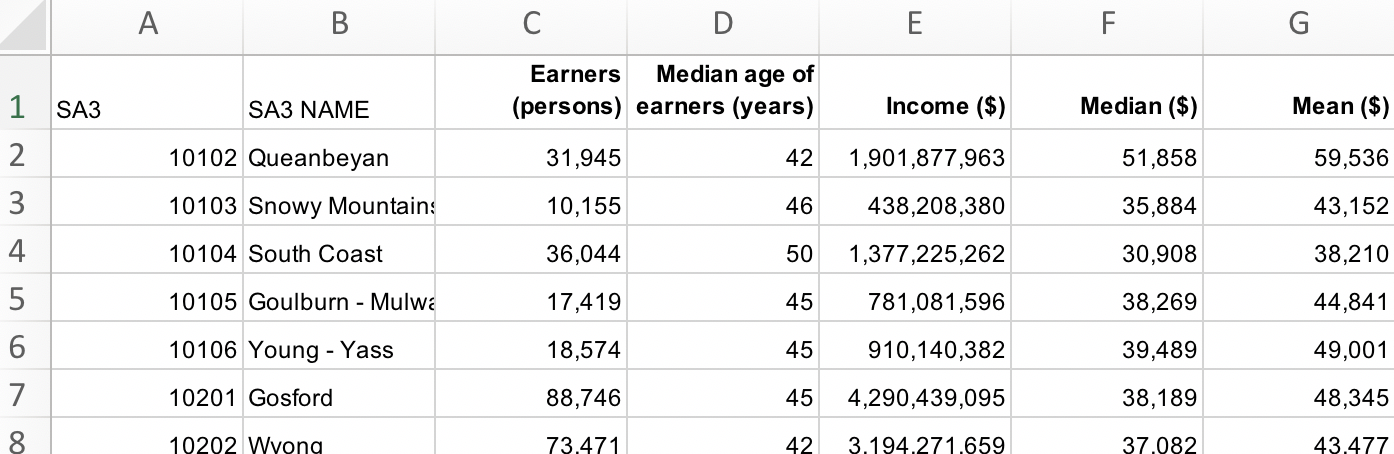
A brief data course

15.01.2019

# PART I: Super fun with Excel

**Get your data**

1. Create a “session” folder on your system that will contain all the bits and pieces used today.
2. Open a Word document and title it “Data Notes: Does living far away from the capital city reduce per worker income?”. We will use this throughout to document any decisions we make about the data.
3. Go to the ABS website and find:
   * + 1. [- Estimates of Personal Income for Small Areas, 2011-2015](http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/6524.0.55.0022011-2015?OpenDocument)
4. Download at the estimates of total income by Statistical Area 3 in **2010-11**.
   * The sheet name is “Total Income - Income Distribution, SA2, SA3, SA4, GCCSA, S/T, Australia, LGA, 2010-11”. *Read the titles carefully!*
   * Open to Table 3.
5. Copy the sheet to another workbook (ie in a new Excel document). We start a new sheet to maintain an original copy of the data before editing. This is good practice [**why?**].
   * Save your new workbook as something descriptive (eg income\_sa3\_2011).
   * Remove all columns past “Mean”. Give appropriate names to the columns.
   * Now, remove all unnecessary rows at the top. Set up your sheet so the first row contains names and the data runs from the second row. So, combine any information about the variable into a single row.
   * Your sheet should look something like this:



1. In your Data Notes document, note down relevant information so far. These notes can be scrappy, but must be interpretable by another reader if necessary.
   * Source of data (ABS) and document name (Estimates of Personal Income for Small Areas, 2011-15, Table 3).
   * What variables you have kept and plan to use.

**Exploring and manipulating data**

1. Back to the main sheet. We first guess that the column Mean is just the total income of an area divided by how many workers there are. Check that this is true:
   * Calculate your own average Mean\_test in a new column:



* + Format this number so it shows zero decimal places and a comma to separate thousands.
    - Those numbers look the same! Good start with n = 1.
  + Copy this formula down the entire column by double-clicking on the little green square in the bottom-right corner.
    - **[What is another way to do this?]**
  + Manually check if they are all the same. [kidding! That would take ages! Let’s do it another way].
  + Write a formula to check it for you. First, round the Mean\_test column to zero decimal places using the function =ROUND().



