Census Graduate Earnings R code

Programming-level documentation for the Graduate Outcomes Census analysis project

*This documentation is intended to supplement the high-level purpose and methodology report,* 2018 Census Graduate Outcomes Project Overview. *Also note that further brief explanations and comments have been inserted into the code directly.*

This set of R codes transforms census data relating to tertiary qualifications and income into a set of datasets that are used for the Graduate Outcomes census-based tool on the Universities New Zealand website.

# Set-up

## Folders

You need to have the following folders:

* datasets
* input\_data
* metadata
* program\_costs
* Rbooks
* scripts
* working\_outputs

### Datasets

Currently only holds pre-imputation data. All other datasets are being stored in the SQL database.

### Input\_data

This folder contains the data which has been output from the Datalab or obtained from Stats NZ via custom request.

Excel workbook JOB-10148-1 - FIELD OF STUDY INCOME.XLSX contains the following columns:

* Work status
* Highest qualification
* Post School Qualification Field
* Age group
* Income brackets (including Loss and Zero income)
* Total count (stated and not stated)
* Mean Income
* Median Income

The t3 files, *t3 - Occupation by field of study.xlsx* and *t3 - Occupation by narrow field of study*.*xlsx* contain the occupation by field of study data which we could not get from the custom request so had to output from the IDI.

### Metadata

These are mostly lookup files downloaded from Stats NZ.

### Program\_costs

Contains two files: one, containing SAC funding codes mapped to NZSCED codes and NZQF levels, is required by the screen scraping notebook. The other is the output of this screen-scraping.

### Rbooks

This folder has several R Markdown notebooks, containing code chunks and comments. These need to be run in the order of their prefix as they create datasets which the subsequent notebooks use.

Explanation of what each set of codes does, including what datasets are produced, is given after this Set-up section.

### Scripts

Contains R scripts which contain functions which can be shared between R notebooks.

### Working outputs

Some CSV or Excel files are or were written to this file to have a look at the output.

* *Unz\_common.r*
* *Cross-tabulation.r*

## Database

This process needs an SQL database to be setup. The connection is created in the connect\_to\_GO\_analysis\_database function, defined in the unz\_common script.

## Output/final result

<https://www.nzgraduateoutcomes.ac.nz/EmploymentResearch>

* For each level of study, field of study, employment code (fulltime and all) and age bracket (30-39 and 20-65+), the following variables:
  + Counts
  + Average annual (age-standardised) income,
  + lifetime earnings (above school leaver)
  + Age when better off than a school leaver
  + % on benefit 2 and 5 years post graduation
* Most common occupations by field of study and most common fields of study by occupation

# Summary of process

1. Calculate average earnings for every age from 17-65, imputing values where necessary and allowed
2. Age-standardise incomes by calculating a weighted mean based on the proportion of people in each age bracket
3. Calculate tax and student loan payments for each qualification. Set income and tax of school-leavers (no post-school qualification) as baseline variables. Calculate average difference in tax paid and income received for each qualification graduate compared to baseline, taking into account student loan repayments. Sum incomes to get lifetime earnings and tax paid minus SAC funding.
4. Use these comparative values to calculate age at which ­graduates become ‘better off’ than school leavers, using profitability measure of Net Present Value (NPV) which takes into account inflation (‘discount rate’).
5. Output this data, grouping by 30-39 year-olds and everyone (limited to those who earned something). Also output most common occupations by field of study and most common fields of study by occupation.

# SS\_get\_fees\_durations\_and\_SAC\_funding (screen-scraping)

This file contains some Python and takes a long time to run. It web-scrapes SAC funding data from the TEC website and qualification fees and duration data from the Careers NZ website. It should not need to be run often. For further information on how this screen-scraping was done, see: [GO Obtaining Annual Fees, Durations and SAC funding](https://universitiesnewzealand.sharepoint.com/Shared/Shared%20Documents/Shared/Graduate%20Outcomes%20tool/Census%20based/2018_process/GO%20Obtaining%20Annual%20Fees,%20Durations%20and%20SAC%20funding.docx?web=1) > Obtaining SAC Funding

It outputs the dataset **NZSCED\_SAC\_rates.csv** and outputs **fees\_durations\_SAC.csv**, which contains the following variables for each field of study and qualification level:

* duration\_yr = Duration of qualification in years
* fee = course fees per year
* cum\_duration = total duration of study, including previous study time for postgrad qualifications
* fee component = total course fees for qualification
* none\_fee\_component = total non-fee borrowing for qualification, **set to $1 per year** (maximum living cost withdrawal)
* duration\_whole\_yr =total (cumulative) duration of study, rounded down to nearest whole year
* loan\_for\_course = total fees for qualification+ total non-fee borrowing
* loan\_at\_graduation = total fees for course + total non-fee borrowing, cumulative over all study
* sac\_funding = sac funding for qualification per year
* total\_funding = total sac funding for qualification

# 00\_env\_setup.rmd

Sets up database environment, checking necessary files and folders are at hand.

