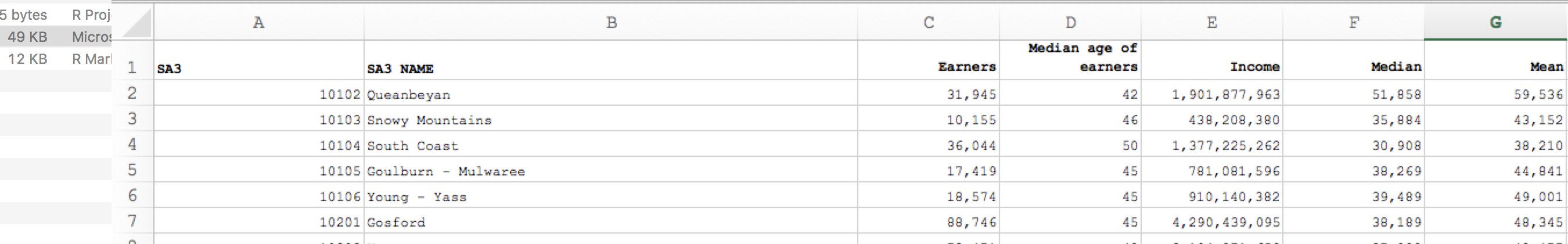
A brief data course

11.01.2018

# 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 another Excel document). We start a new sheet to maintain an original copy of the data before editing. This is good practice. Save your new workbook as something descriptive (eg income\_sa3\_2011). Now, remove all unnecessary rows at the top. Remove all columns past “Mean”. Give appropriate names to the columns. 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.

**Check your data**

1. PivotTables are a tool in Excel that allows you [quickly analyse data to help you make better business decisions](https://support.office.com/en-us/article/Create-a-PivotTable-to-analyze-worksheet-data-a9a84538-bfe9-40a9-a8e9-f99134456576) (?). They are handy for exploring small-moderate sized datasets. They are not as flexible as something you would build yourself, but lots of people still use them because they are reasonably self-explanatory.[[1]](#footnote-1) To make a PivotTable, highlight all of your data (CTRL+A) and select “Recommended PivotTables” from the Insert tab.
   * *What’s going on with the default?*
   * Use the filters, rows and columns functions to find the following:
     + How many “earners” are there in the dataset?
     + What is the average median age, and average income?
     + What is the lowest/highest median age, and lowest/highest income?
     + What is the sum of all income for all SA3s?
   * *Does all of this look reasonable?*
   * *What variables could be created that would enable this PivotTable to be more…exciting?*

**Manipulating data**

1. Back to the main sheet. We 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 by:
   * Calculating your own average Mean\_test in a new column



* + Copy this formula down the entire column.
    - *What’s the quickest way to do this?*
  + Manually check if they are all the same.
  + Ah, that’s boring: write a formula to check it for you. First, round the Mean\_test column to zero decimal places (to match the Mean column).



Digression: **IF functions**

Make up a large part of ‘coding’, regardless of language. It looks for a TRUE or FALSE statement (eg 10+10 = 20 is TRUE) and returns a value based on the outcome. 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 damn lie”**)**

would return:

this is a damn lie

but:

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

“this is a damn 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 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 “” 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 cell 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?*
    - *Hey – 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([function you want], [value you want if your function returns an error])

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

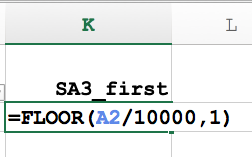
eg2: iferror(2+3,”Error”) = 5

1. We need to decide what to do with the “na” rows. There are two options: leave as is or delete. We decide to delete them. First, we document 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, and you return to your work only to find “Blue Mountains – South” missing from your data. It’s better to have detailed notes for all major decisions made.

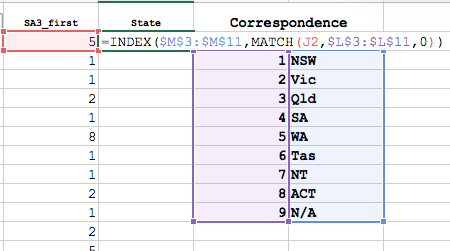
To delete the observations, we can try filtering the data: select the entire dataset and click “Sort & Filter”.



1. Now we want to get an indicator of the state for each region. Use the correspondence table (provided). Copy the correspondence table into your main sheet into column M. In columns K and L we’re going to build a formula that reads the correspondence table for each observation.
   * We start by writing a function that returns the first digit of the SA3 code:

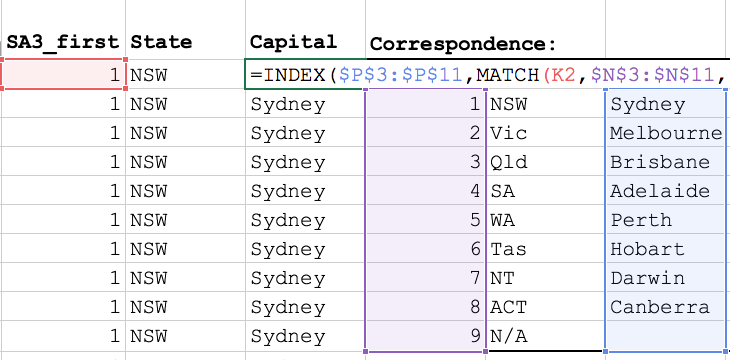


* + We use an [INDEX/MATCH](https://www.deskbright.com/excel/using-index-match/) function to “look up” the correspondence table:

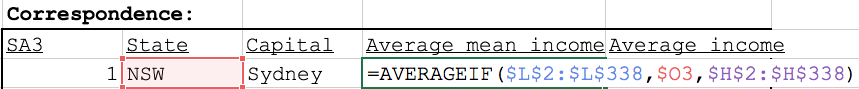


* + If you’re not familiar with INDEX/MATCH functions, try playing with just an INDEX function to see what it does. Then play with just a MATCH function to see how they combine.

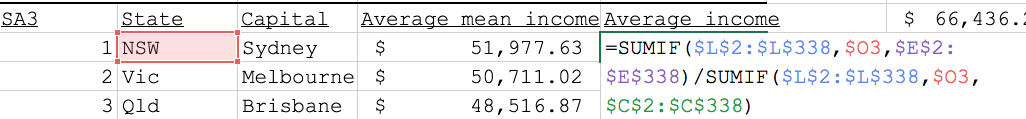
1. Similarly, use a similar function to list the capital cities of each observation.



1. Great. Now, finally, we’ll generate the 1) the average mean income per SA3 in each state, and 2) the average income per person in each state (note the difference).
   * 1) Average mean income per SA3 in each state



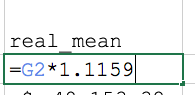
* + 2) Average income in each state



* + *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.**
   * 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: Short introduction to R Studio

1. Open R Studio
2. Set working directory to be your main folder.
3. Look up “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.*
5. **Bonus activity**: using the package ggplot to create scatters with OLS regression lines and confidence intervals separated by state.
   * A great resource for ggplot data visualisation is [here](http://socviz.co/).

# PART III: Creating Charts

1. Have a browse at the “Charts for reports” Powerpoint file to familiarise yourself with the “Grattan style”.
2. Use the template in 7\_scatter\_template.ppt and the data you have generated to create

**Further reading:**

* 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).
* The placement package reference manual can be found on [CRAN](https://cran.r-project.org/web/packages/placement/placement.pdf).
* A great resource data visualisation is [here](http://socviz.co/). Note that it’s a full book, but it’s pretty excellent.

1. There is enormous room for error in downloading a dataset and jumping into it with PivotTables. [↑](#footnote-ref-1)