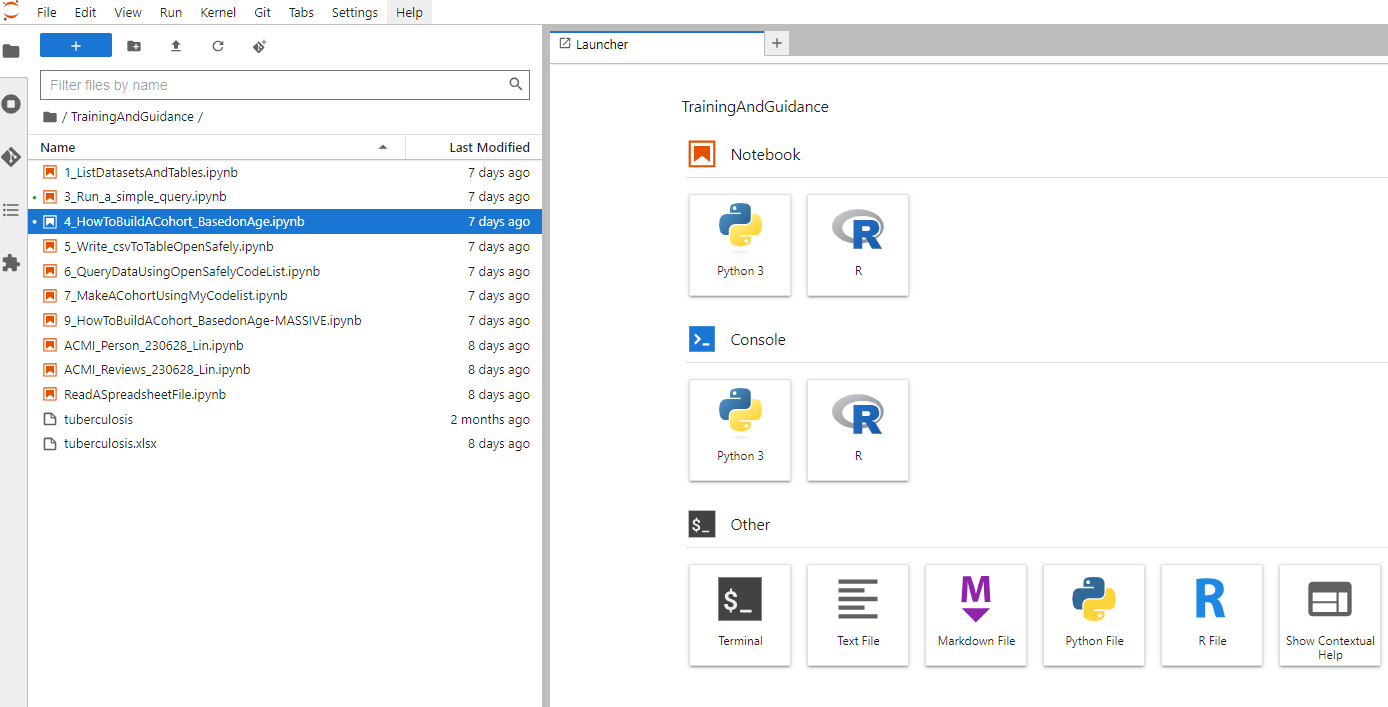
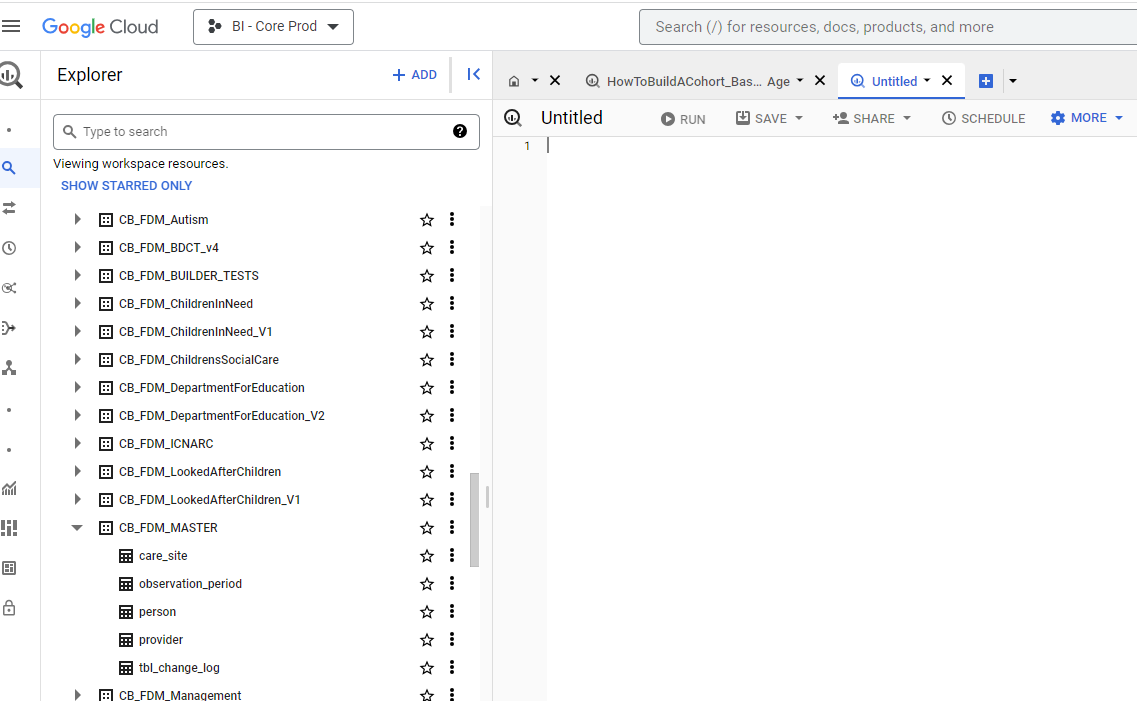
**The use of Bigquery and Jypter Notebooks**