Digression: **IF statements**

IF statements make up a large part of ‘coding’, regardless of language. IF something is TRUE it will do something, and (sometimes) IF it is FALSE it will do another. They’re handy! In Excel this is coded using the if() formula:

**=if(**[something is true], [do this],

[if not true do something else]**)**

**eg:**

**=if(**35=10, “this is correct”,

“this is a lie”**)**

would return:

this is a lie

but:

**=if(**“apple”=”apple”, “this is correct”,

“this is a lie”**)**

would return:

this is correct

Like all formulas in Excel (and elsewhere), IF functions can be “nested”, ie there is a formula within a formula. **eg:** what would this function return?

**=if(**“apple”=”oranges”, “don’t compare”,

if**(**“chalk”=”cheese”, “shouldn’t compare”,

if**(**10/5=1+1,”numbers to the rescue”,

“this is an endulgent example” **)))**

* + Second, in a new column, write a formula that returns nothing (empty quotation marks) if there is a match, and yells “NO MATCH” if there is not a match. We can do this with the IF function:



* + What’s going on with the empty “” in this formula? We use quotation marks to signal a string (aka text). Where there is no match, the function will return NO MATCH as text. When there *is* a match, the function will return an empty string because there is nothing between the quotation marks. This makes it easy to identify the cells that read “NO MATCH”.
    - *Is there a more elegant way to do this?*
    - ***[why did we have to round the* mean\_test *variable?]***

1. Scrolling down, we can see that all observations (rows) with proper values matched. That’s good news. The bad news is that our IF function breaks down when it meets an error cell of #VALUE! (*Why?).* To address this, we can use a handy little function called IFERROR:

**=iferror**([value you want],   
 [alternative if your orignal value is an error])

eg1: iferror(sum(“apples”,”oranges”),”Error”) = “Error”

eg2: iferror(2+3,”Error”) = 5

1. We need to decide what to do with these “na” rows. There are some options: leave as-is or delete.
   * We decide to delete them.
   * First, we record in our Data Notes document **what** we are doing and **why** we are doing it.
   * This is mainly for yourself down the track. In a few months’ time—when this project is long behind you—you’ll return to your work only to find “Blue Mountains – South” missing from your data. It’s better to have detailed notes for all decisions made.

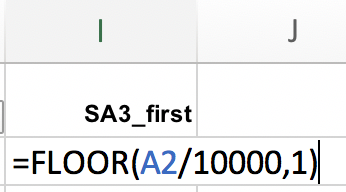
To delete the observations, we can try filtering the data: select the entire dataset and click “Sort & Filter”.



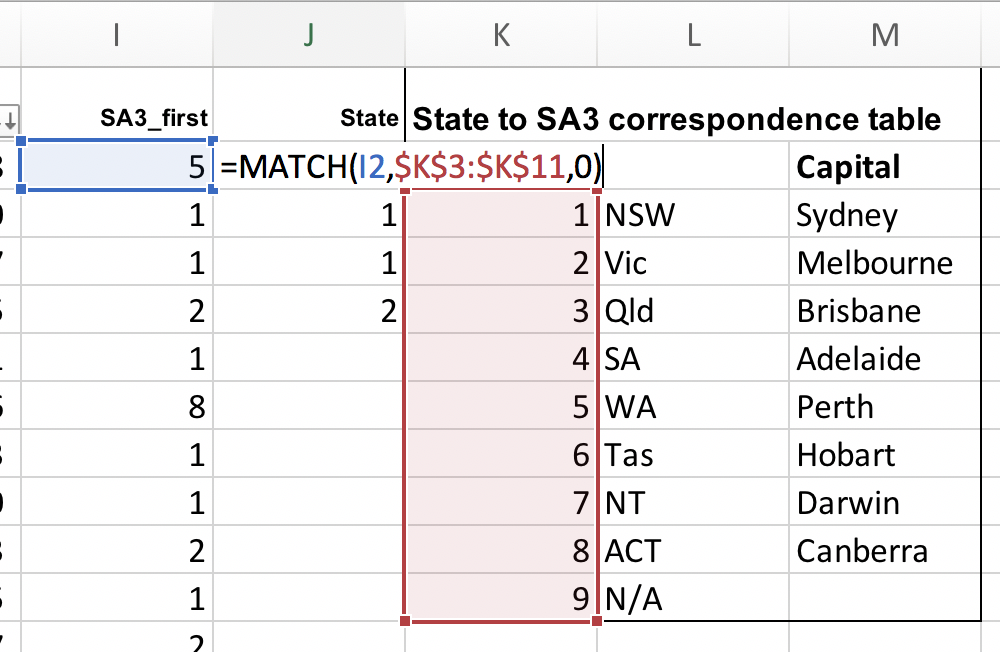
Sort Mean\_test as descending. This will bring the na values to the top. Select the entire row by clicking on the row number, shift-click to the 22nd row, then right-click ‘delete’.