Creates ‘schemas’ - wide, long, tab – akin to dataset libraries in SAS.

# 01\_clean\_income\_dataset.rmd

Imports raw data (custom request from 2018 Census of income data: income by age, employment status, and field of study).

Cleans data: renaming columns, applying variable codes, labelling *total* rows (to remove later) and standard qualification levels we are interested in.

*Employment codes:*

1 = Full-time

2 = All

3 = Part-time

4 = (constructed) = Full-time earning $25,000+

We already have mean and median income for the whole group, but we want mean incomes for two subpopulations: those who earned anything and those who earned more than $25,000.

Within these two groups, calculate average income of each level and qualification using the following formula:

Where is the income for employment code , level of study , qualfication , and age bracket .

We round incomes to the nearest hundred.

*Tables such as the one below show which tables are saved to the database to be called from subsequent codes. + indicates this table has this feature added, - indicates it has been taken away from the previous table.*

|  |  |
| --- | --- |
| Table saved to database | Content |
| **lng.cen\_income\_bands** | Income band counts and average income for each employment status, qualification, and age bracket |
| **lng.cen\_annual\_avg\_income**  *Main reference dataset* | - income band columns  +average income for people who earned anything (*2)*  +average income for people who earned $25,000+ (*res*) |
| **lng.cen\_annual\_avg\_income2** | +average income for people who worked full-time and earned $25,000+ (*employment type 4)* |

# 02\_income\_imputation\_v2.rmd

Converts data from long to wide form, with age bracket as columns and mean income (of those who did not earn nothing or make a loss) as observations.

Using the imputation rules detailed in *GO high\_level*, we create an indicator for the rows on which imputation can and can’t be performed.

**We perform special functions on the 15-19 year old age range as they tend not have completed qualifications:**

Find maximum weighted income for each employment code across levels of study 1-7. This will be the imputation value at levels 8-10 for each of these level codes for this age bracket. We do this because people in this age bracket are very unlikely to have completed such high-level qualifications. We add these values to the imputation dataframe for this age bracket. we now have an imputation value for every combination of employment code and highest qualification level. We then add these values as an extra column in the wide income dataset.

Saves the table in **datasets** folder as **dump\_pre\_impute\_ft**

Wide-form data of average incomes (age bracket as columns) with age imputation indicator and 15-19yr imputation values.

Perform imputation, where possible, using rules outlined above.

**Calculating income at each age (17 to 65)**

Now that we have a more complete dataset, we can proceed to interpolate incomes for every age. The income for the midpoint age of each age bracket, e.g., 17 in the 15-19yr age bracket, is set to the income for that age bracket, with the other incomes falling between these values.

|  |  |
| --- | --- |
| Table saved to database | Content |
| **wide.cen\_annual\_income\_with\_imputation** | Wide-form post-imputation table with average income for every age bracket |
| **lng.cen\_yearly\_gross\_income.** | + converted to long-form, with interpolated average income for every age from 17-65 |

# 03\_age\_standardised\_income.rmd

Calculating weighted mean incomes for each employed-qual-field combination, based on age distribution of working population (incomes for more common age brackets are weighted more heavily than less common ones).

1. Converts complete (imputed) income age-bracket data to long format.
2. Finds proportion each age bracket makes up in the working population, for each employment code.
3. Use these values to calculate weighted mean incomes.

Connects to database, saving the table as **lng.cen\_inc\_age\_sd\_annual.**

Connects to database, saving the tables as **tab.cen\_inc\_age\_sd\_annual** and **tab.cen\_inc\_age\_sd\_annual\_counts**

*Wide-format of* **tab.cen\_inc\_age\_sd\_annual**, first with income responses and second with counts

|  |  |
| --- | --- |
| Table saved to database | Content |
| **lng.cen\_inc\_age\_sd\_annual** | *Long-format table with weighted mean income for every employ-qual-field combination* |
| **tab.cen\_inc\_age\_sd\_annual** and **tab.cen\_inc\_age\_sd\_annual\_counts** | + converted to wide-form, first with income as observations and second with counts |

# 04\_income\_30\_39\_yr\_analysis.rmd

Sets up two large functions:

get\_30\_39\_income\_summary\_set(), calculates average income by employment status, field of study and qual level.

tabulate\_and\_save\_income\_set() produces two wide-format tables (individual levels as columns) and saves them to the database: one of average incomes and one of counts.

As per usual, we use the variables mean\_income2 and people\_count2 (those who received any positive income).

