow, LETTER_GRADE_COLUMN)
Do While (rowData <> EMPTY_ROW)
    count = count + 1
    currentRow = currentRow + 1
    rowData = Cells(currentRow, LETTER_GRADE_COLUMN)
 Loop
 MsgBox ("Num. rows=" & count)
```

Sorting Spreadsheets In Excel

- Select the range

	A	B	C	D	E	F
1	Province or territory	Population[1]	Total Covid infections[2]	Total infections/ 100,000	New infections[2]	New infections/100,000
2	Alberta	4,439,066	40,962	923	730	16.44
3	British Columbia	5,181,164	24,422	471	761	14.69
4	Manitoba	1,382,406	12,007		399	28.86
5	New Brunswick	783,462	388	50	9	1.15
6	Newfoundland & Labrador	521,870	307	59	2	0.38
7	Northwest Territories	45,245	15	33	0	0.00
8	Nova Scotia	983,417	1,154	117	4	0.41
9	Nunavut	39,892	70	175	0	0.00
10	Ontario	14,811,258	98,162	663	1417	9.57
11	Prince Edward Island	160,754	68	42	1	0.62
12	Quebec	8,611,402	127,233	1477	1179	13.69
13	Saskatchewan	1,180,677	5,553	470	131	11.10
14	Yukon	42,271	25	59	1	2.37

- Set sort criteria



VBA Sort Code Criteria

- You must first do this:
  - (According to MS-docs) Clears all the SortFields objects:
    - `ActiveWorkbook.Worksheets(1).Sort.SortFields.Clear`
- Specify the criteria used in the sort 'key':
  - `ActiveWorkbook.Worksheets(1).Sort.SortFields.Add Key:= Range("A1")`
- Specify the sorting order (ascending "A-Z" or descending "Z-A")
  - `ActiveWorkbook.Worksheets(1).Sort.SortFields.Add Order:=xlAscending` **`'x1Descending is the other option`**
- Specify the range of cells to be sorted
  - `ActiveWorkbook.Worksheets(1).Sort.SetRange Range("A1:F14")`
- Specify if there is a header row
  - `ActiveWorkbook.Worksheets(1).Sort.Header = xlYes`
  - **`'x1No=range has no header, x1yes=range has header`**

For more information: https://docs.microsoft.com/en-us/office/vba/api/excel.sort

VBA Example: Simple Sort

- **Name of the spreadsheet that contains the VBA example**: `Excel8_simple_sort`
  - **Learning objective**: sorting with a predetermined fixed range in the currently active worksheet.

```
ActiveWorkbook.Worksheets(1).Sort.SortFields.Clear
    ActiveWorkbook.Worksheets(1).Sort.SortFields.Add Key:= _
        Range("A1"), Order:=xlAscending
    With ActiveWorkbook.Worksheets(1).Sort
        .SetRange Range("A1:F14")
        .Header = xlYes 'Options: x1No, x1yes
        .Apply
    End With
```

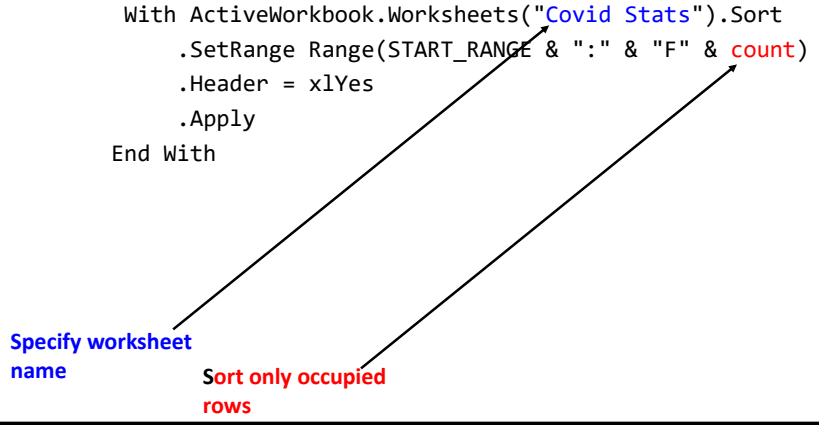
VBA Example: More Advanced Sort

- **Name of the spreadsheet that contains the VBA example**: `Excel9_advanced_sort`
  - **Learning objective**: sorting only rows that contain data, sorting the worksheet with the specified name.