* + The reasons I tediously write that out:
    - For instructional purposes; and
    - to show that ‘point-and-click’ steps are actually quite numerous and detailed! And, after you do this, you’ll have no record that you did it! The rows will just be gone! Madness!

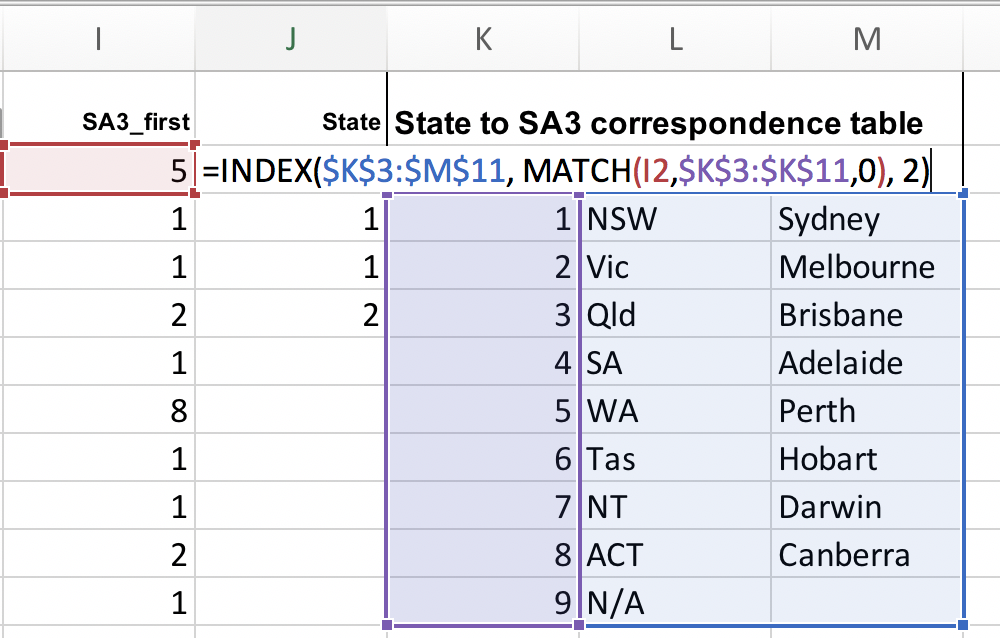
1. Now we want to get an indicator of the state for each region. Use the correspondence table (provided in 3\_sa3\_state\_corresp.xlsx). Copy the correspondence table into your main sheet from cell K2. In cell K1, write “State to SA3 correspondence table”. Give the table a border for clarity.
   * In columns I and J we’re going to build a formula that reads the correspondence table for each observation.
   * The first digit of an SA3 code represents the state. We start by writing a function that returns the first digit of the SA3 code, and then copy that down the column:



* + We use an [INDEX/MATCH](https://www.deskbright.com/excel/using-index-match/) function to “look up” the correspondence table.
    - This is a nested function (a function within a function). The MATCH part takes three arguments:   
      MATCH([take this thing], [look for it here], 0)   
      and returns the row or column number when it matches.
    - So in the first row it will take whatever is in cell I2 [which is 5], look for it in the range K2:K11, and return the row number.