**The similarities, and what you can do in them, and some Do’s and Don’ts**

The Notebook looks like this :



Bigquery looks like this:



The key difference between the two is that the Bigquery environment can only be used to look at data, to build new tables (based on existing data and update data (if you have permissions.

A Juypter notebook can be used to do all this, but you can then use the power or R to analyse your results.

The following example will show you how to run the same query in both environments: It will be used to

|  |  |
| --- | --- |
| The Query in R | The Query in Bigquery |
|  |  |

As you can see the query produces exactly the same results. The only difference is that the query in Notebook uses a R library to send the query into Bigquery to execute. This requires R to be told how to do it, how to structure the query and how to display the results.

Whilst at first glance its easier to run the query in Bigquery the data is not yest accessible to R for analysis. The final statement : table <- bq\_table\_download(tb3) actually brings the data into a dataframe. Also, the R script, in this example, uses parameters to specify the two table locations that the query uses. This means that if you have a series of queries using the same tables, and you want to “point” them somewhere else you simply change targetdb1 and targetdb2.

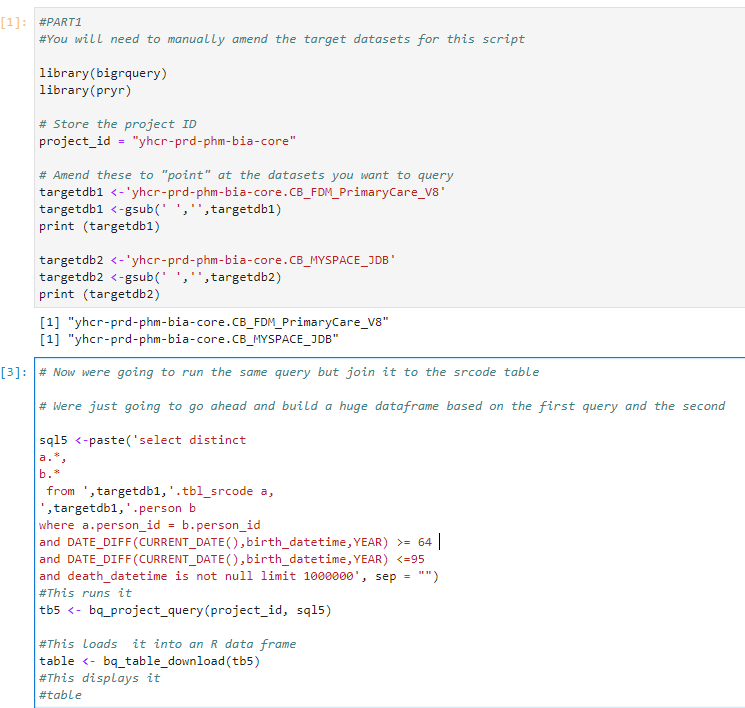
As you will be doing the analysis in R it is sometimes simpler to work purely within R.

Whichever way you do it, the part of the R scripts (tb3 <- bq\_project\_query(project\_id, sql3)) does exactly the same as running the script in Bigquery.