```
'Count number of rows (i.e. contain data)
currentRow = START_ROW
count = 0
rowData = Cells(currentRow, 1)
Do While (rowData <> EMPTY_ROW)
    count = count + 1
    currentRow = currentRow + 1
    rowData = Cells(currentRow, 1)
Loop
```

VBA Example: More Advanced Sort

```
'Sort only occupied cells worksheet called "Covid Stats"
ActiveWorkbook.Worksheets(1).Sort.SortFields.Clear
ActiveWorkbook.Worksheets(1).Sort.SortFields.Add Key:= _
  Range(SORT_CRITERIA), Order:=xlAscending
    With ActiveWorkbook.Worksheets("Covid Stats").Sort
        .SetRange Range(START_RANGE & ":" & "F" & count)
        .Header = xlYes
        .Apply
   End With
```



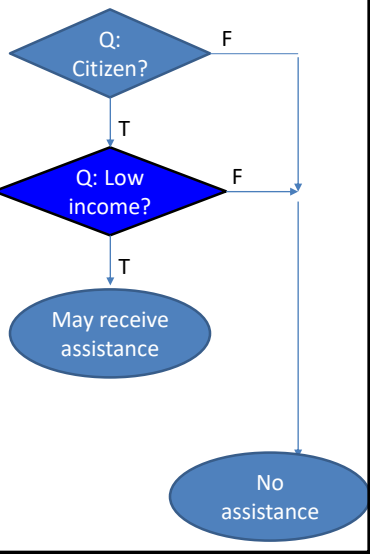
General Programming Concept: Nested Loops

Recognizing When **Nesting** Is Needed

- **Review**: A second question is asked only if a first question answers true:
  - Example: If it's true the applicant is a Canadian citizen, then ask for the person's income (checking if eligible for social assistance).
  - Type of nesting: an IF-branch nested inside of another IF-branch

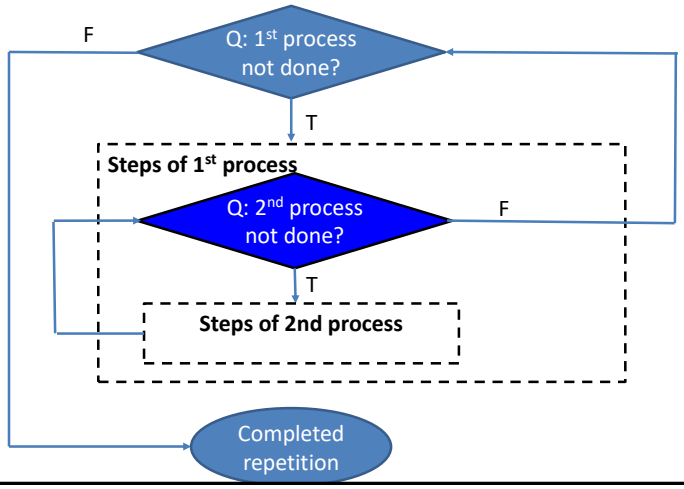
```
If (Boolean) then
    If (Boolean) then
        ...
    End If
End if
```

Nested branch/IF



New: Recognizing When **Nested** Repetition Is Needed

- For each step that a process repeats, repeat another process from start to end.



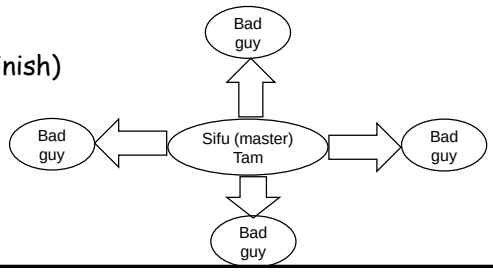
Example 1: Washing Dishes

```
While (there are dishes left unwashed)
    Get a dirty dish
   Apply soap to dish
    while (dish is still dirty)
       Rub dish with wet cleaning tool
       If (more soap needed)
           Apply soap to dish
```

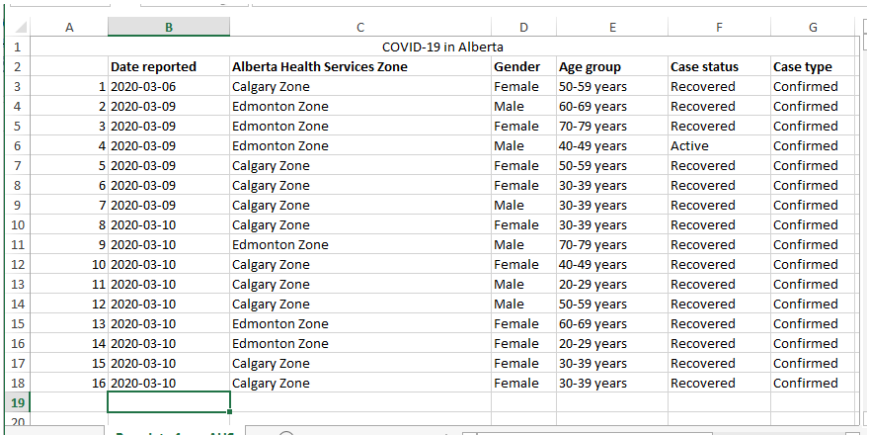
Example 2: Practicing A Martial Arts Set