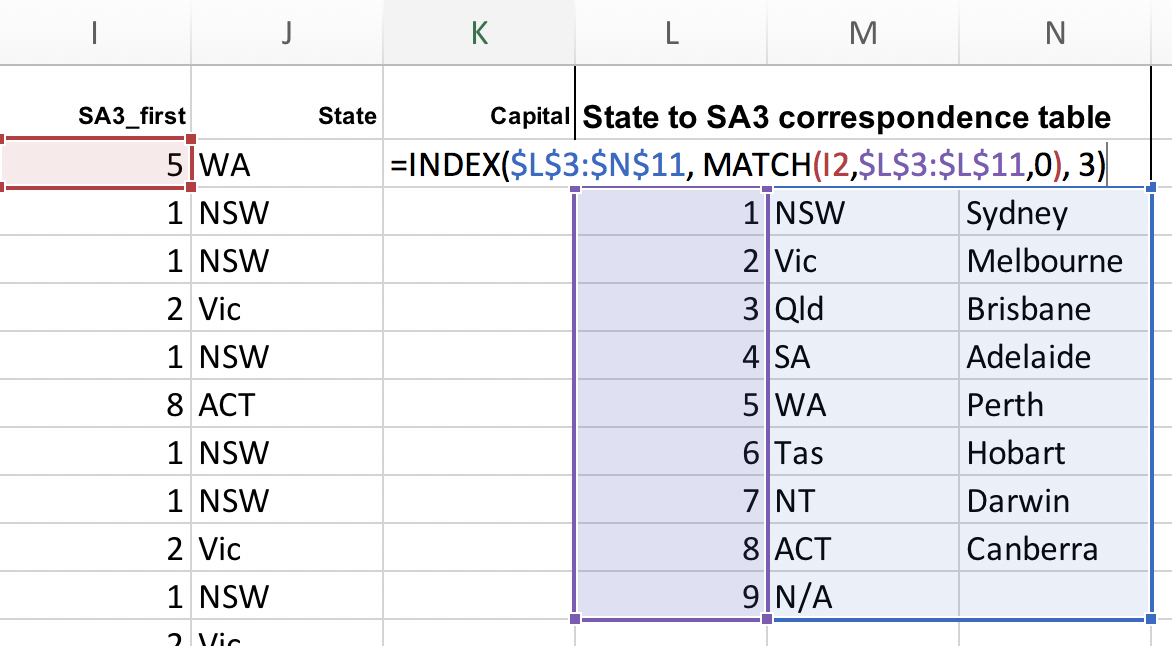


* + The INDEX function simply asks:   
    INDEX([range], [row number], [column number]).   
    and it will return the value of the cell that is the [row number] and [cell number] of the [range]. A simple example: INDEX(A1:B2, 2, 2) would return B2.
  + So:
    - let’s make the correspondence table our range: K3:M11.
    - then, the row number will be our MATCH function:   
      MATCH(I2, K3:K11).
    - and the column number is what we want to return, the state: 2.

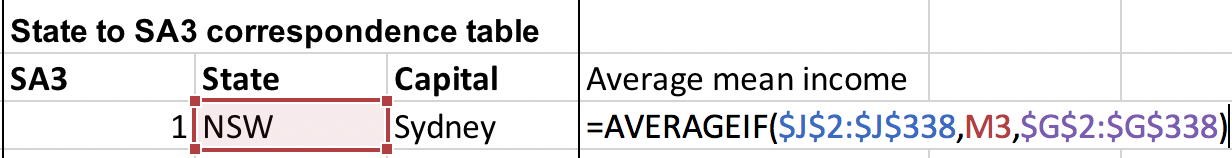


* + Wonderful! It returned WA. Now copy that down to the whole column.

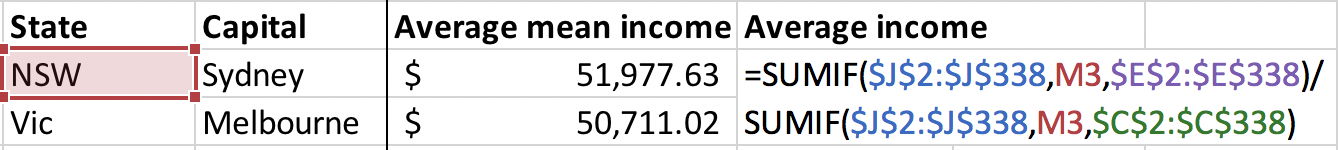
1. Actually, let’s also use the correspondence table and the INDEX/MATCH function to return the name of the capital in each state for each SA3. First: create a new column (right click on column L, select ‘insert’). Name the new column Capital. Write a new INDEX/MATCH function to return the capital name.