**What should you not do in R**

A R notebook is essentially a virtual pc. It has a limited amount of RAM, this can quickly be filled if you try to bring everything into a dataframe. To reduce this we need to think about what fields R will actually use to analyse and present data.

The following is an example of a bad script:



As you can see we are querying two tables and bringing back all the fields associated with those tables. These are being brought back into a single dataframe. So what are the issues?

1 – We may have duplicated fields

2 – Fields we don’t need for analysis

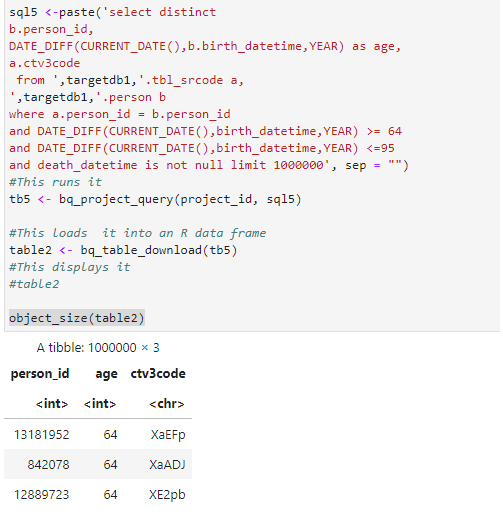
3 – too much data. Do we really need each line or can we aggregate it ?

In this example using actual data we use 371Mb of memory, given that there are over 700 million rows in the actual table this would be over 28Gb of data that we are trying to bring into a dataframe. Which we will then try to analyse

So what do we need to do ?

Remove repeating fields (in this case the person\_id is on each table.

Reduce the number of fields to those actually needed. The following is an example using this same join , but bringing back a very limited set of fields:



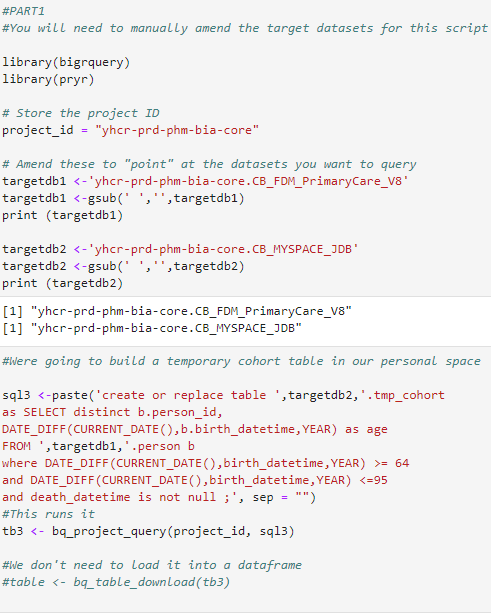
This query brings back just those fields we are interested in

In this case we use just 17Mb of memory that we are loading into the dataframe table2.

This means that we are using the power of Bigquery (via our sql statements) and R simply to do the analysis.

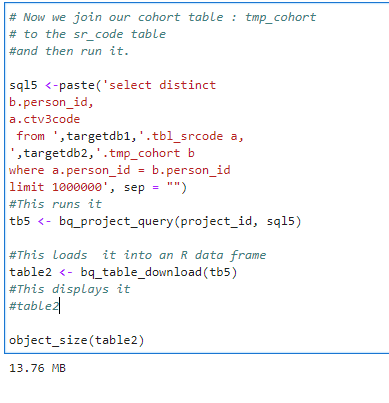
**How can I leverage this even more ?**

As you can see from this example the person\_id brings back the age, it brings that back for every record where we have a ctv3code. Do we need that for this part ? If all I need is the person\_id and the code then I can bring that back in a separate dataframe/query. Will I be analysing more tables for the same cohort ? If so I can create a temporary cohort table and use that. The following example tables the sql above and breaks it down even further:



This second part creates the temporary cohort table

We then run the actual query :



As you can see it is essentially the same query (less the repeating age column) but it is not only smaller, but it also runs quicker as it isn’t trying to filter by age for every record . This part :

“where DATE\_DIFF(CURRENT\_DATE(),birth\_datetime,YEAR) >= 64

and DATE\_DIFF(CURRENT\_DATE(),birth\_datetime,YEAR) <=95

and death\_datetime is not null”

Was run just the one, rather than every time, it will be quicker in both Bigquery and in R. Furthermore other queries can be run using the same tmp\_cohort.

The recommendation is therefore to use your CB\_MSPACE\_*AA*  to build temporary objects that define your cohorts, reduce the actual data fields you are analysing, aggregate data where possible, and then bring these into dataframes for analysis.