```
Assume guard position
While (there is still a compass point with opponent)
    Turn left to face opponent
     while (opponent is still standing)
        Throw right reverse punch
         Left rising block
         Throw right reverse punch
     Assume guard position

Assume the curtesy (bow and finish)
```



Example 3: Counting Covid Alberta Cases



	A	B	C	D	E	F	G
1		COVID-19 in Alberta					
2		Date reported	Alberta Health Services Zone	Gender	Age group	Case status	Case type
3	1	2020-03-06	Calgary Zone	Female	50-59 years	Recovered	Confirmed
4	2	2020-03-09	Edmonton Zone	Male	60-69 years	Recovered	Confirmed
5	3	2020-03-09	Edmonton Zone	Female	70-79 years	Recovered	Confirmed
6	4	2020-03-09	Edmonton Zone	Male	40-49 years	Active	Confirmed
7	5	2020-03-09	Calgary Zone	Female	50-59 years	Recovered	Confirmed
8	6	2020-03-09	Calgary Zone	Female	30-39 years	Recovered	Confirmed
9	7	2020-03-09	Calgary Zone	Male	30-39 years	Recovered	Confirmed
10	8	2020-03-10	Calgary Zone	Female	30-39 years	Recovered	Confirmed
11	9	2020-03-10	Edmonton Zone	Male	70-79 years	Recovered	Confirmed
12	10	2020-03-10	Calgary Zone	Female	40-49 years	Recovered	Confirmed
13	11	2020-03-10	Calgary Zone	Male	20-29 years	Recovered	Confirmed
14	12	2020-03-10	Calgary Zone	Male	50-59 years	Recovered	Confirmed
15	13	2020-03-10	Edmonton Zone	Female	60-69 years	Recovered	Confirmed
16	14	2020-03-10	Edmonton Zone	Female	20-29 years	Recovered	Confirmed
17	15	2020-03-10	Calgary Zone	Female	30-39 years	Recovered	Confirmed
18	16	2020-03-10	Calgary Zone	Female	30-39 years	Recovered	Confirmed
19							
20							

Review Example: Nested Branch Inside Loop

- **Name of the spreadsheet that contains the VBA example**: Excel10_counting_occurences
  - **Learning objective**: review of how to write a program that **checks a condition (IF-branch)** each time that a **process repeats** (or runs for the first time) – which is a WHILE-loop.

```
currentRow = START_ROW
currentStatus = Cells(currentRow, STATUS_COLUMN)
Do While (currentStatus <> EMPTY_ROW)
    If (currentStatus = STATUS_OF_INTEREST) Then
        count = count + 1
    End If
    currentRow = currentRow + 1
    currentStatus = Cells(currentRow,
      STATUS_COLUMN)
Loop
Cells(OUTPUT_ROW, OUTPUT_COLUMN) = count
```

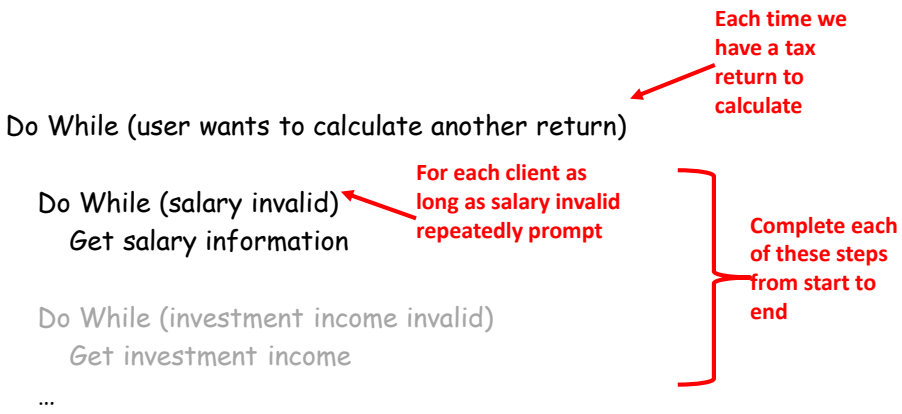
```
Const START_ROW = 3
Const STATUS_COLUMN = 6
Const EMPTY_ROW = ""
Const STATUS_OF_INTEREST = "Recovered"
Dim count As Long
Dim currentRow As Long
Dim currentStatus As String
Const OUTPUT_ROW As Long = 3
Const OUTPUT_COLUMN As Long = 10
```

Example Nested Loop

- While the user indicates that he/she wants to calculate another tax return (first, outer loop) prompt the user for income, while the income is invalid repeatedly prompt for income (second, nested inner loop).