1. Great. Now, *finally*, adding new columns to the correspondence table, we’ll generate:
   1. average mean income per SA3 in each state; and
   2. average income per person in each state (note the difference).
   * 1) Average mean income per SA3 in each state



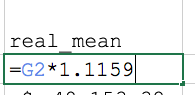
* + 2) Average income in each state



* + *What’s the difference?*

**Dealing with inflation/CPI**

1. Our figures are from 2011, and are in $2011. If we want to – and we usually do – report figures in $2017, we will need to “inflate” them. This is done using ABS CPI tables. The steps to do this follow. **But, for now, we’ll leave the figures as they are** (this is just for your reference later)
   * Find the ABS CPI table: <http://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/6401.0Jun%202017?OpenDocument>
     + Download the Excel file: “TABLES 1 and 2. CPI: All Groups, Index Numbers and Percentage Changes”
     + Go to sheet “Data1”
     + Go to Column J
   * We want to inflate 2011 to 2017, so we take the midyear CPI index of 2011 (equal to 99.2) over the midyear CPI index of 2017 (110.7), giving 110.7/99.2 == 1.1159
   * Back to our main sheet, in a new column called “mean”, generate real mean incomes for each SA3.



1. Copy the columns SA3, SA3 NAME, state, capital and real\_mean to a new workbook.
   * *What happened when we pasted State?*
   * Select Paste -> Paste Values to fix it.
2. Rename SA3 NAME to sa3\_name and real\_mean to mean.
3. Save as a .csv file.

# PART II: Creating Charts

1. Have a look at the **Charts for reports.potx** Powerpoint file to familiarise yourself with the Grattan style (in **additional\_resources/**)
2. Use the **6\_making\_powerpoint\_charts.ppt** and the data you have generated to replicate the chart and create a new one.
   * Note the formatting:
     1. 18 point Arial font
     2. Black x-axis line.
     3. No y-axis line.
     4. Grey major horizontal gridlines
     5. Comma separating thousands
     6. No off-plot legend. Guides are on the chart next to an example.

# PART III: Short introduction to R and R Studio

1. Open the R project file called 7\_intern\_data\_class1811.Rproj
2. From R Studio, CTRL+O and open 8\_partii\_r\_script.R
3. Google “placement package in r” and find the [CRAN entry](https://cran.r-project.org/web/packages/placement/index.html).
   * CRAN is the “Comprehensive R Archive Network”, in which people put (very) detailed descriptions of the packages (programs) they have created. There are *lots* of packages written for R. If you’re ever thinking “I wish there was a way to do this faster/better/at all”, there is probably someone out there who has written a package for it. People are wonderful.
4. Follow R script provided.

**Further fun!**

* See [here](http://www.abs.gov.au/websitedbs/D3310114.nsf/4a256353001af3ed4b2562bb00121564/6b6e07234c98365aca25792d0010d730/$FILE/Statistical%20Area%20Level%203%20-%20Fact%20Sheet%20.pdf) for more information about ABS’s statistical area 3.
  + See a fun (maybe a strong word) map of ABS structures [here](http://stat.abs.gov.au/itt/r.jsp?ABSMaps).
* [Data Camp](https://www.datacamp.com/courses/dplyr-data-manipulation-r-tutorial) has some free courses on manipulating data with R (using Tidyverse packages).
* Google tip: *“how to [X] with tidyverse”*. Eg:
  + *how to add a new variable with tidyverse*
  + *how to import an excel sheet with tidyverse*
* A great resource data visualisation is [here](http://socviz.co/). Note that it’s a full book, but it’s a pretty excellent way for beginners to quickly develop visualisation skills using ggplot in R.
* [Flowing Data](https://flowingdata.com/) is another resource for data visualisation in R.