We also limit to rows with counts > 3, and after performing the first function above, further limit to those with counts > 20. This is to remove observations which would be suppressed or otherwise behave unexpectedly in analysis.

|  |  |
| --- | --- |
| Table saved to database | Content |
| **lng.cen\_annual\_3\_avg\_income\_30\_39.** | *Long-format table with average incomes for 30-39 year-olds.*  SAME FOR EMPLOYED\_CODE=4 (tables have res suffix) |
| **tab.annual\_avg\_3\_income\_30\_39** and **tab.annual\_avg\_3\_income\_30\_39\_people\_counts** | *+ wide format* |

# 04\_lifetime\_earnings.rmd

Here we are doing two main things.

1. Calculate the lifetime earnings of the graduate of a certain qualification and how this compares to that of a school leaver with no qualification. Taking into account inflation, we can then find the average age at which a graduate is ‘better off’, given the years spent studying and not earning. *Assuming all students are studying full-time.*
2. Calculate the lifetime taxes paid by a graduate of a certain qualification and how this compares to a school leaver. Taking into account inflation and the amount of SAC funding provided by the government for each qualification, we can then find (though it is not yet done here) the number of years after which this investment becomes profitable for the government.

The prefix discounted indicates inflation-adjusted, using a constant rate of 3% inflation.

From an external file fees\_duration\_SAC we import and link qualification information including qualification length, fees, and student loan at graduation (broken down into fee\_component and none\_fee\_component (living costs and course-related costs). This is linked to income dataset to create income\_with\_fees.

For a given age and qualification,

1. offset\_income = gross\_income, unless the age is younger then someone could have feasibly completed the qualification, in which case it = 0.
2. loan\_repayments = 12%\*offset\_income
3. We calculate a *baseline income* and *tax*. This is the average income and tax paid each year for a school leaver who did not get a qualification. We add these figures to the income table and calculate the difference for each qualification from these numbers
4. Above\_school\_leaver = offset\_income - baseline

## Calculating Net Present Value (NPV)

Investigate whether the increased tax paid by graduate over their lifetime, relative to a school-leaver, is greater than the SAC funding put towards their qualification (which would result in a positive NPV and therefore a profitable government investment)

*First,* calculated discounted\_income (the inflation-adjusted income for each age since 17-years-old). Sum these across lifetime to get npv.

Then, calculate the difference in (un-adjusted) income from school leaver income = above\_school\_leaver. Adjust these for inflation (discounted\_compared\_school\_leaver), and sum these to get npv\_over\_school\_leaver

We find the NPV by calculating the difference between the funding for a qualification and the tax paid by the graduate each year (tax\_paid\_above\_school\_leaver\_m\_sac), adjusting these values for inflation, and then summing across all years to get npv\_tax\_over\_school\_leaver

Find age when NPV (profitability measure) becomes positive – this is the age at which the average graduate of a qualification becomes better off than the average school-leaver

|  |  |
| --- | --- |
| Table saved to database | Content |
| **lng.cen\_lifetime\_income** | *Long-format table with statistics of lifetime earnings relative to school-leavers and accounting for inflation* |
| **tab.cen\_lifetime\_earnings\_above\_school\_leaver** | + wide-format |
| **tab.cen\_age\_better\_off** and **lng.age\_better\_off** | Wide and long tables where responses are age at which graduates are better off than school-leavers |

# 05\_calculate\_typical\_study\_level

This code find the most common levels of study for people in different occupations.

This follows rules as explained in the Grad ROI Project Write Up document.

1 = school

2 = certificates and diplomas

3 = mixed

4 = tertiary

5 = degree

Connects to database, saving the table as **typical\_study\_levels**.

Typical study level (>85% of people in job or >90% for all tertiary quals) for each occupation

# 05\_consolidate\_long\_income\_datasets.rmd

Connects to database, saving the table as **lng.consolidated\_income**

Dataset with 30-39 year old fulltime income, all ages standardised fulltime income, counts (all limited those who earned anything), and ‘better off’ variables (which apply to those who work full-time)

# 06\_produce\_grad\_ROI\_executive.rmd

This code links all the datasets together and outputs to an Excel workbook with four sheets, complete with formatting.

Connects to database, saving the table as **lng.top\_occupations**

Very wide table with average incomes and ‘better off’ variables for each qualification, as well as up to 30 occupations for each qualification

Saves data as **Graduate ROI Occupation by Field of Study.xlsx**

Sheets:

1. Top 30 Qualifications
2. Top 25 Jobs
3. Occupation by field of study
4. Field of study by occupation

# 10\_populate\_GO\_website\_DB.rmd

Populates new databases with analysis data creates thus far, lookup table information, and additional information from external websites. This is then used to feed into the website.

Established 3 database connections, only the first of which (go\_analysis\_data) which has been created in the codes thus far.