```
Do while(another tax return)
     Do while(income is negative)
         ...
     Loop
 Loop
```

Nested Loop: Example Process In Pseudo Code



Loop Nested Inside A Loop

- **Name of the spreadsheet that contains the VBA example** : `Excel11_nested_loops_taxes`
  - **Learning objective**:
    - Summary: Each time a process repeats (**calculate a new tax return**) prompt for income so long **as income is less than zero**.
  - Program details:
    - **Outer loop** (first repeated process):
      - As long the user indicates there is another tax return to
      - calculate the program will go through all the steps needed.
      - to calculate taxes owed.
    - **Inner loop** (second process repeated each time the outer loop runs):
      - As long as the user enters a negative income the program will keep prompting for an income.
      - The prompt will involve getting the user to enter a value and error checking that value.

First Nested Loop Program

```
Const MIN_INCOME As Long = 0
Const TAX_RATE As Double = 0.17
Dim runAgain As Long
Dim income As Double
Dim taxOwed As Double

runAgain = vbYes
'vbYes = 6, vbNo = 7
```

First Nested Loop Program (2): Loop Inside An Outer Loop

```
Do While (runAgain = vbYes)
    income = InputBox("Income $")
    Do While (income < MIN_INCOME)
        If (income < MIN_INCOME) Then
            MsgBox ("Income cannot be less than $" & MIN_INCOME)
        End If
        income = InputBox("Income $")
     Loop
     taxOwed = income * TAX_RATE
     MsgBox ("Taxes owed $" & taxOwed)
     runAgain = MsgBox("Calculate another tax return?", vbYesNo)
Loop
```

Second Nesting Problem

- Counting the number of students in each tutorial for each lecture.

	A	B	C
1	**Lecture**	**Tutorial**	**Student information**
2	L01	T01	Student1
3	L01	T01	Student2
4	L01	T01	Student3
5	L01	T02	Student4
6	L01	T02	Student5
7	L02	T01	Student1
8	L02	T01	Student2
9	L02	T01	Student3
10	L02	T01	Student4
11	L02	T02	Student5
12	L02	T02	Student6
13	L02	T03	Student7
14	L02	T03	Student8
15	L02	T03	Student9
16			
17			

Counting Students In Each Tutorial

- **Name of the spreadsheet that contains the VBA example** : `Excel12_nested_loops_counting_students`
- **Learning objective**:
  - Summary: applying nested loops in the processing of data in a spreadsheet.
  - **Repeated process (outer loop)**: while the end of the spreadsheet has not yet been reached process the row in the spreadsheet and move onto the next row.
    - **Second repeated process (inner loop)**: each time that a new row in the spreadsheet has been reached:
      - Check if the end of the tutorial has been reached.
      - If not increase the student count for the tutorial
      - Move onto the next row

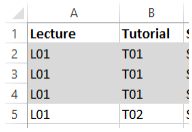
Counting Students In Each Tutorial

```
tutorialCount = 0
currentRow = 2                          2
cellContents = Cells(currentRow, TUTORIALS)
Do While (cellContents <> EMPTY_ROW)  ""
    startTutorial = cellContents
    currentTutorial = startTutorial
    tutorialCount = 0
    Do While (currentTutorial = startTutorial)
        tutorialCount = tutorialCount + 1
        currentRow = currentRow + 1              2
        currentTutorial = Cells(currentRow, TUTORIALS)
    Loop
    MsgBox ("Tut: " & startTutorial & ", count=" & _
             tutorialCount)
    cellContents = currentTutorial
Loop
```

Counting Students: Alternative Solution

- Students sometimes find nested loops complex.
  - Sometimes it's unavoidable and the program must have one loop inside the other.
- Here is an alternative to using the nested loops of the previous example.
  - The solution is implemented with a **branch** inside of a **loop**
- **Name of the example**:
  `Excel12_nested_loops_counting_students_NO_NESTED_LOOPS`
  - High level view of the solution:

```
Do-while (End of data – an empty row not reached)
    If (current row is still part of the current tutorial) Then
       increase the count for # students in the current tutorial
       move onto the next row
    Else 'Current row is part of the next tutorial
       reset count for # of students in tutorial (starting a new tutorial)
End-Loop
```



	A	B
1	Lecture	Tutorial
2	L01	T01
3	L01	T01
4	L01	T01
5	L01	T02

Counting Students: Alternative Solution (2)

```
Const TUTORIALS As Long = 2

tutorialCount = 0
currentRow = 2
cellContents = Cells(currentRow,
  TUTORIALS)
startTutorial = cellContents
currentTutorial = startTutorial
tutorialCount = 0
```

T01

	A	B	C
1	Lecture	Tutorial	Student information
2	L01	T01	Student1
3	L01	T01	Student2
4	L01	T01	Student3
5	L01	T02	Student4
6	L01	T02	Student5
7	L02	T01	Student1
8	L02	T01	Student2
9	L02	T01	Student3
10	L02	T01	Student4
11	L02	T02	Student5
12	L02	T02	Student6
13	L02	T03	Student7
14	L02	T03	Student8
15	L02	T03	Student9

Counting Students: Alternative Solution (3)

```
Const EMPTY_ROW As String = ""
Const LECTURES As Long = 1
Const TUTORIALS As Long = 2

Do While (cellContents <> EMPTY_ROW)
    If (currentTutorial = startTutorial) Then
        lecture = Cells(currentRow, LECTURES)
        tutorialCount = tutorialCount + 1
        currentRow = currentRow + 1
        cellContents = Cells(currentRow, TUTORIALS)
        currentTutorial = cellContents
    Else
        MsgBox (lecture & " " & startTutorial & "
          & tutorialCount)
        tutorialCount = 0
        startTutorial = currentTutorial
    End If
Loop
MsgBox (lecture & " " & startTutorial & ", count=
  tutorialCount
```

	A	B
1	Lecture	Tutorial
2	L01	T01
3	L01	T01
4	L01	T01
5	L01	T02
6	L01	T02
7	L02	T01
8	L02	T01
9	L02	T01
10	L02	T01
11	L02	T02
12	L02	T02
13	L02	T03
14	L02	T03

Visualizing Information In Excel

Pivot Tables

- A useful tool for visualizing information:
  - Summarize data
  - Filter criteria
  - Create reports
  - (And more!)

- References:
  - https://support.microsoft.com/en-us/office/create-a-pivottable-to-analyze-worksheet-data-a9a84538-bfe9-40a9-a8e9-f99134456576
  - https://eitsc.com/blog/the-benefits-of-using-pivot-tables-to-manage-your-data/

Basic Pivot Table Example

- **Example spreadsheet used**: `Excel13_Visualization_example`

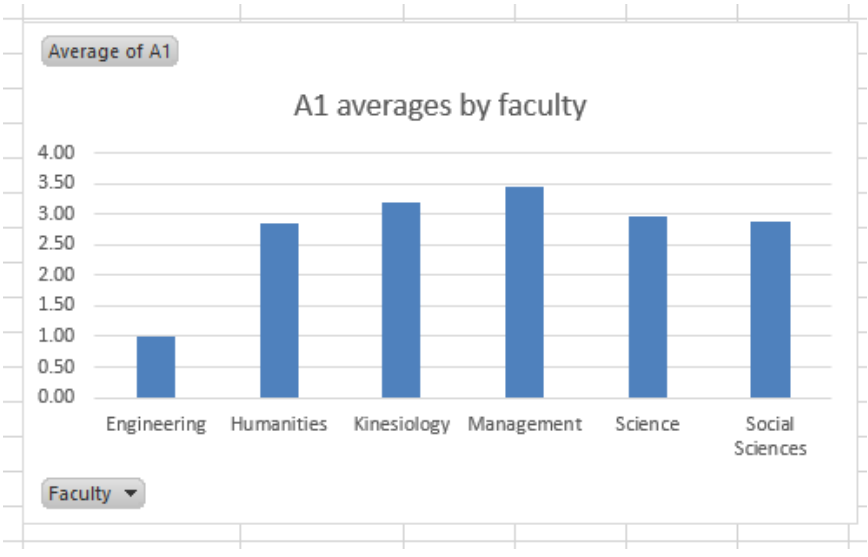
	A	B	C	D	E	F	G
1	CPSC 203: winter 2063						
2	Student ID	Faculty	A1	A2	A3	A4	Book Ex: W
3	111	Science	4	4.3	4.3	4.3	4
4	112	Social Sciences	3.3	3.3	3	3	4
5	113	Social Sciences	3	3.3	3.7	4	3
6	114	Management	4	4	4.3	3	4
7	115	Management	4	4	4	4	4
8	116	Management	3.3	2.7	3	4	4
9	117	Humanities	2.3	3.3	3	4	4
10	118	Social Sciences	3.3	2.7	3.3	4	4
11	119	Management	4	1.7	3.3	4	3
12	120	Management	4	4	3.7	4	3
13	121	Kinesiology	4	4	4	3	3
14	122	Management	4	4	4	4	3
15	123	Management	4	4	4	4	4
16	124	Humanities	3	2.7	3	3	4
17	125	Science	2.3	3.3	3	4	3
18	126	Social Sciences	4	3.7	0	4	4
19	127	Social Sciences	3.7	1.7	3	4	3
20	128	Social Sciences	4	4	3.7	3	3

Pivot Table Example: Summarizing Data

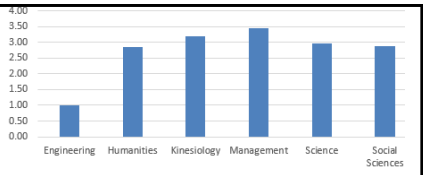
	A	B	C	D	E	F	G
1	CPSC 203: winter 2063						
2	Student ID	Faculty	A1	A2	A3	A4	Book Ex: W
3	111	Science	4	4.3	4.3	4.3	4
4	112	Social Sciences	3.3	3.3	3	3	4
5	113	Social Sciences	3	3.3	3.7	4	3
6	114	Management	4	4	4.3	3	4
7	115	Management	4	4	4	4	4
8	116	Management	3.3	2.7	3	4	4
9	117	Humanities	2.3	3.3	3	4	4
10	118	Social Sciences	3.3	2.7	3.3	4	4
11	119	Management	4	1.7	3.3	4	3
12	120	Management	4	4	3.7	4	3
13	121	Kinesiology	4	4	4	3	3
14	122	Management	4	4	4	4	3
15	123	Management	4	4	4	4	4
16	124	Humanities	3	2.7	3	3	4
17	125	Science	2.3	3.3	3	4	3
18	126	Social Sciences	4	3.7	0	4	4
19	127	Social Sciences	3.7	1.7	3	4	3
20	128	Social Sciences	4	4	3.7	3	3

Faculties	Average of A1
Engineering	1.00
Humanities	2.86
Kinesiology	3.18
Management	3.44
Science	2.98
Social Sciences	2.87
Grand Total	**3.06**

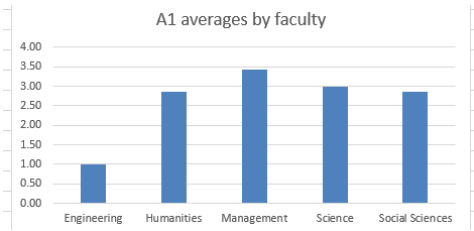
Pivot Table Example: Charting Results



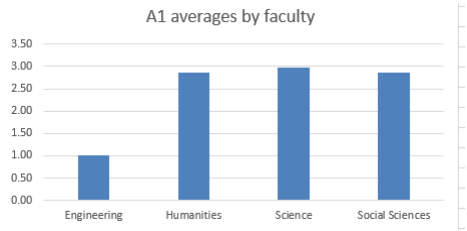
Pivot Table Example: Filtering



- Faculty of kinesiology removed.



- Only show averages for faculties that are 3.0 GPA or less.



After This Section You Should Now Know

- How to write VBA instructions to insert a chart into a spreadsheet with hard coded or variable properties (title and range) for the chart
- How to write the instructions to count the number of non-empty rows in a spreadsheet
- How to write the VBA instructions sort the rows of spreadsheet
- Nested loops
  - What is a nested loop
  - How to trace a nested loop
  - Scenarios when nested loops can be applied
- Some of the benefits of using a pivot table, how to insert one into a spreadsheet and how to chart the data

Images

- “Unless otherwise indicated, all images were produced by James Tam
- Sound effects produced by James Tam

slide 41